

Федеральное Агентство связи

**Государственное образовательное учреждение высшего профессионального
образования**

Г. К. Сосновиков, Л. А. Воробейчиков

**ТЕХНОЛОГИЯ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ УПРАВЛЕНИЯ РЕЛЯЦИОННОЙ
БАЗЫ ДАННЫХ**

ЛАБОРАТОРНЫЕ РАБОТЫ

Москва 2016

УДК 004.652.4

Издание утверждено _____ Протокол № _____ от _____. 16.

Рецензент: Т.И. Семенова, канд. техн. наук, доцент

Оглавление

ВВЕДЕНИЕ	6
ЛАБОРАТОРНАЯ РАБОТА 1	6
ТЕМА: СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦ. ИЗМЕНЕНИЕ	
СТРУКТУРЫ ТАБЛИЦ. СОРТИРОВКА И ПОИСК ДАННЫХ.	6
1.1. Цель работы.....	6
1.2. Общие сведения. Основные принципы проектирования баз данных. Основные определения	6
1.3. Типы связей между объектами	7
1.4. Структура MS Access	8
1.5. Начало работы с MS Access. Создание новой таблицы.....	8
1.6. Типы данных MS Access	10
1.7. Определение ключевых полей	10
1.8. Определение связей таблиц.....	11
1.9. Использование Мастера подстановок при определении структуры таблиц и вводе данных в них.....	11
1.10. Изменение структуры таблиц	12
1.11. Переименование и удаление таблиц	13
1.12. Редактирование связей	13
1.13. Изменение порядка следования столбцов	14
1.14. Другие изменения макета таблицы	14
1.15. Сортировка данных	14
1.16. Поиск данных	15
1.17. Порядок выполнения работы	15
1.18. Содержание отчета	18
1.19. Контрольные вопросы	18
ЛАБОРАТОРНАЯ РАБОТА 2	19
ТЕМА: СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ	19
2.1. Цель работы	19
2.2. Основные сведения о формах	19
2.3. Способы создания форм	19
2.4. Использование Мастера по созданию форм	20
2.5. Создание форм в режиме Конструктора.....	20
2.6. Разделы форм.....	20
2.7. Панель элементов и Список полей	20
2.8. Свойства объектов формы.....	22
2.9. Создание управляющих кнопок	22
2.10. Управление элементами формы	22
2.11. Изменение порядка обхода элементов формы	23
2.12. Разработка сложных форм	23
2.13. Порядок выполнения работы	25
2.14. Содержание отчета	27
2.15. Контрольные вопросы	27
ЛАБОРАТОРНАЯ РАБОТА 3	28
ТЕМА: РАБОТА С ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ЗАПРОСОВ. ЗАПРОСЫ-	
ВЫБОРКИ.	28
3.1. Цель работы: изучение работы с данными при помощи запросов	28
3.2. Понятие запроса	28
3.3. Создание запроса	28
3.4. Окно конструктора запроса	30
3.5. Включение полей в запрос	30
3.6. Установка критериев отбора записей.....	31
3.7. Виды критериев.....	31
3.8. Вычисляемые поля	33
3.9. Использование построителя выражений	33
3.10. Итоговые запросы.....	33
3.11. Перекрестные запросы	34

3.12. Использование Мастера для создания перекрестной таблицы	35
3.13. Выполнение запроса	36
3.14. Порядок выполнения работы.....	37
3.15. Содержание отчета	39
3.16. Контрольные вопросы	39
ЛАБОРАТОРНАЯ РАБОТА 4	40
ТЕМА: РАБОТА С ДАННЫМИ С ПОМОЩЬЮ ЗАПРОСОВ-ДЕЙСТВИЙ.	40
4.1. Цель работы: приобрести навыки по работе с данными с помощью запросов-действий.....	40
4.2. Понятие запросов-действий.....	40
4.3. Запросы на создание таблицы	40
4.4. Запросы на обновление записей	41
4.5. Запрос на удаление записей.....	42
4.6. Запрос на добавление записей	43
4.7. Порядок выполнения работы	44
4.8. Содержание отчета	45
4.9. Контрольные вопросы	45
ЛАБОРАТОРНАЯ РАБОТА 5	46
ТЕМА: ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ. РАЗРАБОТКА ER-МОДЕЛИ. СОЗДАНИЕ БД, РАЗРАБОТКА ЗАПРОСОВ К БД.	46
5.1. Цель данной работы состоит в получении навыков в проектировании БД и построении запросов к ней в соответствии со своим индивидуальным вариантом задания.	46
5.2. Вопросы, подлежащие изучению.....	46
5.3. Задание	46
5.4. Содержание отчета	46
5.5. Пример разработки ER-модели БД «Учебный процесс».....	47
5.6. Варианты заданий.....	48
ЛАБОРАТОРНАЯ РАБОТА 6	57
ТЕМА: СОЗДАНИЕ ОТЧЕТОВ.....	57
6.1 Цель работы: освоить приемы создания отчетов и графического представления данных в СУБД ACCESS	57
6.2. Основные сведения об отчетах.....	57
6.3. Выбор мастера по разработке отчетов	57
6.4. Разделы отчета	58
6.5. Окно отчета и его инструменты	58
6.6. Сортировка и группировка.....	59
6.7. Задание диапазонов группировки.....	60
6.8. Использование вычисляемых значений	60
6.9. Добавление текущей даты и номера страницы.....	60
6.10. Вычисления в области данных	61
6.11. Вычисление промежуточных итогов для групп и общего итога.....	61
6.12. Объединение текстовых значений и скрытие повторяющихся значений.....	61
6.13. Вычисление процентов.....	62
6.14. Вычисление итогов с накоплением	62
6.15. Создание и внедрение подчиненных отчетов	62
6.16. Порядок выполнения работы.....	63
6.17. Содержание отчета	67
6.18. Контрольные вопросы	67
ЛАБОРАТОРНАЯ РАБОТА 7	68
ТЕМА: ИСПОЛЬЗОВАНИЕ МАКРОСОВ В ACCESS	68
7.1. Цель работы: изучение возможностей Access по созданию и использованию макросов.	68
7.2. Понятие макрокоманды и макроса	68
7.3. Создание макроса	68
7.4. Запуск макроса	68
7.5. Выполнение макроса по шагам	69
7.6. Создание кнопки для запуска макроса в режиме формы	69
7.7. Создание кнопки с помощью мастера	69
7.8. Создание кнопки с помощью мыши	70

7.9. Изменение вида кнопки.....	70
7.10. Макрос AutoExec	70
7.11. Порядок выполнения работы.....	70
7.12. Содержание отчета	72
7.13. Контрольные вопросы	72
ЛАБОРАТОРНАЯ РАБОТА 8	73
ТЕМА: ИЗУЧЕНИЕ CASE СРЕДСТВА ERWIN.....	73
8.1. Использование Erwin для разработки модели данных. Логическая модель. Сущности и связи	73
8.2. Порядок выполнения 1-ой части работы.....	74
8.3. Использование Erwin для разработки модели данных. Логическая модель. Атрибуты сущностей. Физическая модель	75
8.4. Порядок выполнения 2-ой части работы.....	76
8.5. Содержание отчета	78
ЛАБОРАТОРНАЯ РАБОТА №9	79
ТЕМА: ГЛАВНАЯ КНОПОЧНАЯ ФОРМА. ИТОГОВОЕ ЗАДАНИЕ.....	79
9.1. Цель работы: изучить способы создания кнопочной формы. Создать БД и интерфейс для работы с БД.....	79
9.2. Главная кнопочная форма.....	79
9.3. Порядок выполнения лабораторной работы.....	79
Индивидуальные задания по проектированию БД.....	82
Задание №1: БД Ресторана	82
Задание №2: БД Банка.....	83
Задание №3: БД Больницы.....	84
Задание №4: БД Гостиницы.....	85
Задание №5: БД Отдела внутренних дел.....	86
Задание №6: БД Авиакомпании.....	87
Задание №7: БД Видео-проката.....	88
Задание №8: БД Библиотеки.....	89
Задание №9: БД Радиостанции.....	90
Задание №10: БД Таксопарка.....	91
Задание №11: БД Туристического агентства.....	92
Задание №12: БД Страховой компании.....	93
Задание №13: БД Брачного агентства.....	94
Задание №14: БД Сервис-центра.....	95
Задание №15: БД Школы.....	96
Задание №16: БД Транспортной компании.....	97
Задание №17: БД Проката автомобилей.....	98
Задание №18: БД Оптового склада.....	99
Задание №19: БД картинной галереи	100
Задание №20: БД Спортивного клуба.....	101
Задание №21: БД Рекламного агентства наружной рекламы.....	102
Задание №22: БД Компьютерной фирмы.....	103
Задание №23: БД ГИБДД.....	104
Задание №24: БД Кинотеатра	105
Задание №25: БД Автосалона.....	106

ВВЕДЕНИЕ

Данное пособие включает в себя лабораторные работы по использованию СУБД Access. Каждая работа содержит учебный материал (применительно к Access 2003), который необходимо изучить, и упражнения, которые необходимо выполнить студенту. Упражнения ориентированы на использование Access 2010. Особенностью пособия является включение в работу студента использования технологии проектирования СУБД с помощью ER-схем и автоматизации этого процесса с использованием CASE - средства ERWin.

Использование СУБД Access более старших версий, чем Access 2003, требует изучения особенностей этих версий с помощью соответствующей литературы (например [4-6]). Порядок выполнения работ остается тот же.

Лабораторная работа 1

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

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.

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:

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 студентов и наименования дисциплин, по которым сдаются 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.

В настоящей лабораторной работе будет создана база данных, содержащая информацию о студентах различных групп и факультетов, а также об их успеваемости по различным дисциплинам. При этом можно выделить следующие классы объектов и установить соответствующий им состав таблиц:

таблица **Факультеты**, содержащая информацию о факультетах, на которых обучаются студенты;

таблица **Группы**, содержащая информацию об учебных группах, в которых обучаются студенты, в том числе о принадлежности каждой группы к тому или иному факультету;

таблица **Студенты**, содержащая информацию об анкетных данных студентов и об их принадлежности к той или иной учебной группе;

таблица **Дисциплины**, содержащая информацию об учебных дисциплинах, по которым фиксируется успеваемость студентов;

таблица **Оценки**, содержащая информацию о полученных оценках на экзаменах отдельными студентами по различным дисциплинам.

Для ввода, редактирования и обработки данных, содержащихся в этих таблицах, между ними должны быть установлены связи.

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. Одна из связываемых таблиц представляет сторону "один", другая - сторону "много". Например, каждая запись таблицы **Факультеты** содержит уникальную информацию о данном конкретном факультете. В то же время многие записи в таблице **Группы** относятся к одним и тем же факультетам. Поэтому между таблицами **Факультеты** и **Группы** должна быть установлена связь типа “один-ко-многим”.

Для установления этого типа связи в таблице со стороны "один" должно быть выделено поле, однозначно идентифицирующее и уникальное для каждой записи (**ключевое поле** или **первичный ключ**). Если среди атрибутов таблицы нет естественных кандидатов на эту роль, то ключевое поле создается искусственным образом. В СУБД ACCESS для этой цели может быть использован специальный тип данных - “счетчик”, формирующий в соответствующем поле (обычно со стандартным именем **Код**) уникальный целочисленный номер для каждой записи.

Таблица со стороны "много" должна содержать поле того же типа и длины, что и ключевое поле (так называемый **внешний ключ**). Каждая запись таблицы должна

содержать в этом поле значение соответствующего первичного ключа таблицы со стороны "один", чем и обеспечивается связь между таблицами.

Так, например, как мы увидим далее, таблица **Факультеты** будет иметь первичный ключ типа “счетчик” с именем **Код**, а таблица **Группы** – внешний ключ с именем **КодФакультета**. Связь между таблицами будет установлена по этим полям.

Связь типа “много-ко-многим” (М : М)

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 "один" и одна со стороны "много". Связь между этими тремя таблицами также осуществляется по ключевым 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, определять 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 Access 2010 - .accdb). 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.

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

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

Формат устанавливает вид данных на экране, например, для типа Дата/Время можно выбрать представление даты вида 30 июня 1999 г. или 6/30/99.

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

Маска ввода позволяет задать шаблон для ввода, который дает некоторую гарантию правильности ввода данных, она применяется для полей типа Дата/Время и текстовых. Маска ввода выводит на экран символы – местоименители, показывает, сколько символов нужно ввести, включает разделительные символы (дефис, скобки). Например, для даты маска ввода может выглядеть так: --.---.---. Данная маска кодируется набором символов 99/99/00. Заполнитель 9 означает, что разрешается вводить только цифры, причём её ввод не является обязательным, заполнитель 0 требует обязательного ввода цифры. При выборе свойства Маска ввода справа появляется кнопка , по которой запускается мастер, помогающий создать маску.

Подпись используется в качестве заголовка столбца в режиме таблицы.

Значение по умолчанию позволяет автоматически вводить в поле какое-либо значение.

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

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

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

Просмотреть и отредактировать созданные индексы возможно в режиме конструктора таблиц в пункте меню **Индексы** (Access 2010).

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

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**. Следует отметить, что такой способ формирования первичного ключа является предпочтительным, обладая рядом важных преимуществ:

снимается проблема обновления внешних ключей в связанных таблицах, так как первичный ключ типа “счетчик” не может быть изменен;

разработчик базы данных получает полную свободу в изменении содержимого и свойств неключевых полей таблицы, так как такие изменения уже не влияют на связанные таблицы;

поле типа “счетчик” имеет относительно небольшую длину (8 байтов), что позволяет экономно использовать память в связанных таблицах;

упрощаются и унифицируются связи между таблицами.

Если по каким-либо причинам разработчик БД хочет использовать другое поле таблицы в качестве первичного ключа, то 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 связей 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 таблицы со стороны “один”, нажать кнопку мыши, перетащить ее курсор на соответствующее поле в связываемой таблице, после чего кнопку мыши отпустить. В результате появится диалоговое окно **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 таблицу со стороны “много” такого значения внешнего ключа, которого нет в таблице со стороны “один”.

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

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

Выбор второй опции при удалении одной из записей в таблице со стороны “один” приведет к удалению тех записей в таблице со стороны “много”, которые имеют такое же значение ключа. Например, если из таблицы **Факультеты** удалить запись об одном из факультетов, то записи обо всех группах удаленного факультета в таблице **Группы** будут удалены автоматически. Поэтому к выбору данной опции следует относиться с осторожностью. При отказе от каскадного удаления при каждой попытке удаления записи таблицы со стороны “один” Access будет сообщать о наличии записей с соответствующим внешним ключом в связанных таблицах, и предлагать пользователю удалить эти записи. Так, например, если таблица **Группы** содержит записи о группах некоторого факультета, то удаление записи об этом факультете из таблицы **Факультеты** станет возможно только после удаления всех таких записей из таблицы **Группы**.

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

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

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

Если первичными ключами являются поля типа “счетчик”, то использование подстановок является единственным разумным способом организации ввода данных в

связанные таблицы. В этом случае вообще отпадает необходимость видеть первичные и внешние ключи таблиц и оперировать с ними.

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

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

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

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

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

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

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

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

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 **Группы** по возрастанию в поле **КодФакультета** в первую очередь и по возрастанию в поле **НомерГруппы** - во вторую, то по возрастанию номера будут расположены только группы 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. Порядок выполнения работы

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

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

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

Имя поля	Тип поля	Свойства поля
Код	Счетчик	Ключевое поле
НомерГруппы	Текстовый	Размер – 7; обязательное; индексированное (совпадения не допускаются) ; маска ввода - >LLL9999 (3 прописные буквы и 4 цифры)
КодФакультета	Числовой	Размер – длинное целое; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков
ГодСоздания	Числовой	Размер – целое; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков; маска ввода – 9999 (4 цифры); значение по умолчанию – 2014; условие на значение - Between 2010 And 2015 ; сообщение об ошибке – Недопустимый год создания группы

8. Добавить таблицы **Факультеты** и **Группы** в схему данных базы.
9. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодФакультета** из таблицы **Факультеты** (см. п. 1.9).
10. Открыть окно индексов (см. п. 1.5) и убедиться в наличии всех индексов, заданных при формировании структуры таблицы.
11. Открыть окно схемы данных и убедиться в появлении связи между таблицами.
12. Перевести таблицу **Группы** в режим таблицы и ввести несколько записей, выбирая из комбинированного списка различные факультеты.
13. Отсортировать таблицу по факультетам и номерам групп, используя для этого расширенный фильтр.
14. Создать в режиме Конструктора следующую структуру таблицы **Студенты**:

Имя поля	Тип поля	Свойства поля
Код	Счетчик	Ключевое поле
НомерЗачетки	Текстовый	Размер – 10; обязательное; индексированное (совпадения не допускаются)
ФИО	Текстовый	Размер – 25; обязательное; индексированное (допускаются совпадения)
БаллыЕГЭ	Числовой	Размер – целое; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков; маска ввода – 999 (3 цифры);
КодГруппы	Числовой	Размер – длинное целое; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков
ДатаРождения	Дата/время	Краткий формат даты; маска ввода формируется конструктором; обязательное; индексированное (допускаются совпадения)
Город	Текстовый	Размер – 20; обязательное; индексированное (допускаются совпадения) ; по умолчанию - Москва
Стипендия	Денежный	0 дес. знаков; маска ввода – 9 999р. ; обязательное; индексированное (допускаются совпадения) ; по умолчанию - 0

15. Добавить таблицу **Студенты** в схему данных базы.
16. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодГруппы** из таблицы **Группы** (см. п. 1.9).
17. Открыть окно индексов и убедиться в наличии всех индексов, заданных при формировании структуры таблицы.
18. Открыть окно схемы данных и убедиться в появлении связи между таблицами **Группы** и **Студенты**.
19. Перевести таблицу **Студенты** в режим таблицы и ввести 10-15 записей, выбирая из комбинированного списка различные группы.
20. Отсортировать таблицу по номерам зачетных книжек.
21. Найти в таблице **Студенты** все записи, удовлетворяющие следующим условиям:
- студенты, чьи фамилии начинаются с определенной буквы;

- студенты, обучающиеся в одной определенной группе.

22. Найти записи для студентов, проживающих в Москве, и очистить поле **Город** в этих записях с помощью функции **Поиск и замена**.

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

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

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

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

26. Создать в режиме Конструктора следующую структуру таблицы **Оценки**:

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

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

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

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

30. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодСтудента** из таблицы **Студенты** (см. п. 1.9). Для удобства идентификации студентов при вводе данных использовать в подстановке поля **НомерЗачетки** и **ФИО** таблицы **Студенты**.

31. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодДисциплины** из таблицы **Дисциплины** (см. п. 1.9).

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

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

34. Отсортировать таблицу по студентам и дисциплинам, используя для этого расширенный фильтр.

1.18. Содержание отчета

1. Тема и цель работы.
2. Краткое описание последовательности выполнения.
3. Описание структуры создаваемой БД и ее таблиц.

1.19. 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. Перечислите типы данных в Access.
6. Как используется Мастер подстановок?
7. Каковы возможности Access по изменению структуры таблиц?
8. Как отсортировать записи по одному полю?
9. Как отсортировать записи по нескольким полям?
10. Как произвести поиск и замену данных в Access?

Лабораторная работа 2

ТЕМА: СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ

2.1. Цель работы

Изучение приемов создания, редактирования и использования форм для ввода и редактирования данных в таблицах ACCESS

2.2. 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 (подробно описаны в лабораторной работе № 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.

2.3. 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.

2.4. 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.

2.5. 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**.

2.6. 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.

2.7. 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.

2.8. 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.

2.9. 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.

2.10. 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**.

2.11. 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.

2.12. 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).

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.

2.13. Порядок выполнения работы

1. Создайте форму для ввода и редактирования записей в таблице **Факультеты**:
 - выберите в списке таблиц базы данных таблицу **Факультеты**;
 - на вкладке меню **Создание/Другие формы** выберите режим **Таблица**;
 - в появившемся окне с формой выберите функцию сохранения и сохраните форму, дав ей имя **Факультеты**; при этом в окне навигации появляется заголовок **Формы** и строка с именем созданной формы.
2. Используйте созданную форму для редактирования списка факультетов, обеспечив наличие не менее 2-3 записей в таблице.
3. Упорядочьте записи в списке по возрастанию наименования факультетов.
4. С помощью кнопок перемещения по записям перейдите на следующую запись, на последнюю запись, на предыдущую запись, на первую запись.
5. Создайте форму для ввода и редактирования записей в таблице **Дисциплины**:
 - выберите в списке таблиц базы данных таблицу **Дисциплины**;
 - на вкладке меню **Создание/Другие формы** выберите режим **Разделенная форма**;
 - в появившемся окне с формой выберите функцию сохранения и сохраните форму, дав ей имя **Дисциплины**; при этом в окне навигации появляется новая строка с именем созданной формы.
6. Используйте созданную форму для редактирования списка факультетов, обеспечив наличие не менее 5 записей в таблице. Редактирование текущей записи следует производить в верхней части формы, а добавление и удаление записей – в списке записей в нижней части формы.
7. Упорядочьте записи в списке по возрастанию названия дисциплин.
8. С помощью кнопок перемещения по записям перейдите на следующую запись, на последнюю запись, на предыдущую запись, на первую запись.
9. Создайте форму для ввода и редактирования записей в таблице **Группы**, сгруппированным по факультетам:
 - выберите в списке таблиц базы данных таблицу **Факультеты**;
 - на вкладке меню **Создание/Формы** выберите и нажмите кнопку **Форма**; при этом появляется окно с формой, в верхней части которой находятся поля с атрибутами текущего факультета, а в нижней – список групп текущего факультета;
 - перейдите в режим **Конструктор** и измените заголовок формы на **Группы по факультетам**;
 - сохраните форму под таким же именем;
 - перейдите в **Режим формы** и отредактируйте списки групп по факультетам, обеспечив наличие не менее 5 групп на каждом факультете;
 - с помощью кнопок перемещения выполните перемещения как по факультетам, так и по группам выбранного факультета.
10. Создайте форму для ввода и редактирования записей в таблице **Студенты** в общем списке:
 - на вкладке меню **Создание** выберите и нажмите кнопку **Мастер Форм**;
 - в появившемся окне выберите таблицу **Студенты**, из списка доступных полей выберите все поля кроме поля **Код** и нажмите кнопку **Далее**;
 - в появившемся окне выберите внешний вид формы **Ленточный** и нажмите кнопку **Далее**, а в следующем окне введите имя формы **Студенты общий список** и нажмите кнопку **Готово**;
 - на экране появляется окно с формой, созданной **Мастером форм**, а форма с именем **Студенты общий список** появляется в списке окна навигации;
 - для подгонки ширины полей, их выравнивания, редактирования надписей и т.п. используйте **Режим макета** и **Конструктор**; обратите внимание на то, что в поле

КодГруппы Мастер форм автоматически поместил комбинированный список для выбора номера группы.

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

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

13. Создайте форму для ввода и редактирования записей в таблице **Студенты**, сгруппированным по группам:

- на вкладке меню **Создание** выберите и нажмите кнопку **Мастер Форм**;
- в появившемся окне выберите сначала таблицу **Студенты**, из списка доступных полей выберите все поля кроме поля **Код** и **КодГруппы**;
- затем выберите таблицу **Группы**, из списка доступных полей выберите поля **НомерГруппы** и **Код факультета** и нажмите кнопку **Далее**;
- в появившемся окне выберите вид представления **Группы** и **Подчиненные формы** и нажмите кнопку **Далее**;
- в следующем появившемся окне выберите внешний вид подчиненной формы **Ленточный** и нажмите кнопку **Далее**;
- в следующем окне задайте имя основной формы **Студенты по группам** и нажмите кнопку **Готово**;
- на экране появляется окно с двумя формами, созданными **Мастером форм**, а их имена (**Студенты по группам** и **Студенты подчиненная форма**) появляются в списке окна навигации; в основной форме расположены поле с номером группы и комбинированный список с наименованием факультета, в подчиненной форме – список студентов текущей группы с полями для ввода и редактирования данных;
- для подгонки расположения форм в окне, ширины полей, их выравнивания, редактирования надписей и т.п. используйте **Режим макета** и **Конструктор**; обратите внимания на то, что хотя созданный набор форм предназначен для редактирования списка студентов и их атрибутов, вы можете при необходимости редактировать и атрибуты групп: номер и факультет; если вы хотите запретить такое редактирование в этой форме, следует в режиме **Конструктор** установить для этих объектов свойству **Блокировка** значение **Да**.

14. Используя созданную форму, добавьте и отредактируйте записи в таблице **Студенты**, обеспечив наличие не менее 5 студентов в каждой группе, а также следующие условия, используемые в дальнейших лабораторных работах:

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

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

- на вкладке меню **Создание** выберите и нажмите кнопку **Мастер Форм**;
- в появившемся окне выберите сначала таблицу **Оценки**, из списка доступных полей выберите поля **КодДисциплины** и **Оценка**;
- затем выберите таблицу **Студенты**, из списка доступных полей выберите поля **НомерЗачетки** и **ФИО** и нажмите кнопку **Далее**;

- в появившемся окне выберите вид представления **Студенты и Подчиненные формы** и нажмите кнопку **Далее**;

- в следующем появившемся окне выберите внешний вид подчиненной формы **Табличный** и нажмите кнопку **Далее**;

- в следующем окне задайте имя основной формы **Оценки студентов** и нажмите кнопку **Готово**;

- на экране появляется окно с двумя формами, созданными **Мастером форм**, а их имена (**Оценки студентов** и **Оценки подчиненная форма**) появляются в списке окна навигации; в основной форме расположены поля с номером зачетки и ФИО студента, в подчиненной форме – список сданных экзаменов текущего студента с комбинированным списком для выбора дисциплины и полем для ввода оценки;

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

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

17. Создать форму для ввода и редактирования записей в таблице **Оценки**, сгруппированным по дисциплинам. Порядок создания формы аналогичен п.15 с учетом изменившейся группировки записей.

2.14. Содержание отчета

1. Тема и цель работы.
2. Краткое описание последовательности выполнения.

2.15. Контрольные вопросы

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

Лабораторная работа 3

ТЕМА: Работа с данными с использованием запросов. Запросы-выборки.

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

3.2. 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.

В Microsoft Access есть несколько типов запросов:

1. Запросы на выборку - наиболее часто используемый тип запроса. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

2. Запросы с параметрами (параметрические запросы) - это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные, например условие для возвращения записей или значение, которое требуется вставить в поле. Можно разработать запрос, выводящий приглашение на ввод нескольких единиц данных, например двух дат. Затем Access может вернуть все записи, приходящиеся на интервал времени между этими датами. Запросы с параметрами также удобно использовать в качестве основы для форм, отчетов и страниц доступа к данным.

3. Перекрестные запросы - запросы, которые используются для расчетов и представления данных в структуре, облегчающей их анализ.

Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

4. Запросы на изменение - запрос, который за одну операцию изменяет или перемещает несколько записей. Существует четыре типа запросов на изменение: запросы на удаление записей, на обновление, на добавление, на объединение записей, на создание таблицы. Последний запрос создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Запрос на создание таблицы полезен при создании таблицы для экспорта в другие базы данных Access или при создании архивной таблицы, содержащей старые записи.

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.

3.3. 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).

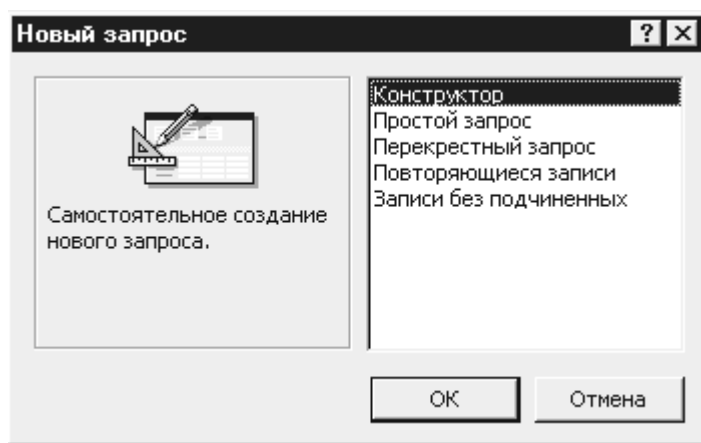


Рис. 1. Диалоговое окно **Новый запрос**

(В Access 2010 после вкладки **Создать** можно выбрать **Мастер запросов** или **Конструктор запросов**).

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

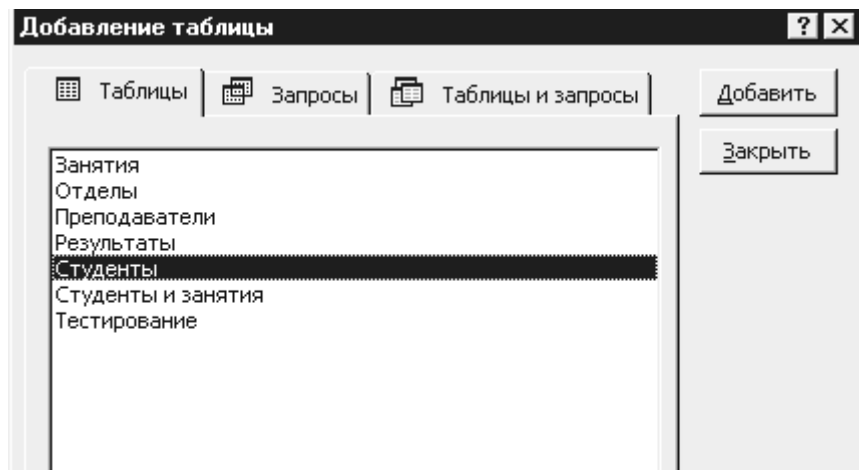


Рис 2. Окно **Добавление таблицы**

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.

3.4. 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.

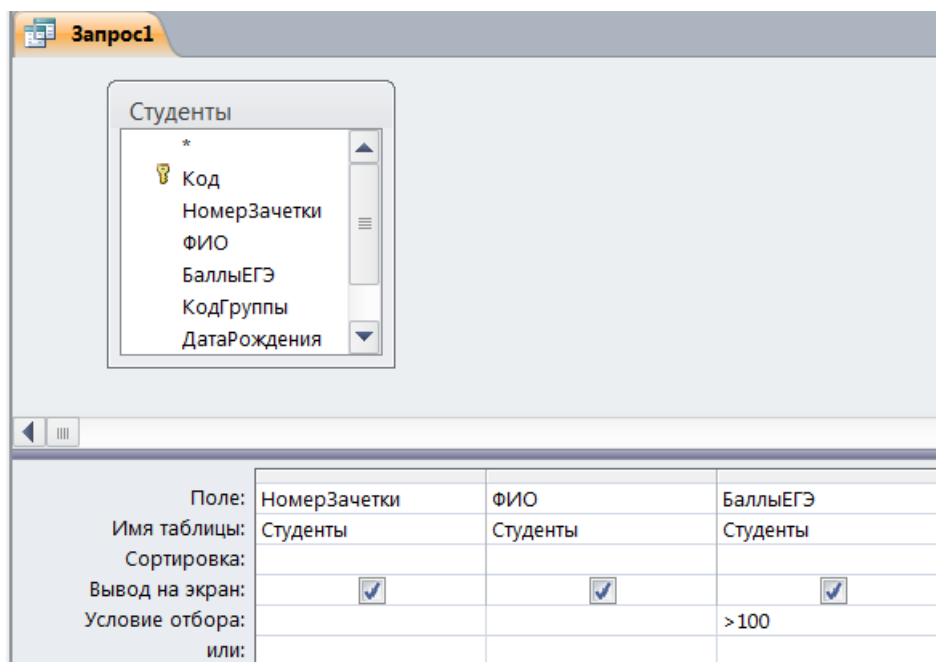


Рис 3. Окно конструктора запроса. Пример ввода условия.

Например, на рис.3 представлен бланк запроса из таблицы **Студенты**.

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.

3.5. 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 НомерЗачетки, ФИО и БаллыЕГЭ из таблицы Студенты.**

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

В общем случае поля, вводимые в набор 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 (заголовков 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**.

3.6. 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 таблице **Студенты** будут выбраны студенты с проходным баллом выше 100.

3.7. Виды критериев

Для создания запроса с несколькими критериями пользуются различными операторами.

Можно задать несколько условий отбора, соединенных логическим оператором ИЛИ (or) или И (and), 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 200 or 250 соответствует тому, что будут выбраны ФИО студентов с проходным баллом 200 или 250.

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

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

При такой записи условия также будут выбраны студенты с проходным баллом 120 или 123.

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

>200 and <250

будут выбраны студенты с проходным баллом в соответствующем диапазоне.

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 студентов, проживающих в городе Москва с проходным баллом 120.

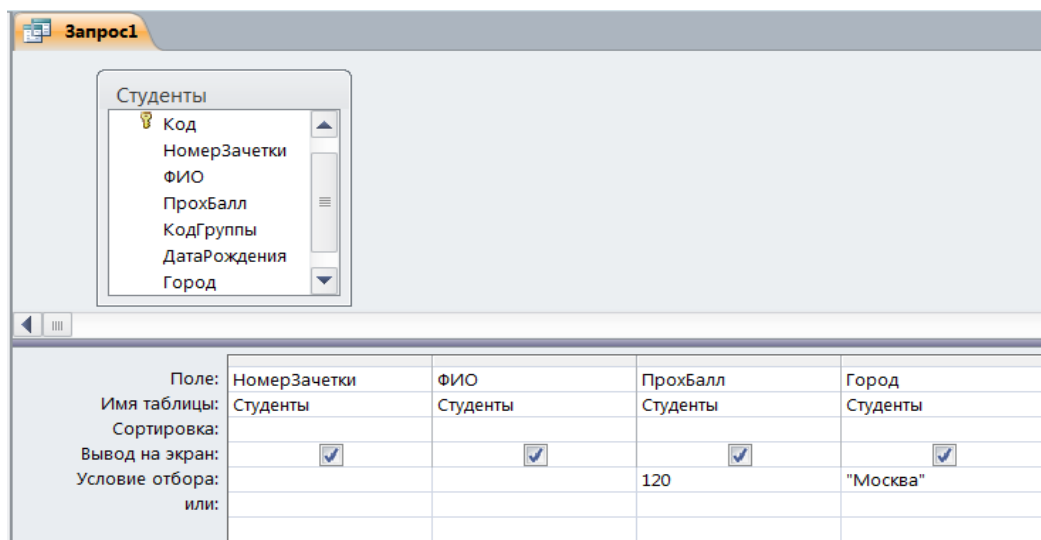


Рис 5. Пример записи условия с использованием оператора *и (and)*

Исключить группу данных из состава анализируемых записей позволяет следующий критерий

< >150

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

3.8. 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:

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

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

3.9. 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.

3.10. 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.

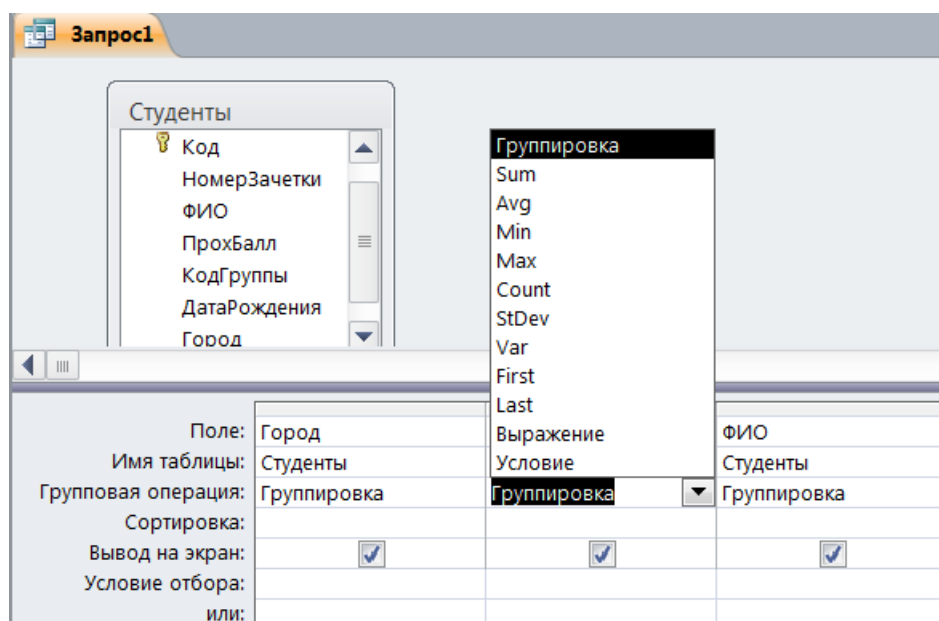


Рис 6. Строка *Групповая операция* в бланке QBE

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 Null (от пустых);

FIRST - определяет первое значение в указанном поле записей;

LAST - определяет последнее значение в указанном поле записей.

3.11. 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).

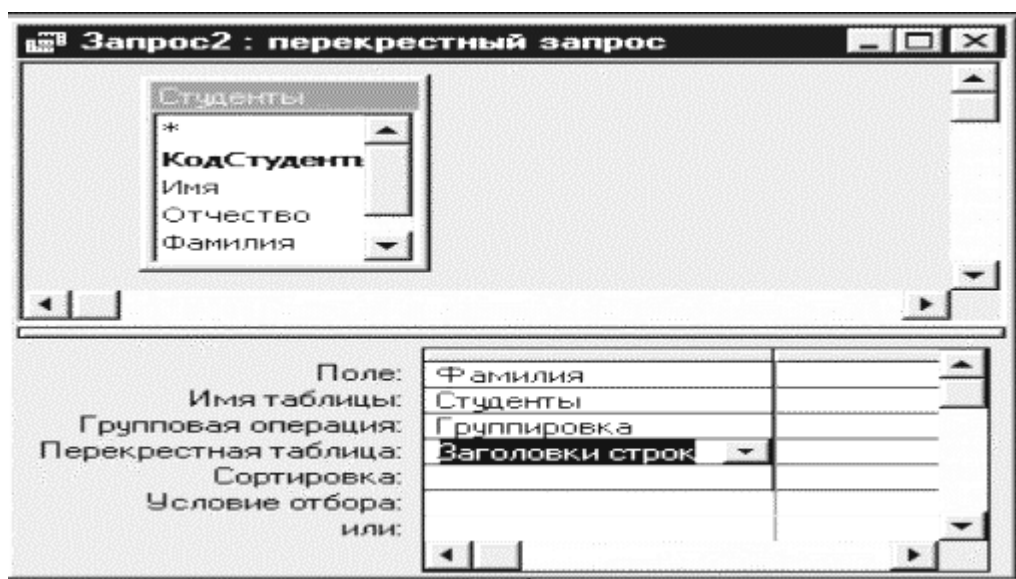


Рис.5. Вид окна перекрестного запроса

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.

3.12. 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).

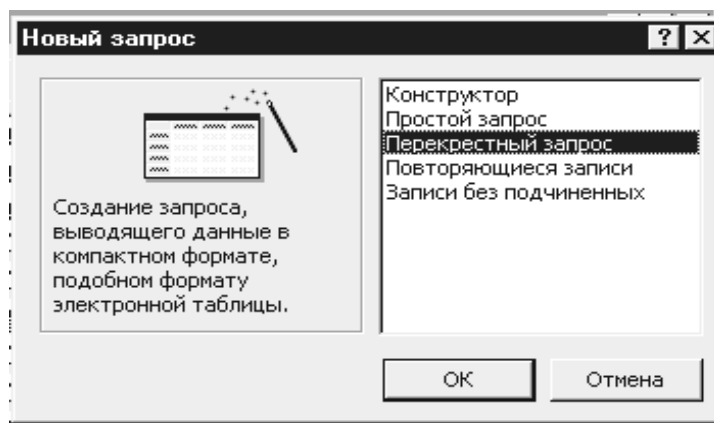


Рис. 6. Вид окна диалога «Новый запрос»

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.

3.13. Выполнение запроса

Готовый запрос выполняется после щелчка по кнопке  панели инструментов в режиме Конструктора запросов или при активизации команды **Запуск** из меню **Запрос**. (В Access 2010 в пункте меню **Конструктор** имеется эта же кнопка). В результате будет получена таблица с ответом на заданные условия. Например, на рис. 7 показан результат запроса, построенного на рис. 5.

Запрос1			
Номер зачет	ФИО	Проходной	Город
9876543210	Сидоров С.С.	120	Москва
*			Москва

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.

3.14. Порядок выполнения работы

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

1. Создать следующие однотабличные запросы:

- получить список деканов и наименований факультетов;
- вычислить ФИО студента с максимальным значением балла ЕГЭ;
- выбрать ФИО студентов, у которых фамилия начинается с определенной буквы;
- получить данные конкретного студента, ФИО которого вводится с клавиатуры;
- выбрать ФИО студентов, дата рождения которых до определенной вводимой даты.

2. Создать следующие многотабличные запросы:

- выбрать номер зачетки, фамилию и инициалы (ФИО), дату рождения и стипендию студентов конкретной группы;
- выбрать студентов конкретной группы, у которых балл ЕГЭ меньше 200;
- выбрать ФИО студентов, дата рождения которых в интервале значений и учащихся на определенном факультете;
- выбрать список номеров групп одного конкретного факультета;
- выбрать список номеров групп для одного факультета с использованием параметрического запроса;
- выбрать ФИО студентов, проживающих в одном из городов или обучающихся на одном из факультетов;
- выбрать ФИО и номер группы студента с наименьшим значением балла ЕГЭ.

3. Создать следующие запросы с вычисляемыми полями:

- вычислить возраст всех студентов одной группы (использовать функцию DateDiff);
- используя построитель выражений, подсчитать надбавку всем студентам, равную 15% от стипендии;
- используя построитель выражений, подсчитать надбавку 15% от стипендии студентам, которые имеют только отличные оценки.

4. Создать следующие запросы с групповыми функциями:

- вычислить среднюю оценку по экзаменам для каждого студента;
- вычислить средний балл ЕГЭ каждой группы;
- вычислить число студентов и средний балл ЕГЭ (в одном запросе) для студентов, у которых средний балл ЕГЭ выше 250;

- вычислить количество студентов каждой группы;
- вычислить среднюю стипендию студентов, оценки которых за экзамен по информатике 4 или 5.
- вычислить номер группы, у которой средний балл ЕГЭ наименьший.

5. Создать перекрестные запросы:

- Создайте перекрестный запрос с помощью **Мастера запросов** на базе таблицы **Студенты**, выбрав в качестве столбцов **КодГруппы**, в качестве строк **Город**, а вычисляемое поле – **НомерЗачетки** и используя для подсчета функцию **Count** (Число для Access 2010).
- Сохраните запрос под именем **Распределение студентов по городам**; просмотрите запрос в режиме конструктора; недостатком этого запроса является использование в заголовках столбцов кодов групп, а не номеров. Причина в том, что **Мастер запросов** может выбрать поля только из одной таблицы. Если нужны поля из разных таблиц (а поле **Номер группы** находится в таблице **Группы**) при использовании **Мастера запросов** необходим отдельный запрос с разными полями, на основе которого и будет построена перекрестная таблица.
- Построим нужный запрос с использованием номеров групп без использования **Мастера запросов** в **Конструкторе запросов**. Для этого:
 - Выполним **Создать/Конструктор запросов**;
 - Выберем две таблицы: **Студенты** и **Группы**;
 - Выполним **Работа с запросами/Перекрестный**; в качестве заголовков строк – **НомерГруппы**, столбцов – **Город**, вычисляемое поле с помощью функции **Count**, итог – функция **Count** по полю **НомерЗачетки** (Рис. 8).
 - Сохраните запрос под именем **Количество студентов по городам и группам**.

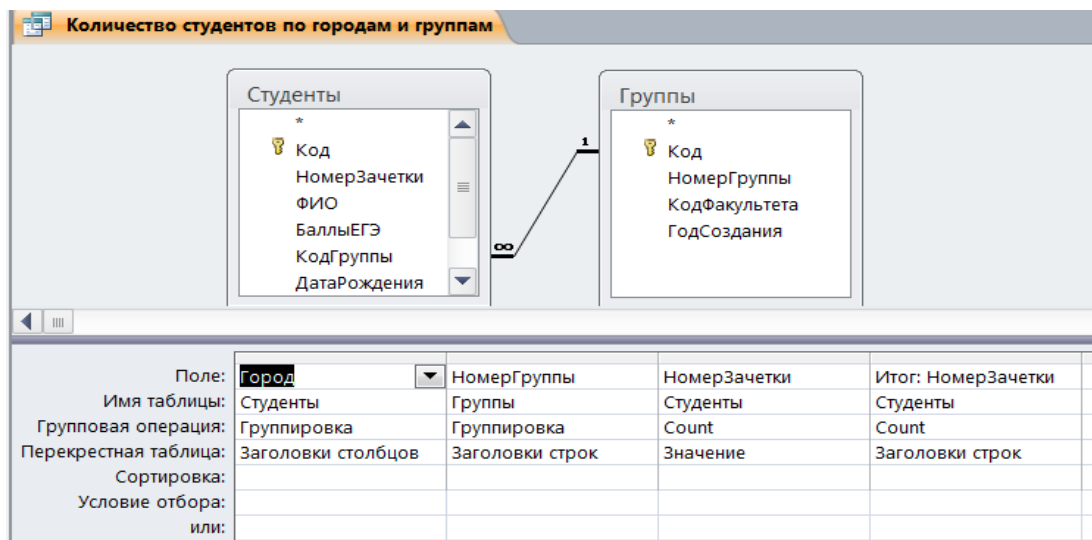


Рис. 8. Перекрестный запрос

- Создайте перекрестный запрос на базе простого запроса, для чего:
 - создайте простой запрос, включив в него поля **НаименованиеФак** из таблицы **Факультет**, **Город** и **НомерЗачетки** из таблицы **Студенты**. Имя запроса: **Факультет Запрос**.
 - Создайте перекрестный запрос с помощью Мастера запросов на базе запроса **ФакультетЗапрос**; поле **НаименованиеФак** выберите в качестве заголовка строк,

поле **Город** - заголовка столбцов, а для подсчета количества студентов используйте функцию **Count** по вычисляемому полю **НомерЗачетки** (функцию **Число** для Access 2010). Выполните запрос и проанализируйте результат.

3.15. Содержание отчета

1. Тема и цель работы.
2. Краткое описание последовательности выполнения.

3.16. 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 построителя 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 4

ТЕМА: Работа с данными с помощью запросов-действий.

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

4.2. 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 таблицы, воспользовавшись кнопкой **Вид**.

4.3. Запросы на создание таблицы

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

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 в том случае, если какие-либо данные нужно хранить в течение длительного времени, да и просто как средство создания резервной копии таблицы.

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).

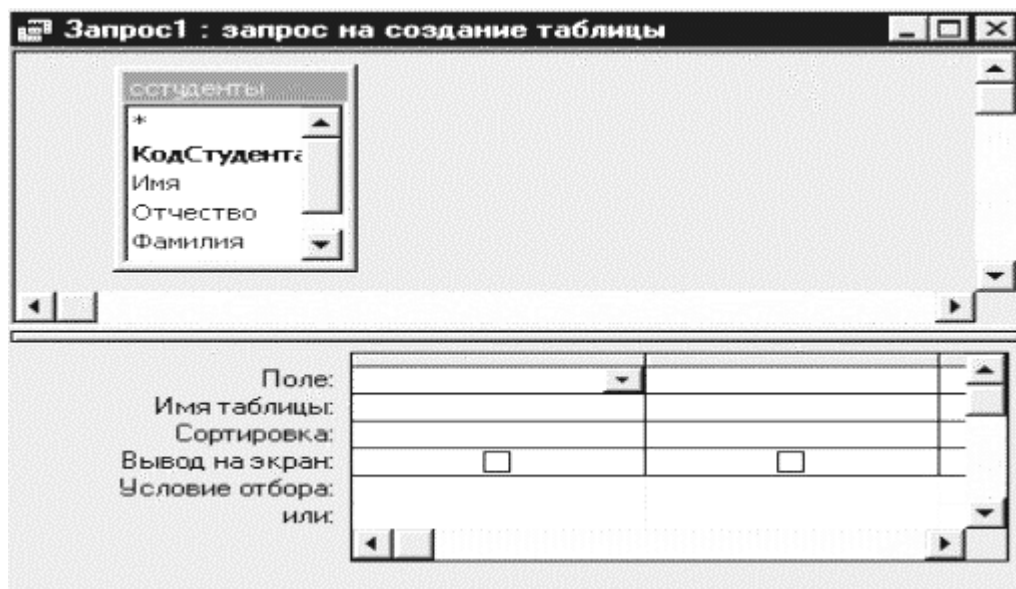


Рис. 1. Вид окна запроса на создание таблицы

4.4. 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).

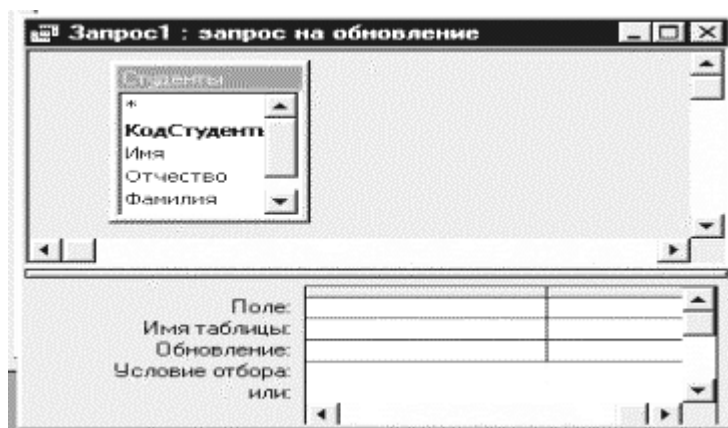


Рис.2. Вид окна запроса на обновление записей

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

4.5. 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).

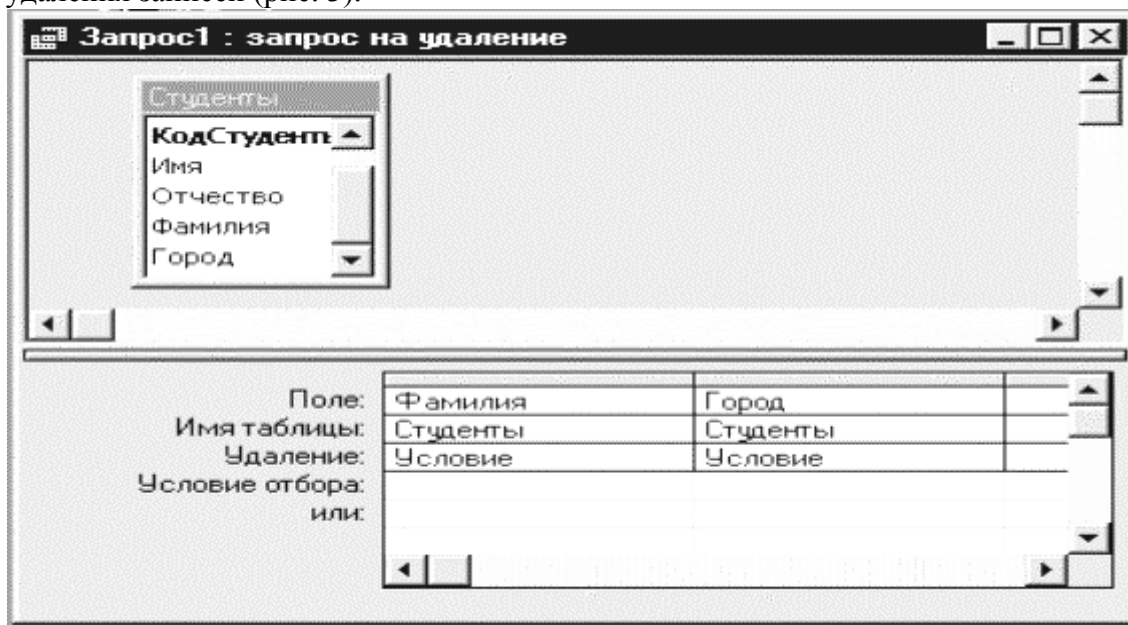


Рис. 3. Вид окна запроса на удаление записей

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.

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.

4.6. 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).

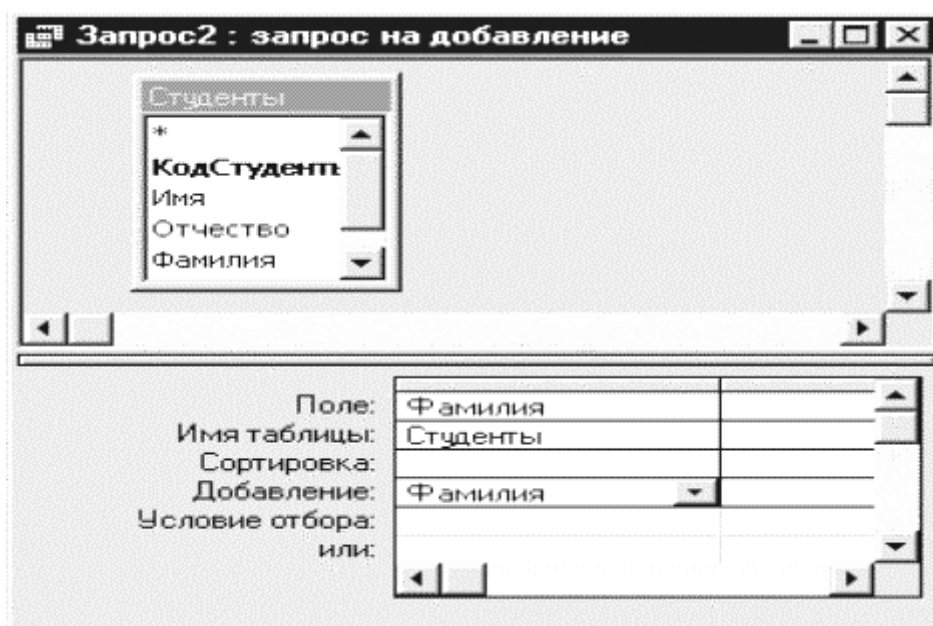


Рис. 4. Вид окна запроса на добавление записей

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.

4.7. Порядок выполнения работы

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

Разработка запросов на создание таблицы

1. Сконструируйте запрос-выборку к таблице **Студенты**, выбрав тех, кто не получает стипендии. Выполните запрос и просмотрите результат.
2. Перейдите в режим Конструктора 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**. Просмотрите полученную таблицу
3. Создайте простой запрос, в который включим все поля таблицы **Оценки**. В строке **Условие отбора** в **КодДисциплины** таблицы **Оценки** поставим код требуемой дисциплины (например, вычислительной математики). Т. е. это запрос будет выводить только экзамены по данной дисциплине. В группе **Тип запроса** выберем **Создание таблицы** и зададим имя создаваемой таблицы, например, **ОценкиПоВычмату**. Выполним этот запрос и посмотрим созданную таблицу.

Разработка запроса на удаление записей

Нами были сохранены записи об экзаменах по вычислительной математике в таблице **ОценкиПоВычмату**), но из исходной таблицы они не были удалены. **Теперь удалим эти записи из исходной таблицы.**

Создадим запрос-выборку к таблице **Оценки** для получения списка экзаменов по вычислительной математике. Преобразуем его на запрос-удаление и выполним его. Убедимся, что из таблицы эти записи удалены!

Разработка запроса на добавление

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

1. Создадим запрос, в который включим все поля таблицы **ОценкиПоВычмату**. Затем выполним команду **Добавление**.
2. В окне **Добавление** выберем таблицу, в которую будут добавляться соответствующие записи - **Оценки**. Выполним запрос, при этом выводится сообщение о количестве добавляемых записей и предлагается подтвердить запрос. Убедимся, что записи добавились в таблицу **Оценки**.

Разработка запроса на обновление записей

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

2. Перейдите в режим **Конструктора** для этого запроса и преобразуйте его в запрос на обновление данных (например, для Access 2010, выбрав **Обновление** в меню **Работа с запросами**). В строке **Обновление** задайте новое значение для поля **Город** (например, перепишите его с заглавной буквы или введите сокращенное название города и т.п.).

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

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

4.8. Содержание отчета

Отчет о проделанной работе должен содержать:

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

4.9. 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 5

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

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

5.2. 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.

5.3. Задание

1. Просмотреть п. 5.4. 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. 5.5.) 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 таблицы, соответствующие сущностям ER- модели БД;
 - создать схему БД, определив две связи типа «один-ко-многим» между двумя сущностями-таблицами и таблицей-связкой;
 - создать форму для ввода данных в БД и с ее помощью ввести записи в БД;
 - 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;
 - просмотреть запросы в режиме SQL.

5.4. 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).

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

БД должна содержать список студентов, сдаваемых предметов (дисциплин) и результаты экзаменов каждого студента.

Данные о каждом студенте должны содержать номер зачетки, фамилию, имя, отчество, название факультета, курс, номер группы. Данные о предметах: название предмета, часы.

Сущности:

Опишем сущности на языке инфологического моделирования:

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

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

ПРЕДМЕТЫ (Код предмета, *Наименование, Часы*)

Ключи:

В сущности **СТУДЕНТЫ** ключевой атрибут **Код студента (РК)**. Такой ключ будет типа “Счетчик” (см. п. 1.7 лабораторной работы №1).

В таблице **Предметы** ключевой атрибут – **Код предмета (РК)**.

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».

Также один предмет может сдаваться разными студентами, поэтому связь между сущностями **Предметы** 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 **Код студента** сущности **СТУДЕНТЫ** должен присутствовать как внешний (FK) в сущности **ЭКЗАМЕНЫ**. Класс принадлежности со стороны сущности **Предметы** также обязательный, поэтому ключ **Код предмета** должен присутствовать как внешний (FK) в сущности **Экзамены**. Таким образом, сущность **ЭКЗАМЕНЫ** на языке инфологического моделирования должна быть следующей:

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

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



Рис. 5.3-1

5.6. Варианты заданий

Таблица 5.1.

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

5	Сеть магазинов	ВЛАДЕЛЕЦ (<u>код владельца</u> , ФИО, адрес, телефон владельца магазина). ВЛАДЕНИЕ (<u>код владельца</u> , <u>код магазина</u> , дата регистрации). МАГАЗИН (<u>код магазина</u> , название, район расположения 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>код автомеханика</u> , ФИО). НАРЯД ((<u>код автомобиля</u> , (<u>код автомеханика</u> , дата обслуживания). АВТОМОБИЛЬ (<u>код автомобиля</u> , регистрационный знак, 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> , номер группы, 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>код работы</u> , название, трудоемкость). ПОРУЧЕНИЕ (<u>код работы</u> , (<u>код сотрудника</u> , плановая и реальная даты окончаний). СОТРУДНИК (<u>код сотрудника</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>код пациента</u> , ФИО, дата рождения, социальный статус) ЛЕЧЕНИЕ (<u>код пациента</u> , <u>код врача</u> , заболевание, срок потери трудоспособности, дата начала лечения). ВРАЧ (<u>код 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>код вида спорта</u> , название, единица результатов, мировой рекорд и его дата). РЕЗУЛЬТАТ (<u>код вида спорта</u> , <u>код спортсмена</u> , 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 виду спорта “бег”.
11	Поставки продукции	ПРЕДПРИЯТИЕ (<u>код предприятия</u> , наименование, дата регистрации, вид собственности, число работников). 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>код вида транспорта</u> , название, средняя скорость движения, количество машин в парке, стоимость проезда). МАРШРУТ (<u>код пути</u> , <u>код вида транспорта</u> , количество остановок на маршруте, количество машин на маршруте). ПУТЬ (<u>код пути</u> , начальный пункт, 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	Население стран	<p>СТРАНА (<u>код страны</u>, название, регион, столица, площадь территории).</p> <p>НАСЕЛЕНИЕ <u>код страны</u>, <u>код национальности</u>, количество людей данной национальности).</p> <p>НАЦИОНАЛЬНОСТЬ (<u>код национальности</u>, название, язык, общая численность).</p>	В одной и той же стране могут быть люди различных национальностей.	<p>1.Определить численность населения по всем странам.</p> <p>2.Определить столицу той страны, где проживает более всего представителей конкретной национальности.</p>
14	Домо-управление	<p>КВАРТИРА (<u>код квартиры</u>, номер квартиры, номер дома, число жильцов, площадь).</p> <p>ОПЛАТА (<u>код квартиры</u>, <u>код вида оплаты</u>, сумма оплаты, дата оплаты).</p> <p>ВИД ОПЛАТЫ (<u>код вида оплаты</u>, наименование, цена за единицу площади, цена за одного жильца).</p>	В одной квартире используются разные виды оплаты.	<p>1.Определить сумму месячной оплаты для данной квартиры конкретного дома.</p> <p>2.Определить задолженность по оплате конкретного вида конкретной квартиры.</p>
15	Аэропорт	<p>САМОЛЕТ (<u>код самолета</u>, тип, число мест, скорость полета).</p> <p>РЕЙС (<u>код самолета</u>, <u>код маршрута</u>, дата и время вылета, дата и время прилета, количество проданных билетов).</p> <p>МАРШРУТ <u>код маршрута</u>, расстояние, пункт вылета, пункт назначения).</p>	Один самолет может летать на разных маршрутах и по одному маршруту могут летать разные самолеты.	<p>1.Определить время полета для конкретного самолета по конкретному маршруту.</p> <p>2.Выбрать тип самолета, который чаще всего летает по конкретному маршруту.</p>
16	Компьютеры	<p>ФИРМА (<u>код фирмы</u>, название, место размещения).</p> <p>РЫНОЧНОЕ ПРЕДЛОЖЕНИЕ (<u>код фирмы</u>, <u>код конфигурации</u>, объем партии, цена партии).</p> <p>КОНФИГУРАЦИЯ ПК (<u>код конфигурации</u>, тип процессора, тактовая частота, объем ОЗУ, объем жесткого диска, дата выпуска).</p>	Одна конфигурация ПК может изготавливаться разными фирмами и одна и та же фирма может собирать ПК разной конфигурации.	<p>1.Определить фирму, которая представляет самую новую модель на базе конкретного типа процессора.</p> <p>2.Выбрать модель с наибольшей тактовой частотой, которая выпускается в конкретном городе.</p>

17	Информация для зрителей	<p>ФИЛЬМЫ (<u>код фильма</u>, название, год выпуска, категория проката)</p> <p>КИНОПРОКАТ (<u>код фильма</u>, <u>код кинотеатра</u>, начало показа, окончание показа)</p> <p>КИНОТЕАТРЫ (<u>код кинотеатра</u>, наименование, адрес, вместимость)</p>	Один и тот же фильм может идти в разных кинотеатрах.	<p>1.Определить количество фильмов за определенный период в данном кинотеатре.</p> <p>2.Определить кинотеатр наибольшей вместимости, в котором демонстрируется данный фильм.</p>
18	Зоопарк	<p>ВИД ЖИВОТНОГО (<u>код вида животного</u>, название, суточное потребление корма, семейство, континент обитания).</p> <p>РАЗМЕЩЕНИЕ (<u>код вида животного</u>, <u>код помещения</u>, количество животных)</p> <p>ПОМЕЩЕНИЕ (<u>код помещения</u>, номер вольера, наличие водоема, наличие отопления).</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>код корабля</u>, название, водоизмещение, капитан).</p> <p>ПОСЕЩЕНИЕ ПОРТА (<u>код корабля</u>, <u>код порта</u>, дата прибытия, дата убытия, номер причала, цель посещения).</p> <p>ПОРТ (<u>код порта</u>, название, страна).</p>	Один корабль может посещать несколько портов.	<p>1.Выбрать список кораблей, посещавших конкретный порт в конкретное время.</p> <p>2.Определить страны, в которые никогда не приходят корабли с целью 'туризм'.</p>

21	Научные конференции	<p>УЧЕНЫЙ (<u>код ученого</u>, фамилия и имя, организация, страна, ученая степень).</p> <p>УЧАСТИЕ (<u>код ученого</u>, <u>код конференции</u>, тип участия, тема доклада, сообщения и пр., публикация (да/нет)).</p> <p>КОНФЕРЕНЦИЯ (<u>код конференции</u>, название, 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>код программного продукта</u>, название, версия, фирма, дата выпуска, прикладная область, стоимость лицензии).</p> <p>ИНСТАЛЛЯЦИЯ (<u>код программного продукта</u>, <u>код пользователя</u>, стоимость инсталляции, дата инсталляции, количество лицензий при инсталляции).</p> <p>ПОЛЬЗОВАТЕЛЬ (<u>код пользователя</u>, наименование, 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><u>ПРОЦЕСС</u> (<u>код процесса</u>, название процесса, приоритет, класс).</p> <p><u>ПЛАНИРОВАНИЕ</u> (<u>код процесса</u>, <u>код ресурса</u>, запрошено/выделено).</p> <p><u>РЕСУРС</u> (<u>код ресурса</u>, название ресурса, количество, цена за единицу).</p> <p><u>Примечание:</u></p> <p><u>Процессы:</u> служба ОС, драйвер, процесс-подсистема клиент/сервер, процесс авторизации, System-процесс, диспетчер задач и др.</p> <p><u>Ресурсы:</u> оперативная память, жесткий диск, внешняя память.</p>	<p>Один и тот же процесс может задействовать много разных ресурсов и один и тот же ресурс может быть задействован разными процессами.</p> <p><u>Классы процессов:</u> критический, серверный, нормальный.</p>	<p>1. Вывести данные по ресурсам для каждого процесса.</p> <p>2. Определить ресурс, у которого "самые большие аппетиты" в ценовом выражении.</p>
24	Добыча полезных ископаемых	<p><u>ПОЛЕЗНОЕ ИСКОПАЕМОЕ</u> (<u>код полезного ископаемого</u>, наименование, единица измерения, годовая потребность, цена за единицу).</p> <p><u>ВЫВОЗ</u> (<u>код полезного ископаемого</u>, <u>код месторождения</u>, объем вывоза)</p> <p><u>МЕСТОРОЖДЕНИЕ</u> (<u>код месторождения</u>, название месторождения, запасы)</p>	<p>В одном и том же месторождении могут добываться различные ископаемые.</p>	<p>1. Определить, какие ископаемые вывозятся из конкретного месторождения.</p> <p>2. Выбрать список месторождений, из которых вывозится конкретное ископаемое.</p>
25	Автотранспортное предприятие	<p><u>АВТОМОБИЛЬ</u> (<u>код автомобиля</u>, регистрационный знак, марка, пробег, расход топлива).</p> <p><u>РЕЙС</u> (<u>код автомобиля</u>, <u>код водителя</u>, дата выезда, дата прибытия, расстояние).</p> <p><u>ВОДИТЕЛЬ</u> (<u>код 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>код актера</u> , ФИО, звание, амплуа, пол). ПОСТАНОВКА НА РОЛЬ (<u>код актера</u> , <u>код роли</u> , дата назначения на роль, режиссер); РОЛЬ (<u>код роли</u> , действующее лицо, тип роли, название пьесы).	На одну и ту же роль могут назначаться разные актеры. Амплуа: герой-любовник, инженеру, 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>код лекарства</u> , название, показания к использованию, противопоказания, производитель). НАЛИЧИЕ (<u>код лекарства</u> , <u>код аптеки</u> , тип, цена, количество, срок годности). АПТЕКА (<u>код аптеки</u> , название 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>код блюда</u> , название блюда, категория, вес порции). СОСТАВ (<u>код блюда</u> , <u>код продукта</u> , количество, способ обработки). ПРОДУКТ (<u>код продукта</u> , название продукта, 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 заданному блюду.
29	Изучение студентами дисциплин по выбору	СТУДЕНТ (<u>код студента</u> , номер зачетной книжки, ФИО студента, адрес, телефон). ЭКЗАМЕН (<u>код студента</u> , <u>код дисциплины</u> , оценка). ДИСЦИПЛИНА (<u>код дисциплины</u> , наименование, количество лекционных, семинарских и лабораторных занятий).	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>код сотрудника</u> , ФИО, образование, телефон, дата рождения). ЗАЧИСЛЕНИЕ (<u>код сотрудника</u> , <u>код подразделения</u> , должность, дата зачисления, оклад). ПОДРАЗДЕЛЕНИЕ (<u>код подразделения</u> , наименование, 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 6

ТЕМА: СОЗДАНИЕ ОТЧЕТОВ

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

6.2. 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 выбрать имя источника данных в поле со списком в нижней части окна диалога.

6.3. 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.

6.4. 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.

6.5. 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.

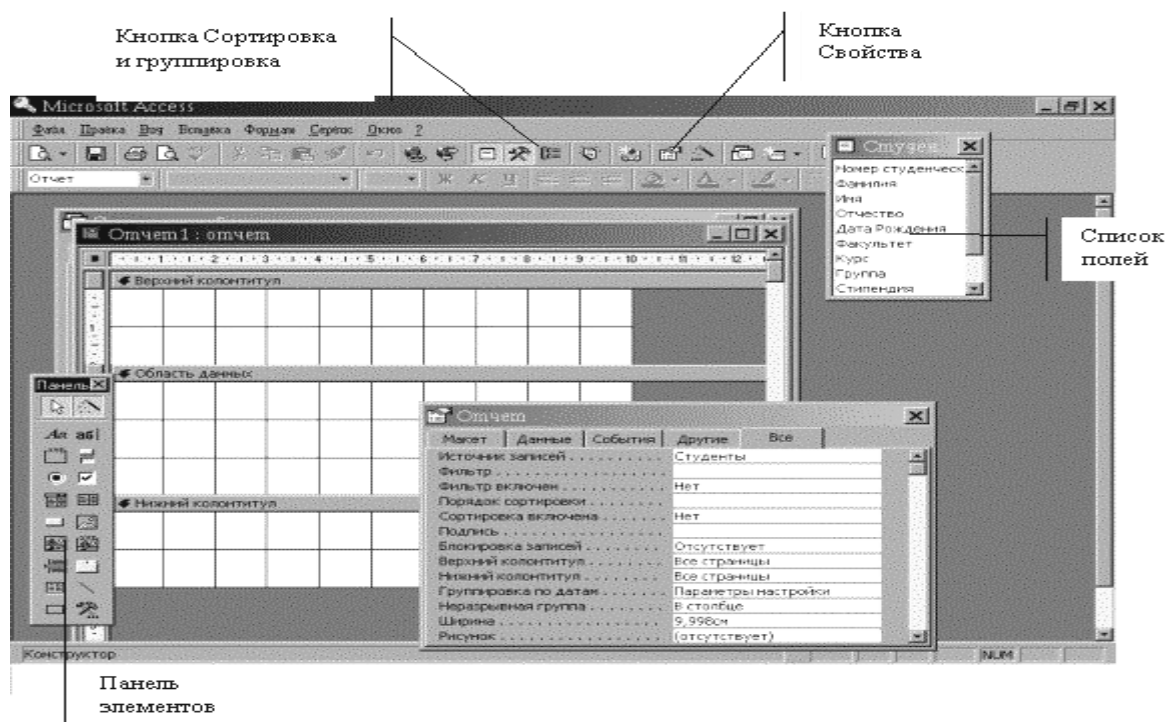


Рис. 1. Окно отчета

6.6. 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.

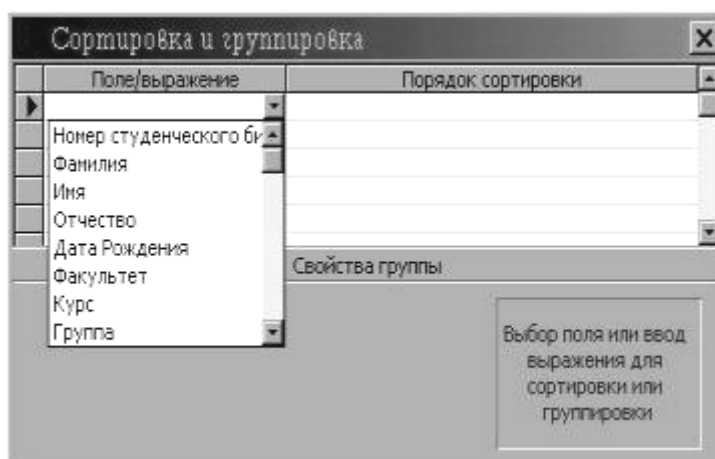


Рис 2. Окно Сортировка и группировка

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**.

6.7. 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.

6.8. 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.

6.9. 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]

6.10. 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.

6.11. 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.

6.12. Объединение текстовых значений и скрывание 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.

6.13. 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.

6.14. 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**.

6.15. 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.

6.16. Порядок выполнения работы

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

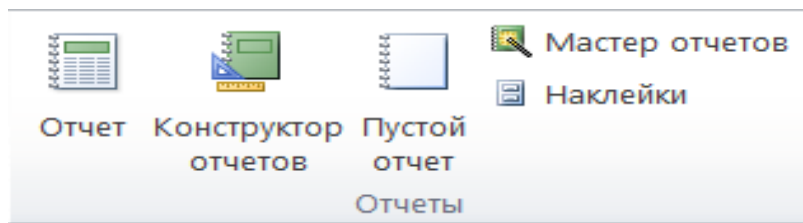


Рис.

- **Отчёты** - создание простого отчета на основе данных из текущего запроса или таблицы;
 - **Конструктор** - позволяет создавать отчет с нуля на бланке отчета;
 - **Пустой отчет** - создание пустого отчета, в который можно вставить новые поля;
 - **Мастер отчетов** - запуск мастера помогающего настроить отчет при его создании;
 - **Наклейки** - предназначены для наклеивания на конверты при рассылке писем адресатам, информация о которых хранится в базе данных.
1. Откройте базу данных **Успеваемость Студентов**.
 2. Составьте отчет № 1 с результатами экзаменов по разным дисциплинам. Макет будущего отчета представлен на рис. 6.16-1.

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

Рис. 6.16-1. Макет отчета №1

- Выполните **Создание/Мастер отчетов**;
- Последовательно выберите из таблицы **Дисциплины** поле **НазваниеДисц**, из таблицы **Группы** поле **НомерГруппы**, из таблицы **Студенты** поле **ФИО** и из таблицы **Оценки** поле **Оценка**;
- На втором диалоге, который появляется только в случае, если для отчета выбрано несколько таблиц, можно установить группировку. Для нашего случая выберем таблицу **Дисциплины**;
- На следующем шаге добавьте для группировки поле **НомерГруппы**;
- Выберите порядок сортировки по полю **ФИО**, по кнопке **Итоги** - вычисление среднего (**Avg**) по полю **Оценки**;
- Выберите вид макета для отчета ступенчатый;
- Сохраните отчет под именем **Результаты экзаменов**.
- Просмотрите отчет на экране, анализируя его содержимое (группировка записей, вид итогов, элементы страницы отчета и т.д.). Очевидно, что полученный отчет необходимо доработать, чтобы он принял вид как на рис. 5.16-1.
- Перейдите в режим **Конструктора** данного отчета (нажать кнопку **Конструктор** на панели инструментов) и произведите в отчете следующие изменения:
- Разместите соответствующие Надписи и Поля во всех разделах отчета;
- В полях раздела **Примечание группы** “Номер группы”, **Примечание Группы** “Код” в свойствах выберите **Тип границы – Отсутствует**;
- Отредактируйте соответствующие подписи. Необходимо подогнать поля отчета в соответствии с образцом.

- Удалите в **Примечании группы “Код”** Итоги для Код и т.д.
 - Отредактируйте отчет, просмотрите и сохраните.
3. Создайте отчет № 2 на основе запроса.
- Создайте базовый запрос к таблицам **Студенты – Оценки - Дисциплины**, включив в него поля **НомерЗачетки, ФИО, НазваниеДисц, Оценка**. Сохраните запрос под именем **Данные для отчета по успеваемости**;
 - В качестве режима создания выберите **Мастер отчетов**;
 - На первом шаге **Мастера отчетов** выберите все поля исходного запроса для включения в отчет;
 - На втором шаге выберите тип представления данных - по таблице **Студенты**;
 - На третьем шаге добавьте уровень группировки по полю **НомерЗачетки**;
 - На четвертом шаге укажите сортировку по полю **НазваниеДисц** и вид итогов - Среднее (**Avg**) по полю **Оценка**;
 - На последующих шагах Мастера задайте стиль оформления отчета и его название **Результаты экзаменов студентов**.
 - Просмотрите отчет на экране, проанализируйте его содержание (группировка, итоговые поля, области отчета). Отредактируйте отчет, следует подогнать поля отчета.
 - Перейдите в режим **Конструктора** для данного отчета и отредактируйте его так, чтобы в примечании группы заголовков был таким: **Средний балл за текущую сессию**;
 - Добавьте 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 $=\text{If}([\text{Avg_Оценка}]>5;150;\text{If}([\text{Avg_Оценка}]>4 \text{ Or } [\text{Avg_Оценка}]=4;100;50))$, т.е. надбавка начисляется в зависимости от результатов сессии.
 - Добавьте в базовый запрос **Данные для отчета по успеваемости** поле **Стипендия**.
 - Добавьте новое поле **Удержания**, учитывая, что удерживается 10% от значения Стипендии. Выполните запрос и просмотрите результат.
4. Создайте отчет №3 в соответствии с новым запросом, включив в него поле **Удержания**. Отредактируйте отчет.
5. Создайте отчет №4 по двум таблицам с включением в базовый отчет подчиненный отчет. Вид макета будущего отчета представлен на рис. 5.16-2. В отчет предполагается выводить данные по каждой группе. Проект макета дает основание выбрать в качестве основной таблицы отчета таблицу **Группы**, а таблицу **Студент** – в качестве источника данных для подчиненного отчета со списком студентов.
- Выберем таблицу **Группы** и выполним **Создание/Отчет**. Перейдем в режим конструктора для данного отчета.

СОСТАВ ГРУПП

ГРУППА

Студенты		
ФИО	Дата рождения	Стипендия
Иванов И.И.	15.04.1998	1 111р.
Петров П.П.	03.05.1998	1 200р.

ГРУППА

Студенты		
ФИО	Дата рождения	Стипендия
Сидоров С.С.	30.08.1999	1 000р.
Кузнецов А.И.	30.08.1997	0р.

ГРУППА

Студенты		
ФИО	Дата рождения	Стипендия
Федосеев И.Т.	20.09.1997	0р.

Рис. 6.16-2. Макет отчета №3

- Очистите **Верхний колонтитул** от всех полей и надписей, а в **Области данных** оставьте только поле и надпись для **Номера группы**. Сохраните отчет под именем **Состав групп**.
- Создайте подчиненный отчет в **Мастере отчетов**. В качестве источника данных выберите таблицу **Студенты**.
- В подчиненном отчете в области данных должны быть включены поля **ФИО**, **ДатаРождения** и **Стипендия**, а в заголовок из верхнего колонтитула перенесены соответствующие надписи. Для этого расширьте раздел заголовка отчета, выделите все подписи полей в верхнем колонтитуле и перетащите их. Поскольку отчет будет использоваться только как подчиненный, можно удалить оба колонтитула. Для этого нажмите правой кнопкой мыши на **Верхний** или **Нижний Колонтитул** и выберите **Колонтитулы страницы**. После этого в открывшемся окне нажимаем ДА (Рис. 5.16-3).

Заголовок отчета													
Студенты													
ФИО				Дата рождения				Стипендия					
Область данных													
ФИО				ДатаРождения				Стипендия					
Примечание отчета													

Рис. 6.16-3. Часть отчета после редактирования

- Сохраните подчиненный отчет под именем **Студенты**.
- Откройте отчет **Состав групп** и перетяните в его область данных подчиненный отчет **Студенты**. Удалите надпись подчиненного отчета.
- Просмотрите отчет **Состав групп** в режиме **Макета** или **Предварительного просмотра**.
- Отредактируйте отчет в соответствии с рис.6.16-2 .

6.17. Содержание отчета

1. Название работы, ее задачи и описание последовательности выполнения.
2. Ответы на контрольные вопросы.

6.18. 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. Чем характеризуется режим **Конструктора**?
6. Назовите разделы отчета.
7. Назовите инструменты окна отчета.
8. Как добавить вычисляемое поле в область данных?
9. Наличие каких объектов необходимо для внедрения подчиненного отчета?
10. Какие изменения можно внести в отчет в режиме **Конструктора**?

Лабораторная работа 7

ТЕМА: Использование макросов в Access

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

7.2. 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.

7.3. 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.

7.4. 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.

7.5. 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.

7.6. 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.

7.7. 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.

7.8. Создание кнопки с помощью мыши

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

7.9. 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.

7.10. 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.

7.11. Порядок выполнения работы

Создание изолированного макроса

1. На вкладке **Создание** в группе **Макросы и код** щелкните **Макрос**. Откроется конструктор макросов.
2. Для добавления макрокоманды в макрос в конструкторе макросов выберите макрос **ОкноСообщения** из выпадающего меню **Добавить новую макрокоманду**.
3. Укажите следующие аргументы для макрокоманды (Рис. 7.-1):

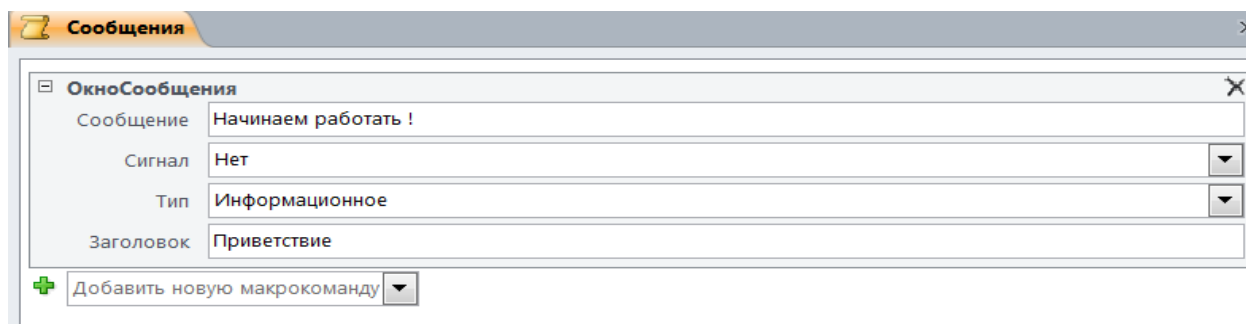


Рис. 7.1 Аргументы макрокоманды

4. Выполните макрос с помощью кнопки **Выполнить**. Подтвердите сохранение макроса, присвоив ему имя **Сообщения**.
5. Переименуйте макрос, дав ему имя **Autoexec**.
6. Закройте базу данных, затем снова откройте ее и убедитесь, что теперь макрос выполняется автоматически при открытии базы.
7. Верните макросу прежнее имя.

Работа с вложенными макросами

1. Создание вложенных макросов
 - 1.1. Откройте макрос **Сообщения** в режиме конструктора.
 - 1.2. Добавьте **Вложенный макрос** из каталога макрокоманд.
 - 1.3. Присвойте имя вложенному макросу **Окончание** и добавьте в окно вложенного макроса макрокоманду **ОкноСообщения** со следующими аргументами:

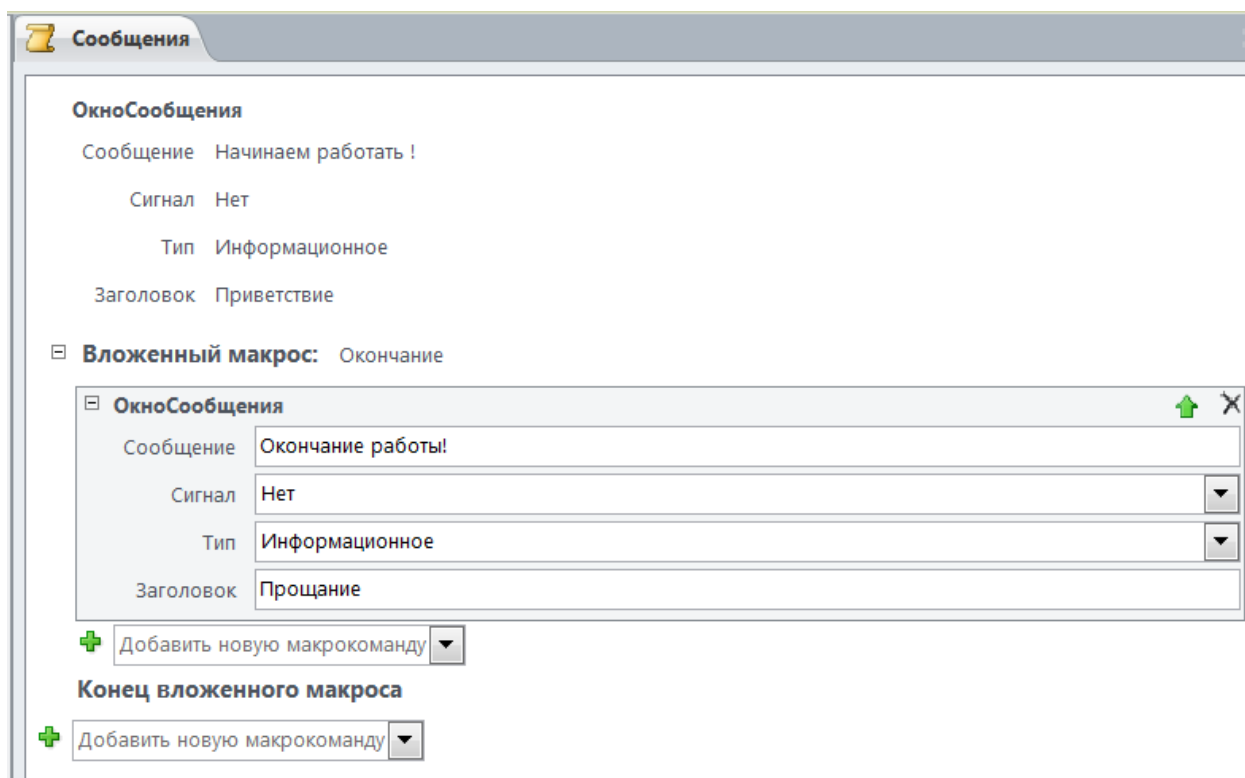


Рис. 7.2. Вложенный макрос

- 1.4. Сохраните изменения и закройте окно макроса.
 - 1.5. Выполните макрос из окна базы данных. Будет выполнен только первый макрос из созданных.
2. Связь группы макросов с формой (работа с формой **Дисциплины**).
 - 2.1. Откройте форму **Дисциплины** в режиме конструктора.
 - 2.2. Откройте окно свойств формы (**Страница свойств**) в группе **Сервис**, выберите вкладку **Свойств формы - События**.
 - 2.3. Для свойства **Открытие** выберите из списка макрос **Сообщения**.
 - 2.4. Для свойства **Заккрытие** выберите макрос **Сообщения.Окончание**.

2.5. Сохраните и закройте форму. Затем откройте форму **Дисциплины** (выполнится основной макрос **Сообщения**, а потом закройте (выполнится вложенный макрос **Сообщения.Окончание**)).

Создание внедренного макроса

1. Откройте отчет **Состав групп** в режиме конструктора.
2. Разместите на форме отчета элемент управления **Кнопка**.
3. Откройте **Окно свойств** для элемента управления **Кнопка**, вкладку **События**, событие **Нажатие кнопки** и выберите **Построитель** (нажав кнопку с троеточием). В открывшемся диалоговом окне **Построителя** выберите элемент **Макросы**. Откроется окно **Конструктора макросов**.
4. Выберите макрокоманду **ОкноСообщения**, введите текст сообщения: «ЭТОТ ОТЧЕТ ПРЕДНАЗНАЧЕН ДЛЯ ВЫВОДА ДАННЫХ ПО ГРУППАМ», остальные аргументы произвольно назначьте. Выполните макрос и сохраните его. Закройте макрос и отчет **Состав групп** и заново его запустите. Проверьте работу кнопки. Обратите внимание, что созданных внедренный макрос не отобразился в списке макросов.

7.12. Содержание отчета

1. Название работы, ее задачи и описание последовательности выполнения.
2. Ответы на контрольные вопросы.

7.13. 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 8

ТЕМА: Изучение CASE средства ERWin

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

Следует подготовить презентацию для начального обсуждения предметной области, создав **хранимые** отображения "Определения сущностей" и "Определения связей".

Хранимым отображением [12] 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 5 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.




8.2. 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 ОК, 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).

8.3. 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.

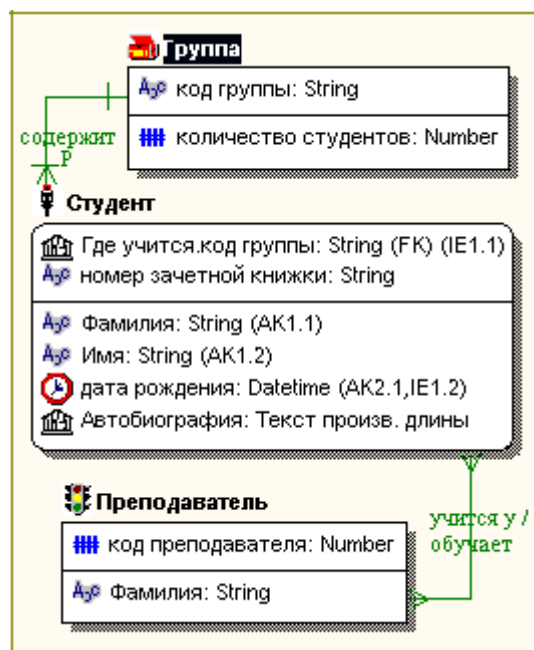
8.4. Порядок выполнения 2-ой части работы

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 *Attributs*
 - 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 **ОК** 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.

8.5. Содержание отчета

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.

Лабораторная работа №9

Тема: Главная кнопочная форма. Итоговое задание.

9.1. Цель работы: изучить способы создания кнопочной формы. Создать БД и интерфейс для работы с БД.

9.2. Главная кнопочная форма.

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

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

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

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

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

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

Аналогично добавляются другие кнопки на кнопочную форму.

9.3. Порядок выполнения лабораторной работы

Лабораторная работа состоит из 2-х частей: изучение диспетчера кнопочных форм и разработка БД.

Изучение диспетчера кнопочных форм.

1. Создайте меню 1-го уровня:

- Закройте все таблицы, формы, отчеты, запросы
- Запустите диспетчер кнопочных форм. В ответ на вопрос «Создать кнопочную форму?» нажать кнопку Да. Перед Вами окно диспетчера кнопочных форм, в

котором перечисляются все страницы кнопочной формы. По умолчанию – одна с именем **Главная кнопочная форма**.

- Нажать кнопку **«Изменить»**, ввести название кнопочной формы - **«Успеваемость студентов»** и нажать кнопку **Заккрыть**.
- Нажать кнопку **Создать** и начнем вводить имена элементов меню 1 уровня.
- В окне **«Создание»** ввести текст - **«Формы»**.
- Нажать кнопку **Создать** и в окне **«Создание»** ввести текст - **«Запросы»**.
- Аналогично создать следующие страницы кнопочной формы :
«Отчеты», «Выход».

2. Создайте меню 2-го уровня.

Теперь последовательно будем раскрывать каждый пункт меню 1-го уровня.

- Отметить курсором главную кнопочную форму **«Успеваемость студентов»** и нажать кнопку **Изменить** и **Создать**, и в окне **Изменение кнопочной формы** ввести текст - **«Формы»**, команда - **«Переход к кнопочной форме»**, выбрать из списка **«Формы»**.
- Точно так же создать элементы кнопочной формы 2 уровня: **Запросы, Отчеты, Выход**. В элементе **Выход** выбрать команду **Выйти из приложения**.

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

3. Создайте меню 3-го уровня:

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

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

- ♦ Долее нажать **Создать**, ввести текст - **«Форма Дисциплины»**, выбрать команду **«Открытие формы в режиме редактирования»**, выбрать из списка форму **Дисциплина**.
- ♦ Точно так же создать элементы меню **Формы** кнопочной формы 3 уровня для выбора двух других любых форм базы данных.

Вы создали меню для ввода и корректировки информации в формах базы данных.

Точно так же создайте меню 3-го уровня для просмотра отчетов, выполнив действия аналогичные действиям при создании меню для форм.

4. Закройте диспетчер кнопочных форм.

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

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

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

- Закройте все таблицы, формы, отчеты, запросы
- Запустите **Диспетчер кнопочных форм**.
- Выделите страницу **«Формы»**, нажмите кнопку **Изменить**.
- Создайте новый элемент кнопочной формы с элементами: текст - **Выход** и командой – **Переход к кнопочной форме**.
- ☐ Кнопочная форма - **«Успеваемость студентов»**.
- Заккрыть окно **«Изменение страницы кнопочной формы»**.
- Выделите **Отчеты**, нажмите кнопку **Изменить**.
- Повторить действия пункта 4. для создания кнопки выхода из пункта меню **«Отчеты»**.
- Заккрыть диспетчер кнопочных форм.

Разработка БД.

Создать БД в соответствии с индивидуальным заданием:

- изучить предметную область проектируемой БД;
- разработать модель «Сущность-связь» - ER-модель проектируемой БД
- создать таблицы, соответствующие сущностям ER- модели БД; создать БД в Access;
- создать схему БД;
- создать формы для ввода данных в БД, ввести записи в БД;
- 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; просмотреть запросы в режиме SQL. **Все запросы на выборку с конкретными данными делать с помощью параметрических запросов.**
- создать с помощью диспетчера кнопочных форм интерфейс для работы с БД.

Индивидуальные задания по проектированию БД

Задание №1: БД Ресторана

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Ингредиенты (Код ингредиента, Наименование ингредиента, Поставщик)[10 записей].4. Меню (Код блюда, Наименование блюда, Стоимость, Время приготовления)[10 записей].5. Состав блюд (Код блюда, Код ингредиента, Объем ингредиента)[30 записей].6. Заказы (Номер заказа, Код сотрудника, Дата, Время, ФИО заказчика, Отметка о выполнении)[10 записей].7. Состав заказов (Номер заказа, Код блюда, Количество блюд)[30 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список ингредиентов определённого поставщика.3. Список ингредиентов и их объемов, затраченных на выполнения всех заказов на определенную дату.4. Список блюд, включающих определенный ингредиент.5. Список из 3 наиболее популярных блюд.6. Список заказов на определенную дату, выполненных данным сотрудником, с подсчетом стоимости каждого заказа.

Задание №2: БД Банка

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Виды вкладов (Код вида вклада, Наименование вида вклада, Минимальный срок вклада, Минимальная сумма вклада, Код валюты, Процентная ставка)[5 записей].4. Валюта (Код валюты, Наименование, Текущий обменный курс)[3 записи].5. Вкладчики (Код вкладчика, ФИО вкладчика, Адрес, Телефон)[10 записей].6. Вклады (Код вкладчика, Код вида вклада, Дата вклада, Дата возврата, Сумма вклада, Сумма возврата, Отметка о возврате вклада, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Виды валютных вкладов с процентной ставкой больше 5%3. Сумма всех вкладов и средний вклад.4. Список вкладов с определенной валютой.5. Список из 3 наиболее популярных видов вкладов.6. Список рублевых вкладов с суммой вклада, большей заданной.

Задание №3: БД Больницы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Лекарства (Код лекарства, Наименование, Фасовка, Стоимость)[5 записей].4. Болезни (Код болезни, Наименование)[10 записей].5. Лечение (Код болезни, Код лекарства, Продолжительность приема, Ежедневная доза)[20 записей].6. Истории болезни (Номер истории, Дата регистрации, Код болезни, Код лечащего врача, ФИО пациента, Возраст, Пол)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список из 3 наиболее часто встречающихся болезней.3. Список лекарств, используемых при лечении определенной болезни.4. Список пациентов старше 50 лет, переболевших заданной болезнью.5. Список пациентов мужского пола, пролечившихся у определенного врача.6. Список лекарств и их объемов, использованных при лечении пациентов определенного врача.

Задание №4: БД Гостиницы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Номера (Номер, Категория, Вместимость, Суточная стоимость)[5 записей].4. Виды услуг (Код услуги, Наименование, Стоимость)[5 записей].5. Заселения (Номер регистрации, Код сотрудника, Дата заселения, Номер, ФИО клиента, Паспортные данные, Дата выезда)[10 записей].6. Предоставленные услуги (Номер регистрации заселения, Код услуги, Дата предоставления)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список номеров гостиницы, суточная стоимость которых превышает заданную.3. Список клиентов, проживавших в гостинице в заданный период4. Список номеров гостиницы, занятых на определенную дату.5. Список регистраций заселения с указанием общих сумм счетов, выставленных клиентам.6. Список из 3 наиболее популярных категорий номеров, занимаемых клиентами

Задание №5: БД Отдела внутренних дел.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности, Код звания)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Код соответствующего звания)[5 записей].3. Звания (Код звания, Наименование, Надбавка)[5 записей].4. Виды преступлений (Код вида преступления, Наименование, Номер статьи, Наказание)[5 записей].5. Преступники (Код преступника, ФИО, Дата рождения, Пол, Адрес)[10 записей].6. Дела (Номер дела, Код вида преступления, Код следователя, Данные о пострадавших, Дата открытия дела, Дата закрытия дела)[10 записей].7. Состав преступников в делах (Номер дела, Код преступника)[15 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список сотрудников, звания которых не соответствуют занимаемым должностям.3. Список незакрытых дел, расследуемых определенным следователем.4. Список преступников-рецидивистов с указанием количества совершенных преступлений.5. Список следователей с указанием количества закрытых дел.6. Список из 3 наиболее частых видов преступлений.

Задание №6: БД Авиакомпаний.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Типы самолётов (Код типа, Наименование, Вместимость)[5 записей].4. Самолёты (Код самолёта, Код типа, Дата выпуска, Налётано часов, Дата последнего ремонта)[10 записей].5. Экипажи (Код экипажа, Номер экипажа)[5 записей].6. Состав экипажей (Код экипажа, Код сотрудника)[15 записей]7. Рейсы (Номер рейса, Код самолёта, Код экипажа, Дата, Время, Откуда, Куда)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список самолетов, дата последнего ремонта которых меньше заданной.3. Список типов самолётов с указанием их количества4. Состав экипажа, наиболее часто летающего на самолетах данного типа.5. Список рейсов, выполненных в определенный день6. Список из 3 наиболее используемых типов самолетов

Задание №7: БД Видео-проката.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Жанры (Код жанра, Наименование жанра)[5 записей].4. DVD (Код DVD, Наименование фильма, Год создания, Страна, Главный актёр, Код жанра, Количество экземпляров, Цена)[10 записей].5. Прокат (Номер проката, Код сотрудника, ФИО клиента, Адрес, Телефон, Дата взятия, Дата возврата, Отметка об оплате)[10 записей].6. Состав DVD в прокате (Номер проката, Код DVD)
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список DVD на руках с указанием количества.3. Список не востребованных DVD4. Список DVD с фильмами за заданный год5. Список из 3 наиболее популярных DVD.6. Перечень сотрудников с невозвращенными из проката DVD с указанием количества DVD и их общей стоимостью.

Задание №8: БД Библиотеки.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Издательства (Код издательства, Наименование, Город, Адрес)[5 записей].4. Жанры (Код жанра, Наименование) [5 записей].5. Книги (Код книги, Наименование, Автор, Код издательства, Год издания, Код жанра, Количество экземпляров) [10 записей].6. Читатели (Код читателя, ФИО, Адрес, Телефон) [10 записей].7. Выдачи (Номер выдачи, Код читателя, Код сотрудника, Дата выдачи, Планируемая дата возврата, Отметка о возврате) [10 записей].8. Состав выдач (Номер выдачи, Код книги)
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданной должности с указанием всех реквизитов.2. Список книг библиотеки конкретного автора с указанием всех реквизитов.3. Перечень книг и читателей, у которых на руках книги с просроченной датой возврата4. Список книг на руках с указанием количества экземпляров.5. Список из 3 наиболее популярных книг.6. Список сотрудников с указанием количества выданных книг и количества возвращенных книг.

Задание №9: БД Радиостанции.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Исполнители (Код исполнителя, Наименование)[5 записей].4. Жанры (Код жанра, Наименование)[5 записей].5. Записи (Код записи, Наименование, Код жанра, Код исполнителя, Год записи)[10 записей].6. Сетка вещания (Код сотрудника, Код записи, Дата)[30 записей].
Запросы:	<ol style="list-style-type: none">7. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.8. Список записей с заданным годом записи.9. Список из 3 записей, наиболее часто звучащих в эфире в заданный период времени.10. Список сотрудников, работающих в эфире в определенный день.11. ФИО сотрудника, наиболее часто работающего в определенный месяц и год.12. Список записей наиболее востребованного жанра.

Задание №10: БД Таксопарка.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Модели (Код модели, Наименование, Мощность, Объем двигателя)[5 записей].4. Тарифы (Код тарифа, Наименование, Цена за км)[5 записей].5. Автомобили (Код автомобиля, Код модели, Регистрационный номер, Год выпуска, Код сотрудника-шофёра, Код сотрудника-механика)[10 записей].6. Вызовы (Код автомобиля, Дата, Время, Телефон заказчика, Расстояние, Код тарифа, Код сотрудника-оператора)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Автопарк с указанием всех реквизитов автомобилей3. Список автомобилей заданной модели4. Список вызовов за заданную дату с вычислением стоимостей отдельных вызовов и их общей суммы5. Список сотрудников-операторов с указанием количества оформленных вызовов6. Перекрестный запрос, вычисляющий количество вызовов, принятых сотрудниками-операторами по каждому конкретному автомобилю.

Задание №11: БД Туристического агентства.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Отели (Код отеля, Наименование, Страна, Город, Адрес, Телефон, Количество звёзд, Контактное лицо)[10 записей].4. Дополнительные услуги (Код услуги, Наименование, Цена) [5 записей].5. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон)[5 записей].6. Путёвки (Код путевки, Код отеля, Код клиента, Код сотрудника, Дата начала, Дата окончания, Отметка о бронировании, Отметка об оплате)[10 записей].7. Состав услуг по путевкам (Код путевки, Код услуги)
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Список путёвок с указанием всех реквизитов3. Список сотрудников с указанием количества проданных путевок4. Список из 3 наиболее популярных стран5. Список отелей с указанием количества проданных путевок6. Список забронированных, но неоплаченных путёвок

Задание №12: БД Страховой компании.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Риски (Код риска, Наименование)[5 записей].4. Виды полисов (Код вида полиса, Наименование)[5 записей].5. Риски по видам полисов (Код вида полиса, Код риска)6. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон)[10 записей].7. Полисы (Номер полиса, Код вида полиса, Код клиента, Код сотрудника, Дата начала, Дата окончания, Стоимость, Сумма выплаты, Отметка о выплате)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Список действующих полисов3. Список оконченных полисов, по которым производились выплаты4. Список сотрудников с указанием количества и суммарной стоимости оформленных полисов5. Список клиентов с указанием номеров и суммы выплаты полисов с выплатой6. Список из 3 наиболее популярных видов полисов

Задание №13: БД Брачного агентства.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Знаки зодиака (Код знака, Наименование)[5 записей].4. Национальности (Код национальности, Наименование)[5 записей].5. Виды услуг (Код вида услуги, Наименование, Цена)[5 записей].6. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Возраст, Рост, Вес, Количество детей, Код знака, Код национальности, Адрес, Телефон)[10 записей].7. Оказанные услуги (Код клиента, Код вида услуги, Дата оказания услуги, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Список сотрудников с указанием количества оказанных услуг3. Список клиентов с указанием всех реквизитов.4. Список клиентов определенной национальности, родившихся под определенным знаком зодиака5. Список национальностей с указанием количества клиентов6. Список из 3 наиболее популярных видов услуг.

Задание №14: БД Сервис-центра.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)3. Запчасти (Код запчасти, Наименование, Цена)[5 записей].4. Ремонтируемые модели (Код модели, Наименование, Тип изделия, Производитель)[5 записей].5. Виды неисправностей (Код вида, Наименование неисправности, Код модели, Цена работы)[5 записей].6. Наборы запчастей (Код вида неисправности, Код запчасти, Количество)[10 записей].7. Обслуживаемые магазины (Код магазина, Наименование, Адрес, Телефон)[5 записей].8. Заказы (Номер заказа, Дата заказа, Серийный номер изделия, ФИО заказчика, Код вида неисправности, Код магазина, Код сотрудника, Отметка о гарантии)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список заказов с указанием всех реквизитов и общей стоимости.3. Список из 3 наиболее частых видов неисправностей4. Наименование магазина, в котором приобретено наибольшее количество ремонтируемых изделий.5. Список 3-х наиболее надежных моделей6. Магазины, в которых были куплены изделия определенной модели.

Задание №15: БД Школы.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. 3. Ученики (Код ученика, ФИО, Дата рождения, Пол, Адрес, Код класса) [10 записей]. 4. Классы (Код класса, Номер класса, Код сотрудника - классного руководителя, Код вида)[5 записей]. 5. Виды классов (Код вида, Наименование)[5 записей]. 6. Предметы (Код предмета, Наименование, Код сотрудника - учителя)[10 записей]. 7. Расписание (Код класса, Код предмета, День недели, Номер урока)[10 записей].
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников, возраст которых превышает заданную величину, с указанием всех реквизитов. 2. Список учеников заданного возраста. 3. Расписание занятий с указанием всех реквизитов, упорядоченное по дням недели и номерам уроков. 4. Список классов заданного года обучения с указанием всех реквизитов 5. Количество уроков в неделю каждого учителя по соответствующим предметам. 6. Список из 3 наиболее загруженных учителей.

Задание №16: БД Транспортной компании.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Модели автомобилей (Код модели, Наименование, Грузоподъемность) [5 записей].4. Виды грузов (Код вида груза, Наименование)[5 записей].5. Грузы (Код груза, Наименование, Код вида груза, Вес, Срок годности)[5 записей].6. Автомобили (Код автомобиля, Код модели, Регистрационный номер, Код сотрудника-механика)[5 записей].7. Рейсы (Код автомобиля, Код сотрудника-водителя, Заказчик, Дата отправления, Дата прибытия, Код груза, Цена, Отметка об оплате, Код сотрудника-диспетчера)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых меньше заданного, с указанием всех реквизитов.2. Автопарк с указанием всех реквизитов автомобилей3. Суммарный вес грузов, перевезенных определенным автомобилем.4. Список рейсов, выполненных с перегрузкой автомобиля5. Список рейсов, выполненных определенным водителем, с указанием веса груза и цены.6. Список из 3 наиболее часто перевозимых видов грузов.7. Распределение количества грузов по видам грузов и автомобилям.

Задание №17: БД Проката автомобилей.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Модели автомобилей (Код модели, Наименование, Мощность) [5 записей].4. Автомобили (Код автомобиля, Код модели, Регистрационный номер, Год выпуска, Пробег, Цена автомобиля, Цена дня проката)[10 записей].5. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Адрес, Телефон, Паспортные данные) [5 записей].6. Прокат (Код автомобиля, Код клиента, Дата выдачи, Дата возврата, Отметка об оплате, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых больше заданного, с указанием всех реквизитов.2. Автопарк с указанием всех реквизитов автомобилей.3. Список автомобилей, находящихся в прокате.4. Список невозвращенных автомобилей с указанием всех реквизитов проката и автомобиля.5. Список возвращенных и оплаченных прокатов с указанием всех реквизитов и стоимости проката.6. Список из 3 наиболее популярных моделей автомобилей.7. Распределение нагрузки (количества прокатов) по сотрудникам и автомобилям.

Задание №18: БД Оптового склада.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. 3. Товары (Код товара, Наименование, Единица отпуска товара) [10 записей]. 4. Поставщики (Код поставщика, Наименование, Адрес, Телефон) [5 записей]. 5. Поставляемые товары (Код поставщика, Код товара) 6. Заказчики (Код заказчика, Наименование, Адрес, Телефон) [5 записей]. 7. Продажи (Номер накладной, Код заказчика, Код сотрудника, Дата выдачи) [10 записей]. 8. Состав продаж (Номер накладной, Код товара, Объем, Цена)
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников заданного возраста с указанием всех реквизитов. 2. Список заказчиков с указанием количества продаж за заданный период 3. Список товаров, выданных по накладной с заданным номером, с указанием всех реквизитов продажи и товаров в накладной 4. Список сотрудников с указанием количества продаж и общей цены проданных товаров. 5. Список из 3 поставщиков, чьи товары пользуются наибольшим спросом. 6. Распределение количества продаж по сотрудникам и товарам.

Задание №19: БД картинной галереи

Таблицы:	<ol style="list-style-type: none">1. Сотрудники галереи (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей]2. Должности (Код должности, Наименование должности, Оклад) [5 записей]3. Залы галереи (Номер зала, Площадь зала, Код ответственного лица) [5 записей]4. Картины (Код картины, Номер зала, Наименование, Автор, Страховая стоимость, Дата поступления) [20 записей]5. Экскурсии (Код экскурсии, Тема экскурсии, Время проведения, Продолжительность, Цена билета, Код экскурсовода) [5 записей]6. Содержание экскурсий (Код экскурсии, Код картины) [25 записей]
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых меньше заданного, с указанием всех реквизитов.2. Список сотрудников определенной должности3. Список залов с указанием количества картин в них4. Каталог картин галереи, упорядоченный по номерам залов, авторам и наименованиям5. Список экскурсий с указанием количества посещаемых залов6. Список из 3 картин, осматриваемых в наибольшем количестве экскурсий

Задание №20: БД Спортивного клуба

Таблицы	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности) [10 записей].2. Должности (Код должности, Наименование должности, Оклад) [5 записей].3. Помещения (Код помещения, Назначение, Площадь) [5 записей].4. Клиенты (Код клиента, ФИО, Адрес, Телефон)[10 записей].5. Абонементы (Код абонемент, Название, Цена) [10 записей].6. Используемые помещения (Код абонемент, Код помещения) [10 записей].7. Продажи абонементов (Код продажи, Код абонемент, Код клиента, Дата продажи, Код тренера, Дата начала, Дата окончания, Отметка об оплате, Код сотрудника) [20 записей].
Запросы	<ol style="list-style-type: none">1. Получить список из 3 сотрудников, которые оформили наибольшее число абонементов2. Вычислить количество абонементов по помещениям и общее количество абонементов.3. Определить название наиболее востребованного абонемент.4. Определить количество продаж абонементов за определенный период времени.5. Определить ФИО наиболее востребованного тренера.6. Получить распределение числа проданных абонементов по сотрудникам (не тренерам) и абонементам.

Задание №21: БД Рекламного агентства наружной рекламы

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад)[5 записей]. 3. Рекламные конструкции (Код конструкции, Номер конструкции, Адрес, Стоимость дня аренды одной стороны конструкции)[5 записей] 4. Рекламные места (Код места, Код конструкции, Сторона конструкции) [10 записей]. 5. Заказчики (Код заказчика, Наименование заказчика, Адрес, Телефон) [10 записей]. 6. Договоры аренды (Номер договора, Дата начала, Дата окончания, Код заказчика, Код заказчика, Код сотрудника, Отметка об оплате) [10 записей]. 7. Списки арендуемых мест (Код договора, Код рекламного места, Описание рекламы)[25 записей]
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников заданного возраста с указанием всех реквизитов. 2. Список рекламных мест с указанием всех реквизитов 3. Список договоров с указанием всех реквизитов и общей стоимости аренды 4. Список арендуемых по заданному договору рекламных мест с указанием всех реквизитов 5. Список неоплаченных договоров с указанием всех реквизитов 6. Список из 3 наиболее популярных рекламных конструкций

Задание №22: БД Компьютерной фирмы.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности) [10 записей]. 2. Должности (Код должности, Наименование должности, Оклад)[5 записей]. 3. Виды комплектующих (Код вида, Наименование)[15 записей]. 4. Комплектующие (Код комплектующего, Код вида, Марка, Цена)[15 записей]. 5. Заказчики (Код заказчика, ФИО, Телефон)[10 записей]. 6. Услуги (Код услуги, Наименование, Стоимость)[5 записей]. 7. Заказы (Номер заказа, Код заказчика, Дата заказа, Отметка об оплате, Отметка об исполнении, Код сотрудника)[10 записей]. 8. Состав комплектующих в заказах(Номер заказа, Код комплектующего, Количество) 9. Состав услуг в заказах (Номер заказа, Код услуги)
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников, возраст которых меньше заданного, с указанием всех реквизитов. 2. Список комплектующих с указанием всех реквизитов 3. Список неоплаченных заказов за определенную дату 4. Список комплектующих конкретного заказа с заданным номером. 5. Список услуг с указанием их общего количества в исполненных заказах 6. Список из 3 заказчиков с наибольшим количеством заказов

Задание №23: БД ГИБДД.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности, Код звания)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Код соответствующего звания)[5 записей]. 3. Звания (Код звания, Наименование, Надбавка)[5 записей]. 4. Модели автомобилей (Код модели, Наименование, Мощность двигателя, Рабочий объем двигателя)[10 записей]. 5. Владельцы (Код владельца, ФИО, Дата рождения, Адрес, Паспортные данные)[15 записей]. 6. Регистрация (Номер свидетельства, Регистрационный знак, Дата регистрации, VIN автомобиля, Код модели, Год выпуска, Номер кузова, Номер двигателя, Цвет, Номер техпаспорта, Код владельца, Код сотрудника)[15 записей]. 7. Автомобили в угоне (Регистрационный знак, Дата угона, Дата обращения, Дата нахождения, Код сотрудника)[5 записей].
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников с указанием всех реквизитов. 2. Список моделей автомобилей с указанием соответствующего количества зарегистрированных машин 3. Список владельцев угнанных автомобилей. 4. Список регистраций заданным сотрудником за заданный период 5. Список угнанных автомобилей, находящихся в угоне более 3 месяцев и до сих пор не найденных 6. Список из 3-х наиболее часто угоняемых моделей автомобилей

Задание №24: БД Кинотеатра.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад)[5 записей].3. Жанры (Код жанра, Наименование)[5 записей].4. Фильмы (Код фильма, Наименование, Код жанра, Длительность, Страна производитель, Возрастные ограничения)[10 записей].5. Сеансы (Код сеанса, Дата, Время начала, Код фильма, Цена билета)[10 записей].6. Билеты (Код сеанса, Номер ряда, Номер места, Код сотрудника)[15 записей].
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых больше заданного, с указанием всех реквизитов.2. ФИО сотрудника, продавшего наибольшее количество билетов за определенный период.3. Список фильмов с указанием общего количества просмотревших зрителей.4. Список проданных билетов на определенный сеанс с заданной датой и временем начала.5. Список из 3 фильмов с наибольшей посещаемостью6. Распределение количества проданных билетов по сеансам и фильмам за определенный день недели.

Задание №25: БД Автосалона.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Телефон, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад)[5 записей]. 3. Производители (Код производителя, Наименование, Страна)[5 записей]. 4. Дополнительное оборудование (Код дополнительного оборудования, Наименование, Цена)[5 записей]. 5. Тип кузова (Код типа кузова, Название)[5 записей]. 6. Автомобили (Код автомобиля, Марка, Код производителя, Год выпуска, Код типа кузова, Номер кузова, Номер двигателя, Цвет, Базовая цена)[10 записей]. 7. Продажная комплектация (Код автомобиля, Код дополнительного оборудования)[10 записей]. 8. Продажи (Номер договора продажи, Дата продажи, Код автомобиля, ФИО покупателя, Адрес, Телефон, Паспортные данные, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников заданного возраста с указанием всех реквизитов. 2. ФИО сотрудника, продавшего наибольшее количество автомобилей за определенный месяц и год. 3. Каталог автомобилей с указанием всех реквизитов. 4. Список продаж за заданный период с указанием всех реквизитов и стоимости с учетом комплектации. 5. Список сотрудников с указанием количества проданных ими автомобилей 6. Список из 3 производителей, лидирующих по объему продаж

ЛИТЕРАТУРА

1. Сосновиков Г.К., Шакин В.Н. Основы реляционных баз данных: Учебное пособие. /МТУСИ.-2013.106 с.
2. Сосновиков Г.К., Шакин В.Н. Технология использования системы управления реляционной базой данных. Лабораторные работы. Электр. Пособие. /МТУСИ. 2012 .
3. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. А. Д. Хомоненко. – СПб.: КОРОНА.-2009.
4. Тарасов В.Л. Работа с базами данных в ACCESS 2010. ЧАСТЬ 1 и 2: Учебно-методическое пособие. – [электронный ресурс]. – Нижний Новгород: Нижегородский госуниверситет, 2014. – 126 с
5. Е.М. Карчевский, И.Е. Филиппов, И.А. Филиппова. Access 2010 в примерах. Учебное пособие. Казанский университет. 2012.
6. С.В. Одиноккина. Разработка баз данных в Microsoft Access 2010. - СПб: НИУ ИТМО, 2012. –83 с.
7. Пушкинов 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.
8. Пушкинов 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.
9. Бекаревич Ю.Б., Пушкина Н.В. Microsoft Access за 21 занятие. – СПб.:БХВ-Петербург, 2005.-544 с.:ил.
10. Зверева Н.Н. Методические указания к лабораторным работам по Access 97/Изд. Уфимского авиационного института./Каф. информатики и информационных технологий.- Уфа, 2001.
11. Палеев А.Б. Уч. пособие по курсу: Базы данных / Лаб. работа: Использование Erwin для разработки модели данных. – М: МГУПП, 2002.
12. Маклаков С.В.. ВРwin и ERwin CASE-средства разработки информационных систем - М.: "ДИАЛОГ-МИФИ", 2000.
13. Малыхина М.П. Базы данных: основы, проектирование, использование: учебное пособие. / М.П. Малыхина. СПб.: БХВ – Петербург, 2007. 528с.
14. Карпова Т.С. Базы данных: модели, разработка, реализация. /Т.С. Карпова. СПб.: Питер, 2001. 304с.
15. Шакин В.Н., Сосновиков Г.К, Юскова И.Б. Методические указания по дисциплине ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ БД/ МТУСИ.- М., 2004.
16. Сосновиков Г.К., Шакин В.Н., Юскова И.Б.. Методические указания и контрольные задания по дисциплине ОСНОВЫ ПОСТРОЕНИЯ БД/ МТУСИ.- М., 2004.
17. Сосновиков Г.К., Шакин В.Н.. Практикум по проектированию баз данных средствами СУБД Access./ МТУСИ.- М., 2006.
18. Дейт К., Дж. Введение в системы баз данных.-6 изд, : Пер. с англ.М.:Издательский дом “Вильямс”, 1999, 848 с.