

ВВЕДЕНИЕ

Методические указания содержат методический материал к восьми двухчасовым лабораторным работам по курсу Базы данных (или к четырем работам по 4 часа). Все работы связаны общей темой и могут выполняться только последовательно. В результате выполнения работ будет создана полностью работоспособная база данных “Деканат”, а студенты получат устойчивые навыки разработки баз данных в среде Access 2000.

ЛАБОРАТОРНАЯ РАБОТА №1

Цель работы: анализ предметной области и определение функций разрабатываемой системы, разработка схемы данных. Первоначальное знакомство с Access: создание таблиц и схемы данных.

Порядок выполнения работы

Прочитать внимательно пояснения, приведенные ниже.

Анализ предметной области

Предметная область разрабатываемой системы — деканат любого факультета. Требуется автоматизировать работу сотрудников деканата и организовать ведение документации в электронном виде.

При опросе сотрудников деканата были выяснены основные **функции** разрабатываемой системы:

- Занесение первичной информации в базу данных и ее корректировка в процессе работы;
- Формирование и печать зачетных и экзаменационных ведомостей и ввод оценок в базу данных;
- Автоматический перевод студентов на следующий курс;
- Формирование и печать вкладышей к диплому, содержащих сведения обо всех оценках студента;
- Занесение сведений об окончивших институт в архив и удаление их из базы данных;
- Анализ успеваемости студента, группы, курса, специальности или факультета в целом;
- Справочные функции (т.е. необходим быстрый поиск информации о конкретной специальности, группе, студенту).

Затем была определена **информация**, которую необходимо хранить для выполнения заданных функций:

- Сведения о специальностях факультета;
- Названия групп и другие сведения о группах;
- Списки студентов групп, разные сведения о студентах;
- Названия предметов;
- Учебный план специальностей (т.е. какие предметы изучаются на различных специальностях факультета);
- Успеваемость студентов по всем предметам учебного плана за весь период обучения.

Следующий шаг проектирования системы — разработка **логической схемы** базы данных. В реляционной базе данных вся информация может храниться только в таблицах. Логическая схема содержит сведения о составе таблиц базы данных и связях между ними. Логическая схема представляет собой модель данных предметной области и не привязана к конкретной среде разработки.

Логическая схема базы данных “Деканат”

Для построения логической схемы анализируется вся информация, подлежащая хранению, и определяется, какие таблицы необходимо выделить и как связать их между собой. Для изображения логической схемы используются стандартные обозначения. При этом таблицы обычно изображаются с помощью прямоугольников, связи — линиями, которые проводятся от одного прямоугольника к другому. Логическая схема базы данных “Деканат” представлена на рис. 1.1 (используется Международный стандарт для изображения схемы базы данных IDEF1X).

Логическая схема базы данных “Деканат”

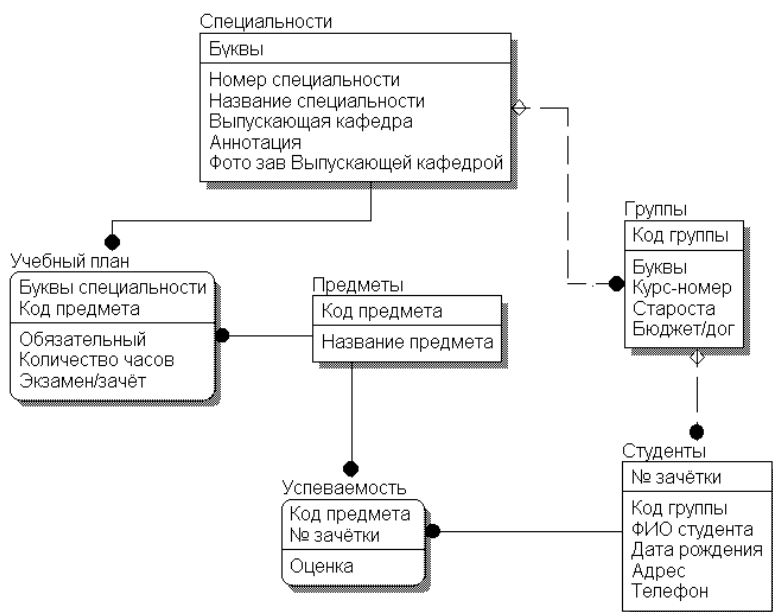


Рис 1.1

Пояснения к схеме данных:

В каждой таблице выделяется первичный ключ. Это один или несколько столбцов таблицы, которые служат уникальным идентификатором каждой строки. На схеме данных первичный ключ отделяется от остальных столбцов горизонтальной линией. **Access** автоматически проверяет уникальность значений в ключевых столбцах.

Эти же столбцы служат и для связи между таблицами. В каждой связи выделяют главную и подчиненную таблицу. Первичный ключ главной таблицы соответствует аналогичному столбцу (или нескольким столбцам) подчиненной таблицы, который называется внешним ключом.

На схеме данных связь обозначается линией с жирной точкой около подчиненной таблицы. Сплошная линия обозначает идентифицирующую (сильную) связь между таблицами, когда первичный ключ главной таблицы входит в состав первичного ключа подчиненной. Пунктирная линия соответствует более слабой (не идентифицирующей) связи. В этом случае подчиненная таблица имеет собственный первичный ключ, а первичный ключ главной таблицы является не ключевым столбцом подчиненной таблицы.

Пояснения к составу таблиц :

Столбец **Буквы** в таблице **Специальности** означает сокращенное название специальности (например, ЭМ, ЭПО, ЭВ, ПМ и т.д.). Именно этот столбец сделан ключевым, поскольку на факультете все аббревиатуры специальностей уникальны.

С другой стороны, в таблице **Группы** этот же столбец не является ключевым, потому что каждой специальности соответствуют несколько групп. Вообще название каждой группы состоит из двух частей (столбцов):

- буквы специальности (неизменная часть);
- цифры, соответствующие курсу и номеру группы на курсе (эта часть при переводе на следующий курс изменяется).

Например: группа ЭМ-11 через год станет группой ЭМ-21 и т.д.

Хотя название каждой группы уникально, неудобно иметь ключ в таблице, который меняется каждый год. Поэтому в данной таблице вводится дополнительный ключевой столбец, который назовем **Кодом группы**. Введение дополнительного ключевого столбца с названием **Код** — типовой прием, который часто используется в базах данных. Этот столбец и будет служить для связи между таблицами **Группы** и **Студенты**, т.е. в таблице **Студенты** для каждого студента будем хранить его код группы.

Аналогичный прием с введением дополнительного ключевого столбца введем в таблице **Предметы**, которая содержит названия всех предметов, которые изучаются на разных специальностях факультета. Такие таблицы называют **Справочниками**, для нашего случая это справочник всех предметов, которые изучаются на факультете. Несмотря на то, что название каждого предмета уникально, оно слишком длинное, чтобы использовать его в

качестве первичного ключа и столбца для связи между таблицами. Поэтому вводим ключевой столбец **Код предмета**.

В таблице **Студенты** имеется хороший первичный ключ — номер зачетной книжки студента, поскольку у всех студентов различные номера зачетных книжек. Внешний ключ для связи с таблицей **Группы** — **Код группы**. Назначение остальных столбцов понятно без комментариев. В таблице **Успеваемость** хранятся последние оценки студента по предметам, которые он изучал, поскольку именно они нужны для вкладыша к диплому. Поэтому сочетание **№ зачетки** и **Код предмета** уникально и может служить первичным ключом (допустим, если студент два семестра сдавал физику, то в базе данных будем хранить только последнюю оценку, т.к. по действующим правилам именно последняя оценка идет в диплом).

Поясним назначение таблицы **Учебный план**. В ней хранятся данные о том, какие предметы изучаются на каждой специальности. При формировании экзаменационных и зачетных ведомостей **Access** будет автоматически проверять, входит ли предмет в учебный план специальности. Кроме того, иногда может возникнуть необходимость получить справочную информацию об изучаемых на каждой специальности предметах. Первичный ключ этой таблицы составной — **Буквы специальности** и **Код предмета**.

Задание:

Выполняется в текстовом редакторе Word. Изобразите все таблицы схемы данных в общепринятом виде, заполнив их минимальным количеством произвольной информации:

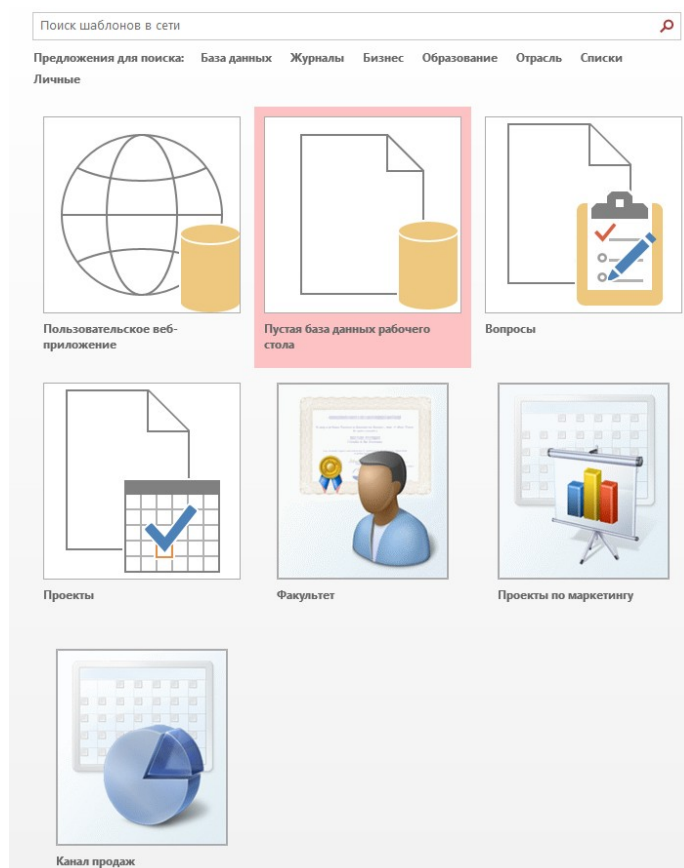
- Данные об одной специальности
- Учебный план из 2-3 предметов
- Две учебные группы, в каждой 1-2 студента
- У каждого студента 1-2 оценки

Проследите, чтобы данные всех таблиц были *согласованы*, т.е. все столбцы связи содержали одни и те же значения. Сохраните текстовый файл для отчета.

После выполнения этого задания структура базы данных будет понятна, и можно переходить к практической части работы в среде Access.

Первые шаги в системе Access

Прежде чем приступить к созданию таблиц базы данных, необходимо **создать Пустая база данных рабочего стола**.



Access формирует файл с расширением .mdb (Microsoft DataBase), в котором хранятся не только все таблицы с данными, но и программные компоненты, предназначенные для обработки данных и организации пользовательского интерфейса. По сути дела, вся разрабатываемая информационная система будет представлять собой один файл.

После запуска Microsoft Access выводит первое диалоговое окно, позволяющее начать создание базы данных. Если это окно не выводится, начать создание базы данных можно с помощью команды **Файл/Создать** или кнопки **Создать базу данных** на панели инструментов.

Задание:

Создать файл базы данных и сохранить его в своей личной папке под именем, например, dekanat.mdb или любым другим подходящим именем.

В результате открывается окно базы данных.

Прочитать внимательно:

Окно базы данных имеет строку заголовка, содержащую название БД, три кнопки управления размером окна, кнопки представления информации в окне. Слева располагается список объектов базы данных: **Таблицы, Запросы, Формы, Отчеты, Страницы, Макросы и Модули (будет появляться постепенно, по мере создания этих объектов)** (см. рис 1.2). Назначение этих объектов будем изучать постепенно. Лента инструментов так же имеет закладки **Файл, Главная, Создание, Внешние данные, Работа с базами данных**

Окно базы данных

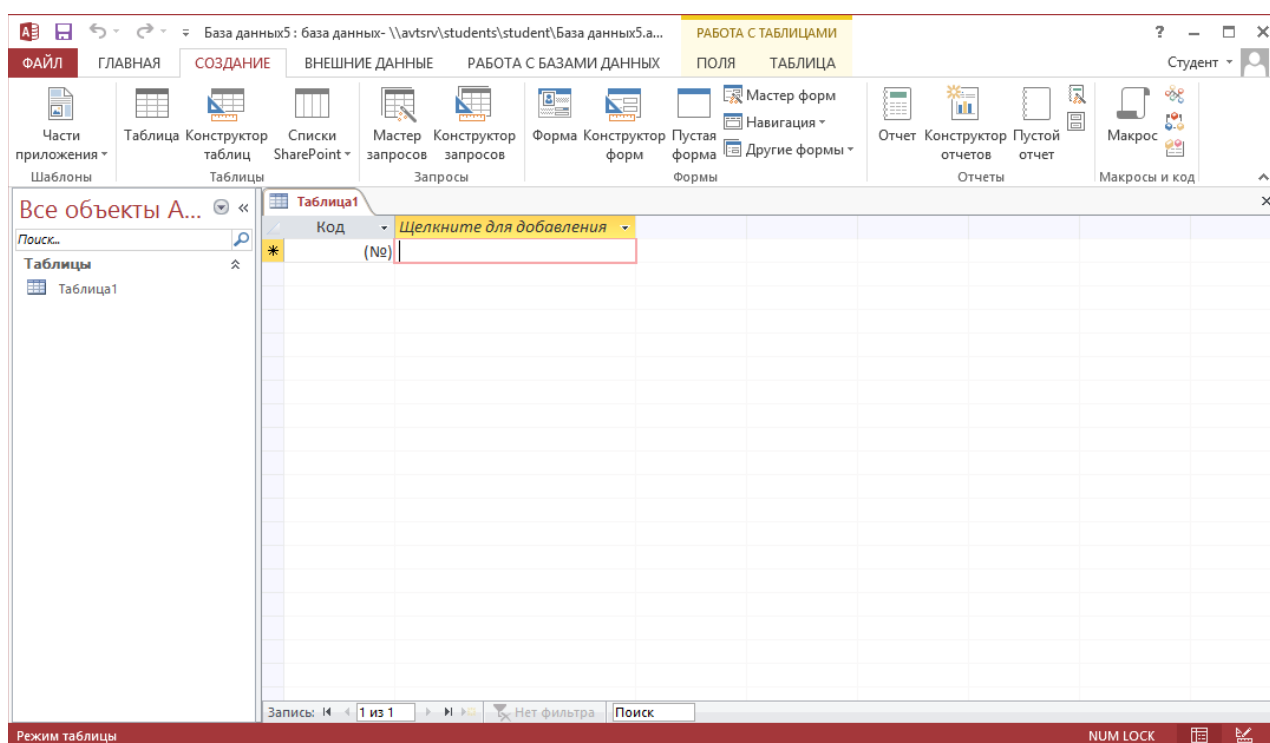


Рис 1.2

Для выбора режима отображения объекта (обычный или режим **Конструктора**), форматирования текста в созданных объектах, сортировка и фильтрация содержания располагается в закладке **Главная**

Для создания необходимого объекта используется вкладка **Создание**

Импорт данных из других баз, обмен информацией с другими источниками данными, импорт и экспорт данных, настраивается на вкладке **Внешние данные**

Управление базой данных, схемы данных, зависимости объектов располагаются на вкладке **Работа с базами данных**

- Режим **Конструктор** (позволяет перейти в режим доработки любого ранее созданного объекта, например, можно добавить еще один столбец к таблице или поменять внешний вид формы);
- На вкладке **Создание** (позволяет приступить к созданию нового объекта любого выбранного типа). Обычно при создании любого объекта возможны два основных способа: *Мастер* или *Конструктор*.

Мастер (wizard — дословный перевод «волшебник») создает нужный объект автоматически, задавая при этом некоторые уточняющие вопросы.

Конструктор позволяет создавать объект «вручную», при этом разработчик может создать объект с любыми нужными характеристиками. Но, конечно, работа с конструктором сложнее и требует больше времени, чем быстрое создание объекта мастером.

При выполнении лабораторных работ рекомендуется использовать в основном конструктор. Отдельно оговаривается, в каких случаях будем использовать мастер.

Создание таблиц Специальности, Группы, Студенты

Создание таблицы базы данных — это определение и сохранение в базе данных ее структуры.

В Access столбцы таблиц называются иначе *полями*, строки — *записями*.

Требуется задать имена и размещение полей в таблице, тип данных каждого поля, размер и другие свойства полей. Также задается первичный ключ таблицы.

В Access можно создавать таблицы различными способами. Самый надежный из них — **конструктор** таблиц. Другие способы в данной работе не рассматриваются.

Задание: создать структуру таблицы Специальности и заполнить таблицу данными для двух специальностей.

Указания к выполнению:

В ленте инструментов выбрать закладку **Создание** и выбрать инструмент **КОНСТРУКТОР ТАБЛИЦ**. Создайте таблицу **Специальности**.

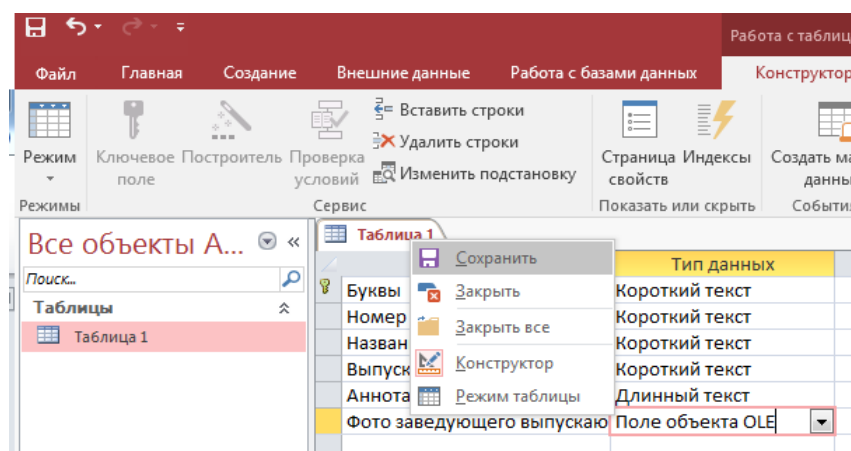
Параметры структуры таблицы Специальности

Таблица 1.1

Имя поля	Ключевое поле	Тип данных	Размер поля
Буквы	Да	Короткий текст	5
Номер специальности		Короткий текст	6
Название		Короткий текст	60
Выпускающая кафедра		Короткий текст	50
Аннотация		Длинный текст	
Фото заведующего выпускающей кафедрой		Поле объекта OLE	

Далее необходимо **создать первичный ключ таблицы**, выделив поле **Буквы**, щелкнуть на области маркировки слева от имени поля, нажать кнопку **Ключевое поле**. Признаком установки ключа является изображение ключа слева от имени поля.

Сохранить созданную структуру таблицы и присвоить имя новой таблице - Специальности, в контекстном меню таблицы выбрать пункт **Сохранить**.



Если структура таблицы не была сохранена, то при выходе из окна конструктора таблиц Access автоматически предложит сохранить таблицу.

Созданная таблица пока не содержит ни одной строки с данными. Для заполнения таблицы откроем ее в режиме таблицы щелкнув на нее два раза и заполним данными для двух любых специальностей. Для заполнения поля **Фото заведующего** выбрать в контекстном меню

Вставить объект, после чего установить переключатель **Создать из файла** и найти подходящий файл с рисунком. Если при этом установить флажок **Связь**, то рисунок не помещается в базу данных, а устанавливается связь с файлом, содержащим рисунок. Лучше не устанавливать этот флажок, а для того, чтобы файл базы данных имел разумные размеры, поместите фото только одного заведующего.

Убедились, что заполнять таким образом таблицу не очень удобно, т.к. она не умещается целиком на экране и приходится пользоваться полосами прокрутки.

Рисунки при просмотре таблицы вообще не видны. В следующей работе будет разработана удобная экранная форма для просмотра и заполнения этой таблицы, тогда добавим данные еще нескольких специальностей.

Задание: создайте структуру таблицы **Студенты**. Внимательно прочитайте, зачем нужны индексы и как создаются простые индексы по одному полю.

Указания и пояснения:

Основные параметры структуры таблицы Студенты

Таблица 1.2

Имя поля	Ключевое поле	Тип данных	Размер поля
Код группы		Счетчик	Длинное целое
№ зачетки	Да	Короткий текст	8
ФИО студента		Короткий текст	50
Дата рождения		Дата и время	
Адрес		Короткий текст	50
Телефон		Короткий текст	10

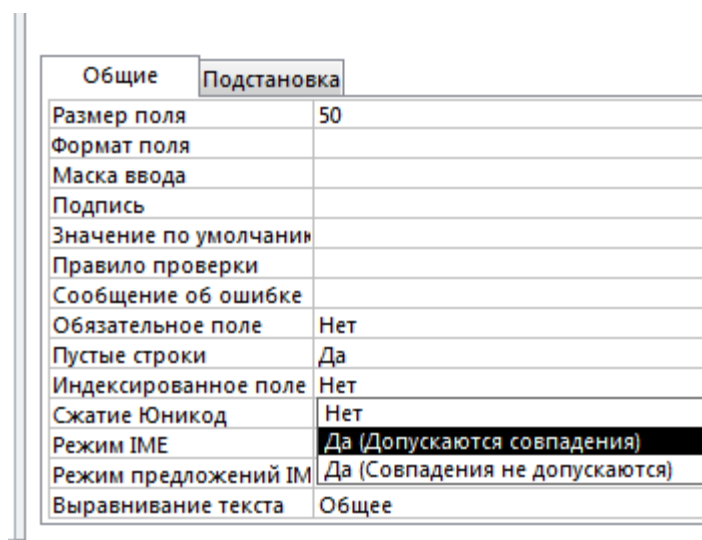
Создание простого индекса по одному полю таблицы Студенты

При создании таблицы необходимо решить еще один вопрос: по каким полям требуется создать **индекс**. Индексы служат для ускорения поиска данных какого-либо столбца таблицы, однако замедляют процессы ввода и редактирования данных.

Различают **уникальный** и **неуникальный** индексы.

По **ключевым** полям **автоматически** создается **уникальный индекс**, таким способом ускоряется проверка на **уникальность значений**.

В таблице **Студенты** достаточно создать индекс только по полю **ФИО студента**, поскольку наиболее часто выполняется поиск именно по этому полю. Для создания индекса в свойстве Индексированное поле необходимо выбрать из списка **Да (допускаются совпадения)**.



Таким образом, создали **неуникальный индекс** на случай полных однофамильцев.

Сохраните таблицу **Студенты**, но пока не заполняйте ее данными. Сделаем это позже.

Задание: создайте таблицу **Группы**, создайте для нее составной уникальный индекс для проверки уникальности названия группы. Создайте поле подстановки.

Указания и пояснения:

Основные параметры структуры таблицы Группы

Таблица 1.3

Имя поля	Ключевое поле	Тип данных	Размер поля
Код группы	Да	Счетчик	Длинн.целое
Буквы		Короткий текст	5
Курс-номер		Числовой	Целое
Староста		Короткий текст	50
Бюджет/дог.		Логический	Да/нет

Создание составного индекса по таблице Группы

В таблице **Группы** название группы, которое должно быть уникальным, состоит из двух полей: **Буквы** и **Курс-Номер**. Для проверки уникальности названия и для ускорения поиска по составному названию группы создадим составной индекс по двум полям таблицы. Для этого следует выполнить команду **Индексы**. В появившемся окне имеется таблица для создания индекса.



Рис. 1.3

Определение индекса будет занимать в ней две строки. Имя индекса задать только в первой строке. В свойствах индекса обязательно задать **Уникальный индекс**.

Теперь в базе данных не смогут появиться две группы с одинаковым названием, т.к. уникальность будет проверяться автоматически.

Создание поля подстановки в таблице Группы

Освоим еще один типовой момент в структуре таблицы: **поля подстановки**. Это такие поля, значения которых не вводятся с клавиатуры, а выбираются из заранее заданного списка значений. Чаще всего список значений берется из какой-либо другой таблицы, которая называется подстановочной. В таблице **Группы** есть одно такое поле, которое удобно сделать полем подстановки: это **Буквы специальности** в названии группы, которые можно выбирать из таблицы **Специальности**.

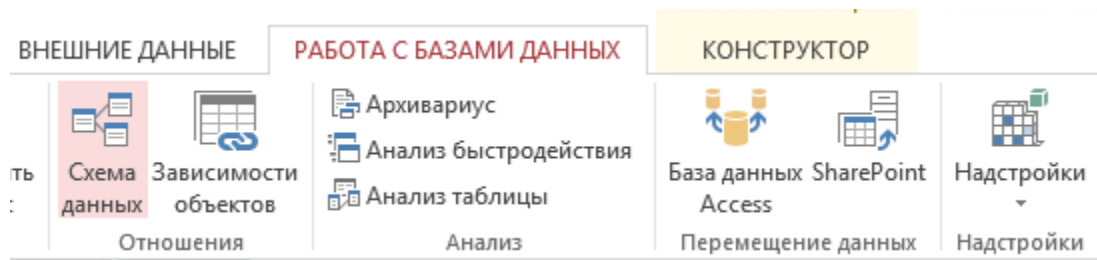
Откройте таблицу **Группы** в режиме Конструктора и поле **Буквы специальности** сделайте **Поле подстановки**. Для этого в графе **Тип данных** выберите **Мастер подстановок**, в окне **Создание подстановки** выберите таблицу **Специальности**, выбранные поля - **Буквы**.

Откройте таблицу **Группы** в режиме таблицы и убедитесь, что поле подстановки создано, т.е. в поле **Буквы специальности** появился выпадающий список с аббревиатурами специальностей.

Можете ввести данные по одной группе, однако заполнение этой таблицы также отложим до следующей работы.

Создание схемы данных

Хотя созданы еще не все таблицы базы данных, уже можно определить связи между созданными таблицами. В Access для этих целей используется специальное окно, которое называется **Схема данных**. Следует знать, что в этом окне изображается **физическая схема базы данных**, в отличие от рис. 1.1, на котором изображена **логическая схема**.



Задание: создайте схему данных, включив в нее созданные таблицы.

Указания и пояснения:

Создание схемы данных начинается в окне базы данных с нажатия кнопки **Схема данных** на вкладке **Работа с базами данных**, после чего открывается окно с инструментом **Отобразить таблицу**, в котором выбирают таблицы и запросы для включения в схему данных, выделяя их и нажимая кнопку **Добавить**. После включения всех нужных таблиц нажимают кнопку **Заккрыть**. В результате в окне **Схема данных** будут представлены таблицы **Студенты**, **Группы**, **Специальности** со списками своих полей. Далее можно приступать к определению связей между ними. **ПЕРЕД УСТАНОВКОЙ СВЯЗЕЙ, УБЕДИТЕСЬ, ЧТО ВСЕ ТАБЛИЦЫ ЗАКРЫТЫ**

Установим *связь между таблицами Группы и Студенты* по ключу **Код группы**. В окне **Схема данных** установите курсор на ключевое поле главной таблицы **Группы** и перетащите его на поле **Код группы** подчиненной таблицы **Студенты**. В открывшемся окне **Связи** в строке Тип отношения установится **один-ко-многим**, установите флажок параметра **Обеспечение целостности данных** и нажмите кнопку **Создать**.

Установите флажок **Каскадное удаление**. Это приведет к тому, что при удалении группы автоматически будут удаляться и все студенты этой группы. Флажок **Каскадное обновление** можно не устанавливать, т.к. изменять **Код группы**, имеющий тип данных **Счетчик**, все равно нельзя.

Между таблицами будет установлена связь, обозначенная на схеме 1 : ∞.

Установите связь между таблицами **Специальности** (главная таблица) и **Группы** (подчиненная таблица) по ключу **Буквы**. Установите флажок **Обеспечение целостности данных**, **Каскадное удаление** и **Каскадное обновление**.

По мере создания новых таблиц схема данных будет дополняться.

ЛАБОРАТОРНАЯ РАБОТА №2

Цель работы: освоить разработку простых и составных экранных форм, закрепить навыки создания таблиц и работы со схемой данных.

Порядок выполнения работы

Вспомним, что созданные таблицы пока пусты (или почти пусты). Пора позаботиться об удобном способе их заполнения. Прочитайте внимательно пояснения.

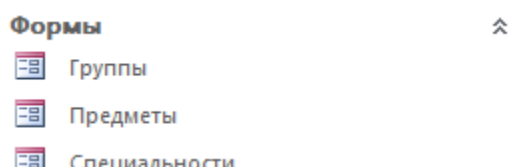
Назначение экранных форм. Способы разработки форм

Можно заполнить любую таблицу, используя встроенные средства Access (двойной щелчок по имени таблицы).

Однако заполнять или просматривать таблицы, используя встроенные средства Access, не всегда удобно, т.к. реальные таблицы обычно содержат большое количество столбцов самых разнообразных типов, которые целиком на экране не умещаются.

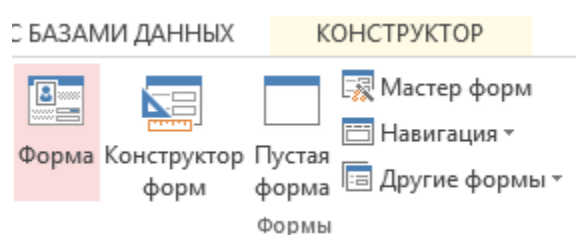
При этом следует помнить, что занесение первичной информации в базу данных — очень ответственная задача, для выполнения которой нужно предоставить пользователю максимум

удобств. Поэтому для удобства просмотра, занесения и корректировки данных таблиц создаются **экранные формы**.



Форма представляет собой диалоговое окно, в котором данные одной или нескольких таблиц отображаются в удобном для восприятия виде. Не следует путать форму с таблицей: таблица — это хранилище данных, а форма — средство отображения и редактирования данных.

Однотабличная форма (на основе одной таблицы) может быть создана пользователем в режиме конструктора форм или с помощью мастера. В первом случае создание начинается с пустой формы, и конструирование полностью возлагается на разработчика. Для создания однотабличной формы целесообразно использовать **Мастер форм**.



В этом случае весь процесс сводится только к выбору полей, размещаемых на форме и способа размещения полей на форме (имеется 3 основных типа форм: в столбец, ленточная, табличная).

В данных методических указаниях рассмотрим создание форм в режиме конструктора форм. Это более сложный процесс, чем использование мастера, однако он предоставляет разработчику любые возможности по оформлению форм.

Заметим, что можно использовать и такой способ создания форм: первоначально форма создается при помощи мастера, а затем дорабатывается в конструкторе. В дальнейшем выбирайте любой удобный для вас способ разработки, приводящий к нужному результату.

Задание: создайте однотабличную форму **Специальности**, предназначенную для удобного просмотра, ввода и редактирования данных таблицы **Специальности**.

Указания и пояснения:

Чтобы начать создание формы, нужно в закладке **Создание**, ленты инструментов, выбрать режим создания формы – **Конструктор форм** и выбрать в конструкторе **Добавить поля**, затем нажать Показать все таблицы, развернуть таблицу **Специальности** и перетащить все поля на форму

Вид формы приведен на рис.1.3.

Форма специальности

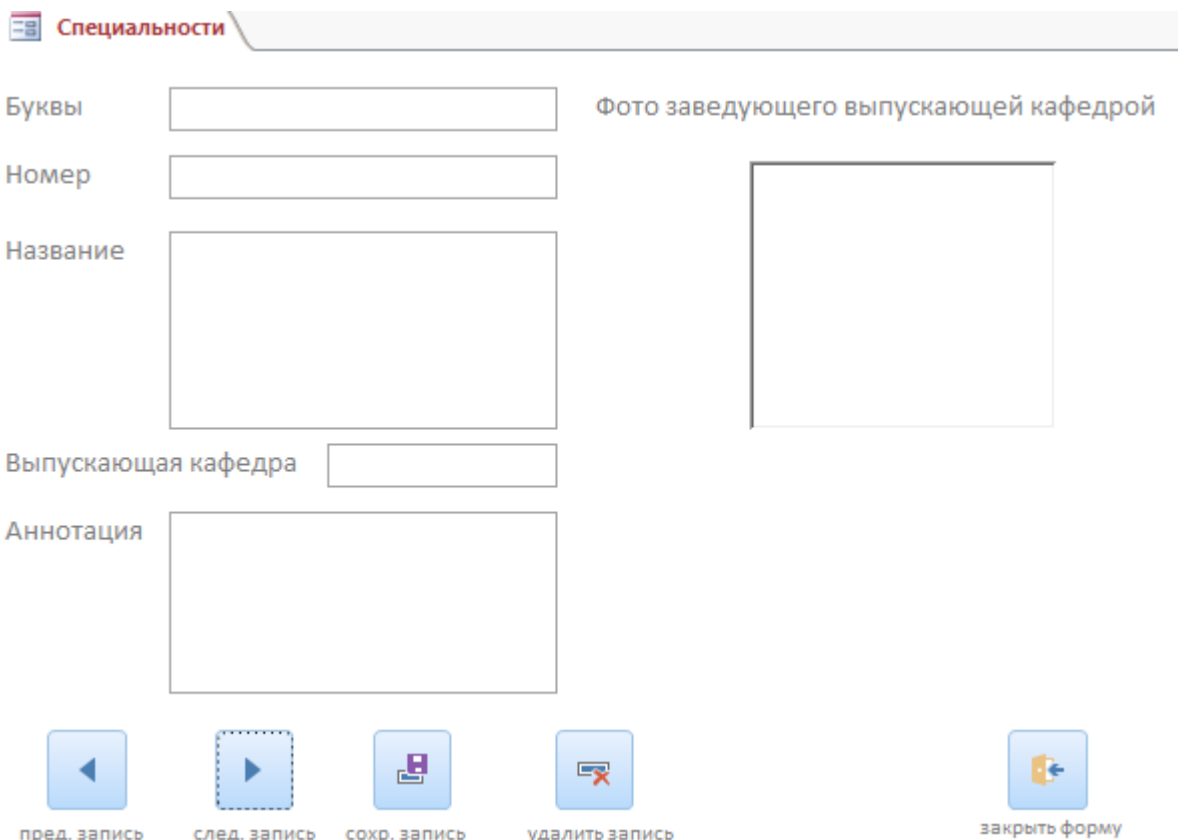


Рис 1.3

Измените свойства макета формы. Для этих целей в Access служит **Окно свойств**, в котором можно просматривать и редактировать свойства как всей формы в целом, так и ее отдельных элементов. Для появления окна свойств необходимо выделить необходимый элемент формы и в контекстном меню (правая кнопка мыши) выбрать **Свойства**. Для того, чтобы выделить всю форму в целом, а не ее конкретный элемент, щелкните правой кнопкой на небольшом квадрате в левом верхнем углу окна формы.

Изменим такие свойства макета формы: **Кнопки Навигации – нет, Область выделения – нет**, из кнопок оставьте только кнопку закрытия.

Еще одним важным элементом при конструировании форм является **ЭЛЕМЕНТЫ УПРАВЛЕНИЯ**, которая содержит пиктограммы с изображением элементов, которые можно помещать на форму (набор строительных материалов).

Поместите на форму пять кнопок и пять надписей к ним, пользуясь панелью элементов (две кнопки для перехода от одной специальности к другой, кнопки для добавления новой специальности и удаления специальности и кнопка для закрытия формы). На рис. 1.3 есть еще две кнопки, которые будут добавлены позже.

Все кнопки можно создать *мастером*, если на панели элементов нажата кнопка **Мастера**. При этом мастер предлагает выбрать действие, которое должно выполняться при нажатии кнопки и внешний вид этой кнопки.

Создание кнопки без использования Мастера — дело непростое, т.к. Мастер автоматически формирует процедуру на языке Visual Basic, которая будет вызываться при нажатии этой кнопки. Текст процедуры можно увидеть, если в окне свойств кнопки выбрать вкладку **События**, а затем в списке событий найти **Нажатие кнопки**.

Задание: создайте самостоятельно простую однотабличную форму **Студенты**, включающую следующие поля: № зачетки, ФИО, Дата рождения, Адрес.

В окне **свойств формы** на вкладке **Макет** установите *Режим по умолчанию – Режим таблицы*. Можно создать форму **Студенты** и с помощью Мастера или автоформы, при этом выбираем вид формы *Табличный*.

Не нужно стремиться улучшить внешний вид этой формы, т.к. это вспомогательная форма, которая будет использоваться в составной форме **Группы**.

Создание составной формы Группы

Составная форма — это удобный способ обеспечить одновременное заполнение двух связанных таблиц (можно соединить в одной форме и более двух таблиц, но такой прием используется реже). В нашем случае удобно создать составную форму, позволяющую просматривать и заносить данные о студентах по группам. Назовем ее **Группы**.

Задание: создать составную форму **Группы** на основе двух таблиц: **Группы** и **Студенты**.

Указания и пояснения:

Форма будет содержать *основную часть* на основе таблицы **Группы** и *подчиненную форму* **Студенты**

Форма группы

№ зачетки	ФИО студента	Дата рождения	Адрес
23461212	Иванов	11.12.1986	Горького 12-11
35432134	Петров	04.11.1990	Разина 15-12
16236624	Сидоров	05.09.1992	Петина 8-15
34271543	Соколов	08.04.1995	Северная 11-45
*			

Рис 2.1

В окне Конструктора форм разработайте основную часть формы на основе таблицы **Группы**, включите необходимые поля и разместите поля так, как показано на рисунке.

Обратите внимание, что для поля **Буквы группы** автоматически создано поле со списком, т.о. при занесении данных буквы будем выбирать из списка.

Поле **Код группы** на форме можно не показывать, т.к. поля, имеющие тип **Счетчик**, заполняются автоматически.

Для добавления в форму **Группы** подчиненной формы **Студенты**, необходимо поместить на форму элемент управления **Подчиненная форма/отчет**. При этом мастер попросит вас выбрать подчиненную форму, выбираем созданную ранее форму **Студенты**. Далее мастер сам предложит вам поле связи **Код группы** (поскольку это общее поле, которое связывает таблицы **Группы** и **Студенты**). После того, как вы подтвердите этот выбор, связь между основной и подчиненной формой устанавливается автоматически.

Поле **Код группы** (внешний ключ или поле связи таблицы **Студенты**) можно не отображать в подчиненной форме, т.к. *при заполнении составной формы поле связи в подчиненной таблице заполняется автоматически*. Собственно, для этого и создаются составные формы. Таким образом, получилось, что служебное поле **Код группы** вообще будет заполняться незаметно для пользователя, однако разработчик не должен забывать о существовании этого поля. В будущем оно еще понадобится.

Заполнение таблиц Группы и Студенты при помощи формы Группы

Перейдите из режима конструктора в режим **Формы** и занесите данные в таблицы **Группы** и **Студенты** при помощи созданной формы (3-4 группы, в каждой по 3-4 студента). Обратите внимание, что список с буквами специальностей предназначен для занесения данных в таблицу **Группы**, а не для поиска нужной группы и ее студентов.

Добавление в форму Группы списка для быстрого поиска Группы

Вспомним, что одной из функций разрабатываемой системы является быстрый поиск информации. Дополним форму **Группы** списком для поиска, который позволит быстро находить нужную группу со всеми ее студентами.

В Access 2016 эта задача решается просто. Поместите в нижнюю часть формы элемент **поле со списком** и рядом с ним надпись «Найти группу». Выберите таблицу Группы В качестве полей для поиска выберите **Буквы и Курс-номер**.

.В некоторых версиях Access нажатие кнопки может не отрабатывать. Следует прописать макрос при нажатии кнопки. Сделать это можно следующим образом: в свойствах списка, в закладке **события**, выбрать **нажатие кнопки** затем выбрать в построителе **Макрос**, из выпадающего списка выбрать **ПоискЗаписи** и в строке условие отбора прописать :

= "[Код группы] = " & Str(Nz([Screen].[ActiveControl];0))

Либо Перейдите в режим формы и убедитесь, что быстрый поиск группы выполняется правильно

Создание таблиц Предметы, Учебный план и Успеваемость

Задание: самостоятельно создать структуру таблиц **Предметы, Учебный план и Успеваемость** и включить их в **Схему данных**, указав все связи.

Основные параметры структуры таблицы Предметы

Таблица 2.1

Имя поля	Ключевое поле	Тип данных	Размер поля
Код предмета	Да	Счетчик	Длинное целое
Название предмета		Короткий текст	25

По полю **Код предмета** рекомендуется создать *уникальный индекс*.

Основные параметры структуры таблицы Учебный план

Таблица 2.2

Имя поля	Ключевое поле	Тип данных	Размер поля
Буквы	Да	Короткий текст	5
Код предмета	Да	Числовой	Длинное целое
Обязательный		Логический	Да/нет
Колич. Часов		Числовой	Целое
Экзамен/зачет		Логический	Да/нет

Поле **Код предмета** в таблице **Учебный план** сделайте полем подстановки, чтобы можно было выбирать название предмета из списка. При этом реально в таблицу будет заноситься код предмета. Таким образом, опять скрываем от пользователя существование поля **Код предмета**. Сами при этом не забываем, что в таблице хранится код, а не название предмета.

Основные параметры структуры таблицы Успеваемость

Таблица 2.3

Имя поля	Ключевое поле	Тип данных	Размер поля
Код предмета	Да	Числовой	Длинное целое
№ зачетки	Да	Короткий текст	8
Оценка		Числовой	Байт

Можно не делать **Код предмета** полем подстановки, т.к. заполнение и просмотр таблицы **Успеваемость** — особая задача, к которой мы еще вернемся.

Дополните самостоятельно схему данных, установив все необходимые связи (см. лаб. работу №1). Не забудьте про обеспечение целостности.

ЛАБОРАТОРНАЯ РАБОТА №3

Цель работы: закрепление навыков создания простых и составных форм. Начальное знакомство с модулем формы и процедурами обработки событий.

Порядок выполнения работы

Задание: создайте составную форму **Учебный план** на основе таблиц **Специальности** и **Учебный план**. Поместите на нее кнопку, которая позволит редактировать **Справочник предметов**, не закрывая форму.

Указания:

1. Создайте простую табличную форму на основе таблицы **Учебный план**, которая будет использоваться в качестве подчиненной.
2. Создайте новую форму на основе таблицы **Специальности**. Можно поместить на форму только поля **Буквы** и **Название специальности**, сделав их недоступными для изменения. Для этого нужно установить **блокировку** в свойствах поля на вкладке **Данные**.
3. Поместите на форму подчиненную форму на основе таблицы **Учебный план**, которая выдает информацию обо всех предметах данной специальности

Добавление справочника предметов

Создайте простую форму **Предметы**, установив свойство **Режим по умолчанию – Режим таблицы**. Она будет использоваться для добавления нового предмета или изменения названия имеющегося предмета. Поместите на форму **Учебный план** кнопку “Справочник предметов”, при нажатии которой будет открыта форма **Предметы**.

Составная форма Учебный план

Код предмета	Обязательный	Колич Часов	Экзамен/за
Информатика	<input checked="" type="checkbox"/>	90	<input checked="" type="checkbox"/>
История	<input checked="" type="checkbox"/>	100	<input checked="" type="checkbox"/>
Математика	<input checked="" type="checkbox"/>	90	<input checked="" type="checkbox"/>
Психология	<input type="checkbox"/>	50	<input checked="" type="checkbox"/>
*	<input type="checkbox"/>	0	<input type="checkbox"/>

Рис 3.1

Вы, наверное, заметили, что справочник всех предметов открывается в режиме простой формы, а не таблицы (хотя при создании вы задавали режим таблицы). Для такого справочника это неудачная форма представления данных. Чтобы исправить положение, нужно открыть текст обработчика нажатия кнопки «Справочник предметов». Дело в том, что когда кнопка создается при помощи мастера, Access автоматически формирует для нее процедуру на языке Visual Basic, которая называется обработчиком события **Нажатие кнопки**. Все обработчики событий для одной формы помещаются в один программный модуль, который называется модулем формы. Для каждой формы создается свой модуль.

В свойствах кнопки на вкладке **События** напротив **Нажатия кнопки** вы увидите кнопку с многоточием. Нажав ее, вы откроете нужный нам обработчик. Там вы увидите строку **Режим** Смените его на режим **Таблица**

Ввод данных

Перейдите в режим формы и заполните учебный план данными для 2-3 специальностей. Предварительно заполните справочник предметов. Пример заполненной формы для одной из специальностей показан на рис. 3.1.

Связывание формы Специальности с формами Учебный план и Группы

Формы **Специальности**, **Группы** и **Учебный план** могут открываться независимо друг от друга, но было бы удобным для пользователя как-то связать эти формы друг с другом. Сделаем так. Поместите на форму **Специальности** кнопку с названием **Учебный план**, при нажатии которой будет открываться разработанная нами форма. Для того, чтобы в форме **Учебный план** отображался учебный план текущей специальности, при создании кнопки мастером установите переключатель **Отображать отобранные записи**, а в качестве поля для отбора выберите **Буквы**.

В результате получилось две основных формы для просмотра, ввода и редактирования данных:

- **Специальности**, из которой открывается учебный план данной специальности, а заодно и справочник всех предметов,
- **Группы**, которая содержит также все сведения о студентах групп.

Задание: чтобы связать между собой и эти две формы, поместите на форму **Специальности** кнопку **Группы** (эта кнопка создается аналогично кнопке **Учебный план**).

Теперь всю необходимую информацию о специальностях, группах, студентах можно получить, не выходя из формы **Специальности**.

Осталось только разработать форму для ввода и просмотра оценок студентов. Она сложнее других, поэтому сначала разберем технику построения запросов.

ЛАБОРАТОРНАЯ РАБОТА №4

Цель работы: освоение техники построения запросов на выборку с использованием конструктора запросов.

Порядок выполнения работы

Прочитайте внимательно следующие пояснения .

Запросы — основное средство обработки данных в базах данных.

При помощи запросов можно выполнять следующие действия:

- поиск нужной информации по одной или нескольким таблицам и анализ данных (запрос на выборку);
- создание новой таблицы для сохранения результатов запроса на выборку (запросы на создание);
- добавление новых строк в таблицу (запросы на добавление);
- удаление строк из таблицы (запросы на удаление);
- изменение значений уже имеющихся строк (запросы на обновление).

Запросы хранятся в базе данных в виде текста, написанного на языке запросов SQL, однако для разработки запросов в Access не требуется знания языка SQL. Дело в том, что имеющийся в Access конструктор запросов *автоматически формирует текст на языке SQL*. При этом пользователь формирует запрос, заполняя специальный бланк, который называется *бланком запроса по образцу (QBE — Query By Example)*. С его помощью можно сформировать запрос простыми приемами.

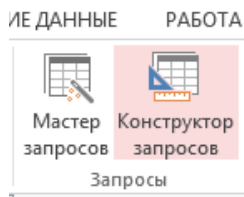
Сформированный текст на языке SQL всегда можно просмотреть из конструктора, выполнив щелкнув правой кнопкой мыши и выбрав пункт **Режим SQL**. При выполнении работы рекомендуется просмотреть полученные SQL-тексты в целях ознакомления с языком SQL. При сохранении запроса в базе данных сохраняется именно текст SQL.

Самые простые наиболее часто используемые - **запросы на выборку**. С них и принято начинать знакомство с созданием запросов. Цель запроса на выборку состоит в создании результирующей таблицы, в которой отображаются нужные по условию запроса данные из базовых таблиц.

Сформированная в результате запроса таблица не сохраняется на диске, однако остается в оперативной памяти до тех пор, пока не будет закрыта. Каждый раз при открытии запроса (кнопка **Открыть**) результирующая таблица создается заново путем выполнения команды SQL, которая была сохранена в базе данных.

Указания по созданию запросов:

Создание запроса начинается с открытия инструмента **Конструктор запросов** в ленте инструментов на вкладке **Создание**.



Затем в диалоговом окне **Добавление таблицы** выбирают те таблицы, на которых будет основан запрос. Выбранные таблицы заносят в верхнюю половину бланка щелчком на кнопке **Добавить**.

В окне **Добавление таблицы** имеется три вкладки: **Таблицы**, **Запросы**, **Таблицы и запросы**. Это означает, что запрос не обязательно основывать только на таблицах, если ранее уже был создан запрос, то новый запрос можно основывать и на нем.

Бланк запроса по образцу имеет две панели: на верхней расположены таблицы, на которых основывается запрос, с указанием связей между ними; строки нижней панели определяют структуру запроса.

Строку **Поле** заполняют перетаскиванием названий полей из таблиц в верхней части бланка, строка **Имя таблицы** заполняется автоматически при перетаскивании поля. Строка **Сортировка** позволяет отсортировать результат запроса по данному полю (по возрастанию или убыванию). Бывают случаи, когда поле должно присутствовать в бланке запроса по образцу, но не должно отображаться в результирующей таблице. В этом случае можно запретить его

Вывод на экран, сбросив соответствующий флажок. Самая интересная строка в бланке запроса по образцу - **Условие отбора**. Здесь записывают тот критерий, по которому выбирают записи для включения в результирующую таблицу. По каждому полю можно создать свое условие отбора, при этом между отдельными условиями будет выполняться операция «И». Если требуется выполнить операцию «ИЛИ», то одно из условий записывают в строке «ИЛИ»

Запуск запроса выполняют щелчком на кнопке **Выполнить** (или команда меню **Режим таблицы**), при запуске образуется результирующая таблица. Чтобы выйти из результирующей таблицы и вернуться к созданию запроса, нужно выбрать команду **Конструктор**.

Например, рассмотрим процесс создания запроса, который выводит список всех студентов группы ЭМ-51.

Сначала определим, какие таблицы необходимы для выполнения запроса. В нашем случае это **Группы** и **Студенты**. Связь между ними установится автоматически, т.к. она указана в схеме данных. Далее определим, какие поля нужно отображать на экране (**ФИО**), а какие потребуются для формирования условия отбора (**Буквы** и **Курс-номер**). Задаем значения для отбора **ЭМ** и **51**. В данном случае эти значения можно ввести вручную. Заполненный бланк для запроса изображен на рисунке.

Пример заполнения бланка запроса

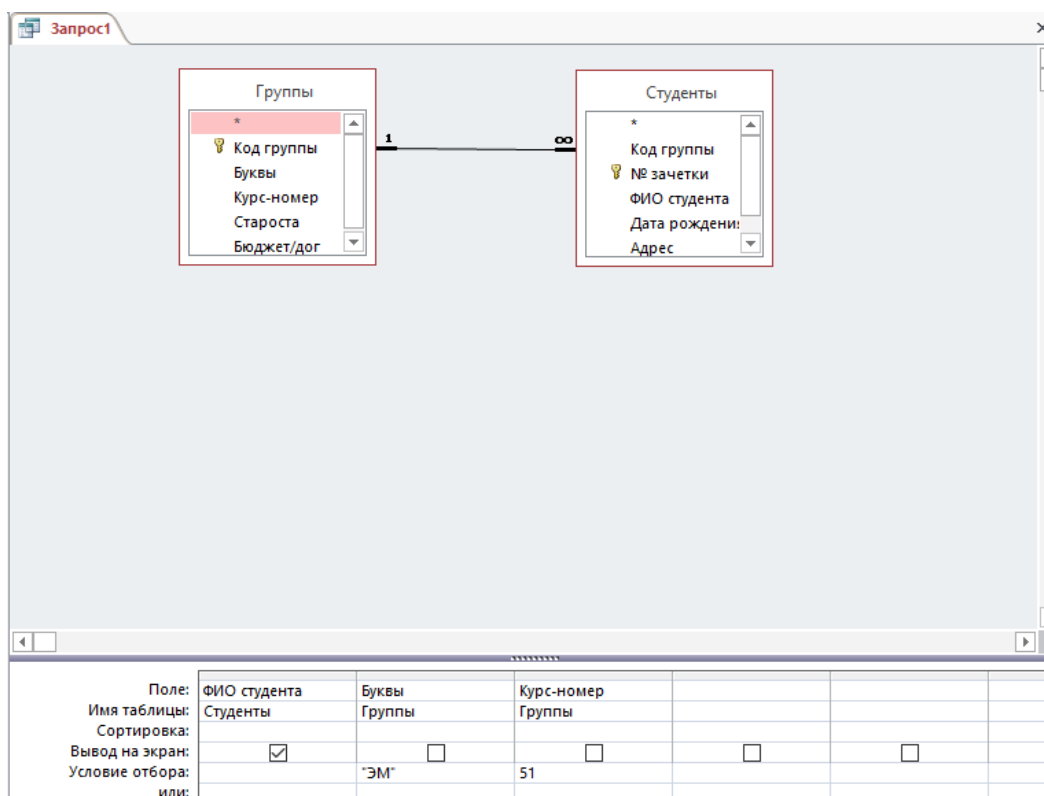


Рис. 4.1

Обратите внимание:

Условие отбора записей и другие выражения можно формировать при помощи **Построителя выражений** (команда **Построить** контекстного меню). **Построитель выражений**, имеющийся в Access, очень прост и удобен. Он позволит избежать ошибок и описок при формировании выражений. Рекомендуем его использовать.

Задание: создайте запросы на выборку:

Список всех старост первого курса.

1. Список всех студентов второго курса (ФИО).
2. Список всех студентов второго курса, у которых фамилии начинаются на букву «А» (ФИО и № зачетки).
3. Список всех студентов второго курса, у которых фамилии начинаются на букву «А» (ФИО, № зачетки, с указанием названий группы для каждого студента)
4. Вывести список предметов специальности ЭМ, которые или обязательные, или по ним зачет.

Запросы с вычисляемыми полями

Эти запросы позволяют создавать новые поля на основе значений других полей таблицы. Для создания вычисляемого поля следует в строке **Поле** Бланка запросов задать выражение для вычислений (вместо того, чтобы перетаскивать туда поле какой-либо таблицы). При этом настоятельно рекомендуется пользоваться инструментом **Построитель** в режиме **Конструктор**.

Задание: создайте запрос на выборку: Список всех студентов с указанием возраста в годах.

Указание: воспользуйтесь функцией *DateDiff()*, которая использует разницу дат. Используйте **Построитель**.

Поле:	ФИО студента	Выражение1: DateDiff('yyyy';[Дата рождения];Дата0)
Имя таблицы:	Студенты	
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Like "**"	
или:		

Запросы с параметрами

Во многих случаях пользователю надо предоставить возможность выбора того, что он хочет найти в таблицах базы данных. Для этого существует специальный вид запроса - *запрос с параметрами*. Параметры запроса заключаются в квадратные скобки и обычно помещаются в **Условие отбора** в том поле, по которому производится выбор. Иногда параметры помещаются в выражение для вычисляемого поля.

После запуска запроса открывается диалоговое окно, в котором предлагается ввести **параметр**. Если в запросе несколько параметров, то последовательно открывается несколько окон для ввода параметров.

Задание: создайте запросы на выборку с параметром:

1. Все сведения о заданном студенте (вводится ФИО).
2. Список предметов (названия), которые изучает заданный студент (вводится ФИО).
3. Для заданной группы (вводим буквы и курс-номер) все предметы, по которым экзамен.
4. Для заданной специальности все обязательные предметы, по которым зачет.

Запросы на выборку с группировкой

Запросы позволяют не только отбирать нужную информацию из таблиц и обрабатывать ее путем создания новых (вычисляемых) полей, но и производить так называемые *итоговые вычисления*. Итоговые функции существуют только для *группы записей*, поэтому записи надо предварительно *сгруппировать* по какому-либо полю. В этом случае формируются группы с одинаковыми значениями в этом поле. Например, сгруппировав таблицу **Студенты** по полю **Код группы**, получим столько групп, сколько всего имеется студенческих групп на факультете. Таким способом можно подсчитать количество студентов в каждой группе, средний возраст студентов группы и другие итоговые характеристики. Если не выбрать поле для группировки, то итоговые вычисления будут выполняться по всей таблице.

Итоговые запросы создают на основе известного бланка запроса по образцу, только в нем появляется дополнительная строка - Группировка. Для введения этой строки в бланк надо

щелкнуть на ленте инструментов конструктора на элемент **Итоги**



Задание:

Создайте запросы на выборку с группировкой:

1. Список всех групп с указанием количества студентов в каждой группе.
2. То же самое для специальностей.
3. Подсчитать общее количество студентов на всем факультете.
4. Общее и среднее количество часов по всем предметам для заданной специальности, предмет с самым большим и самым маленьким количеством часов.

ЛАБОРАТОРНАЯ РАБОТА №5

Цель работы: освоить создание запросов на создание, добавление, удаление и обновление

Порядок выполнения работы

Прочитайте внимательно следующие пояснения.

Все виды запросов на выборку создают временные результирующие таблицы, базовые таблицы при этом не изменяются. Тем не менее, существует особая группа запросов, которые называются *запросами на изменение (или запросами действия)*. Они позволяют автоматически создавать новые таблицы и изменять уже имеющиеся. К этому виду запросов относятся: *запросы на создание таблицы, на обновление данных, на добавление записей и на удаление записей*.

Запросы действия приводят к необратимым изменениям в базе данных, поэтому следует относиться к ним с особой осторожностью.

Перед выполнением данной работы рекомендуется сделать резервную копию базы данных на случай нежелательных изменений в таблицах.

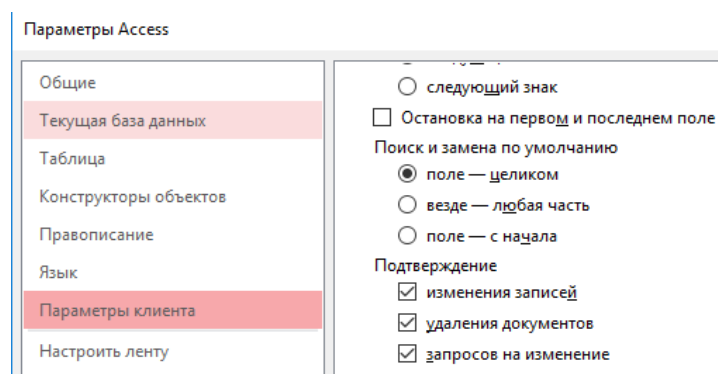
Указания по созданию запросов действия:

Все запросы действия строятся на основе запросов на выборку, т.е. сначала выполняется отбор данных, которые будут добавлены, или удалены, или изменены. Запрос на выборку рекомендуется запустить на выполнение и убедиться, что данные отбираются правильно. Только после этого преобразуем запрос на выборку в конкретный запрос действия, используя соответствующую команду контекстного меню.

Задание:

- 1.Создайте таблицу Архив, которая должна иметь такую же структуру, как таблица Студенты и дополнительно Буквы специальности .
- 2.Создайте запрос на обновление – перевод студентов на следующий курс. (У каждой группы прибавить к номеру группы 10).
- 3.Создайте запрос на добавление: списки студентов с указанием их специальности, закончивших институт, добавить в таблицу Архив (тех студентов, у которых курс-номер > 60)
- 4.Создайте запрос на удаление: удалить списки этих же студентов из таблицы **Студенты**.
- 5.Создайте запрос на обновление: из цифр группы, которые > 60, вычесть 50, и у этих же групп очистить поле «Староста».

Обратите внимание, что при запуске запросов действия Access обязательно выдает запросы на подтверждение выполняемых действий. Выдачу окон подтверждения можно отменить, если в Окне параметров Базы данных (**Файл/Параметры**) на вкладке **Параметры клиента** снять флажки **Подтверждение запросов на изменение** и **Подтверждение изменения записей**.



Однако не советуем снимать подтверждения, т.к. они представляют замечательную возможность отказаться от нежелательных изменений в базе данных в самый последний момент.

Создание макроса «Перевод на следующий курс»

Нетрудно заметить, что только что разработанные запросы решают одну общую задачу — перевод студентов на следующий курс. Так, запросы со второго по пятый должны последовательно выполняться в конце каждого учебного года. Создадим макрос, который обеспечит последовательный запуск на выполнение этих запросов.

Макросы — это простое средство программирования в Access. Конструктор макросов позволяет разрабатывать макросы путем выбора команд из списка.

Задание: создайте макрос «Перевод на следующий курс»

Указания: перейдите на вкладку **Макросы** и щелкните по кнопке **Создать**. В конструкторе макроса выберите четыре раза одну и ту же команду **Открыть запрос**, в качестве параметра выбирайте из списка те запросы, которые нужно выполнить.

Сохраните макрос. Запустите его на выполнение и проверьте, что все действия выполняются правильно.

Создание запроса для автоматического добавления строк в таблицу Успеваемость

Напомним, что таблица **Успеваемость** содержит три поля - № зачетки, код предмета и оценка данного студента по данному предмету. Особенность заполнения этой таблицы состоит в том, что новые строки в ней можно формировать при помощи запроса на добавление. Поскольку экзамены сдаются по группам, то удобно в день экзамена добавить в таблицу **Успеваемость** записи, соответствующие студентам группы, сдающей экзамен. При этом автоматически заполнятся поля **№ зачетки** и **Код предмета** (по которому экзамен), а поле **Оценка**, естественно, придется заполнять вручную после экзамена.

Задание: создайте запрос на добавление в таблицу **Успеваемость** с двумя параметрами: **Код группы** и **Код предмета**.

Указания: для этого запроса нужны таблицы: **Студенты**, **Учебный план**, **Группы** и **Специальности**. В бланк запроса помещаются поля **№ зачетки** из таблицы **Студенты**, **Код группы** из таблицы **Группы** и **Код предмета** из таблицы **Учебный план**. В **Код группы** и **Код предмета** поставьте переменные параметры в условии отбора. Параметры назовем **[Группа]** и **[Предмет]** – не забудьте квадратные скобки!

Проверьте в режиме таблицы, что записи выбираются правильно, для этого введите любые допустимые значения параметров (код группы и код предмета, который входит в учебный план специальности для данной группы).

Затем выберите тип запроса из меню **Запрос**: запрос на добавление. После того, как вы введете имя таблицы **Успеваемость**, в которую нужно добавлять данные, в бланке запроса появится еще одна строка с именами полей, в которые будут добавляться данные.

Запустите запрос на выполнение и убедитесь, что в таблице **Успеваемость** появились записи с номерами зачетов студентов именно той группы, код которой вы вводили в качестве параметра. Значит, все правильно.

Но вводить код группы и код предмета очень неудобно, тем более, что в предыдущих формах удалось полностью скрыть от пользователя эти служебные поля. Создадим форму, которая позволит выбирать параметры запроса (Группу и Предмет), пользуясь раскрывающимися списками.

ЛАБОРАТОРНАЯ РАБОТА №6

Цель работы: познакомиться с созданием более сложных форм, предоставляющих пользователю дополнительные удобства при вводе данных. Освоить создание перекрестного запроса при помощи Мастера.

Порядок выполнения работы

Создание формы для ввода успеваемости

Данная форма предназначена для заполнения одной таблицы **Успеваемость**, но удобно сделать ее составной формой. При этом на главной форме разместим списки для выбора группы и предмета и кнопку **Создать ведомость**, а подчиненная форма будет показывать записи из таблицы **Успеваемость**, соответствующие выбранной группе и предмету.

Вид формы должен быть похож на бланк экзаменационной ведомости (так удобнее и привычнее вводить оценки). Он изображен на рис. 6.1. Кнопка **Создать ведомость** служит для запуска запроса на добавление в таблицу **Успеваемость**, созданного в предыдущей работе. Воспользоваться этой кнопкой для заданной группы и заданного предмета можно будет только один раз, второй раз сама Access не позволит добавить лишние строки (т.к. будет нарушена уникальность ключа).

Подчиненная форма сделана на основе запроса на выборку из таблицы **Успеваемость**. Параметры запроса на выборку и запроса на добавление привязаны к спискам формы, поэтому оба запроса будут выполняться для выбранной группы и выбранного предмета.

Задание: разработайте запрос на выборку для подчиненной формы с двумя параметрами: код группы и код предмета.

Указание: для запроса на выборку нужны таблицы **Студенты** и **Успеваемость**. Поместите в бланк запроса поля **№ зачетки**, **ФИО**, **Оценка**, **Код группы** и **Код предмета** (последние 2 без вывода на экран, но в них задать параметры для условий отбора). Параметры нужно назвать так, как они назывались в запросе на добавление [Группа] и [Предмет]. Проверьте правильность работы запроса.

Задание: создайте составную форму и поместите на нее все изображенные на рисунке элементы, кроме двух кнопок в левом нижнем углу. У главной формы не должно быть никакого источника данных (не выбирайте никакую таблицу), т.к. эта форма служит только для выбора параметров запроса на добавление и запроса на выборку. В качестве источника записей для подчиненной формы выберите созданный запрос.

При создании списка для выбора группы, вероятно, возникнет проблема, как отобразить в списке сразу два поля: **Буквы** и **Курс-номер**. Мастер не умеет этого делать, поэтому после создания поля со списком мастером придется вручную доработать запрос для списка, созданный мастером (чтобы попасть в Конструктор этого запроса, в **Окне свойств** поля со списком на вкладке **Данные** щелкнуть по кнопке с многоточием напротив пункта **Источник записей**).

Рекомендуем создать вычисляемое поле для названия группы.

Форма Успеваемость

№ зачетки студента:	ФИО:	Оценка:
1212312	Смирнов В.В.	
324 5876	Титов Д.Н.	
3435654	Чистякова Т.Е.	
46347857	Говоров Д.В.	
5634633	Воскресенский С	
8234234	Симанков В.В.	
*		

Рис 6.1

Теперь осталось самое трудное – привязать параметры запросов к двум полям со списками. Для этого разумно изменить имена полей со списками на **Группа** и **Предмет** – имя элемента вы найдете в окне свойств на вкладке **Другие**. Войдите снова в бланки запросов и введите или постройте Построителем (он вызывается правой кнопкой мыши - Построить) полные имена полей со списком: **Forms!Успеваемость!Группа** вместо параметра [Группа] и **Forms!Успеваемость!Предмет** вместо параметра [Предмет].

Создайте макрос для кнопки **Создать ведомость**. При создании этой кнопки откажитесь от услуг Мастера, поэтому к кнопке не привязано никакого обработчика. Убедитесь в Окне свойств, что вкладка **События** целиком пустая. Нажимаем кнопку с многоточием напротив события **Нажатие кнопки**. В появившемся окне построителя выбираем **Макросы**.

Макрос будет содержать 2 команды: **Открыть запрос** (запрос на добавление, созданный вами до этого) и **Обновление** (аргументом команды должно быть имя подчиненной формы). Команда **Обновление** нужна для того, чтобы добавленные записи сразу отобразились в форме.

На событие **После обновления** обоих полей со списками назначить макрос, содержащий одну команду **Обновление**, чтобы при выборе из списков автоматически обновлялся запрос в подчиненной форме.

Запустите на выполнение. Введите успеваемость для нескольких групп по нескольким предметам.

Теперь все таблицы базы данных заполнены.

Перекрестные запросы

Пояснения: перекрестные запросы — это особая форма представления результатов запросов на выборку, которая является более наглядной и удобной для пользователя. В результате такого запроса формируется перекрестная таблица, в которой заголовки столбцов представляют собой значения какого-либо поля базы данных.

Создание перекрестных запросов в Access выполняется при помощи Мастера, поэтому обычно не вызывает затруднений.

Задание: Создать при помощи мастера запрос, в котором заголовками строк являются № зачетки студента и его ФИО, а заголовками столбцов – названия предметов. В каждой клетке – оценка определенного студента по определенному предмету. Если предмета нет в учебном плане, то клетка будет пустая. В деканате такой документ называют сводной ведомостью.

Перекрестный запрос (сводная ведомость успеваемости)

№ зачётки	ФИО студента	Базы данных	История	Математика	Программирование
236455	Иванов А.В.		3		4
324325	Николаев Н.А.			4	5
364563	Смирнов Б.В.		4		5
412421	Сорокин Р.Г.				
458890	Романов С.С.			5	4
485678	Николаев А.М.		3		5
623442	Петров И.А.		5		3
938476	Смирнова Н.А.	5		5	

Рис 6.2

Указание: сначала создайте обычный запрос на выборку, который содержал бы все необходимые поля. Убедились, что результаты такого запроса выглядят ненаглядно. Затем создайте мастером перекрестный запрос, в результате которого получите результат как на рис.6.2.

ЛАБОРАТОРНАЯ РАБОТА №7

Цель работы: освоить процесс создания отчетов при помощи конструктора отчетов.

Порядок выполнения работы

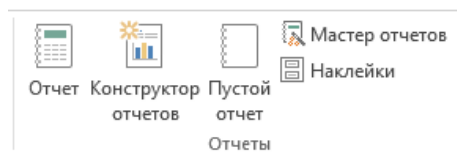
Задание: создание отчета Список студентов факультета по группам.

Указания и пояснения:

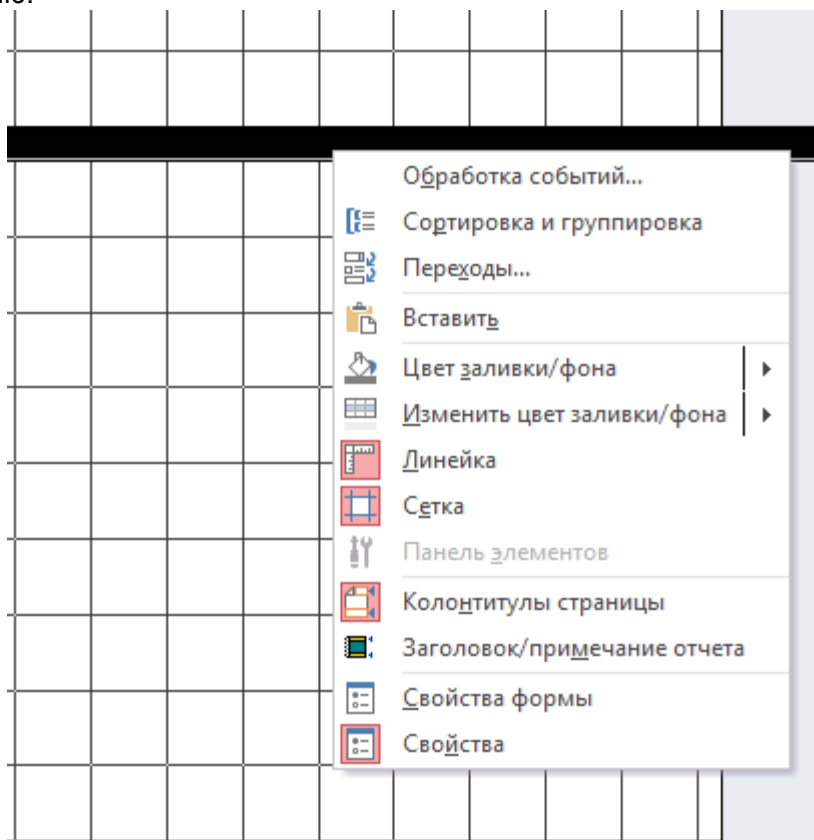
Отчет, также как и форма, должен быть основан на одной таблице или запросе. Поэтому, если требуется включить в отчет данные нескольких таблиц (это типичный случай), предварительно создают запрос на выборку, с помощью которого отбираются все необходимые для отчета данные.

Создадим *запрос на выборку* на основе таблиц **Группы** и **Студенты**, в структуру которого включим поля: *Код группы, Буквы, Курс-номер, № зачетки, ФИО*.

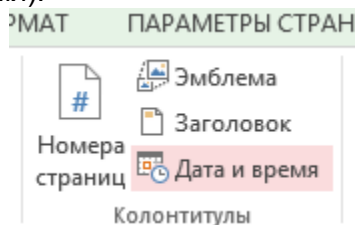
Далее нужно выполнить следующие действия: В линейке инструментов выбрать элемент **Конструктор отчетов**



В открывшемся окне конструктора отчет содержит разделы: **Заголовок отчета, Верхний колонтитул, Область данных, Нижний колонтитул, Примечание отчета**. Если отсутствует раздел **Заголовок отчета**, добавим его с помощью кнопки **Заголовок/Примечание отчета** в контекстном меню.

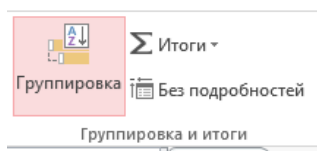


Разместим в **Заголовке отчета** название отчета - СПИСОК СТУДЕНТОВ НАШЕГО ФАКУЛЬТЕТА, используя кнопку **Надпись** на Панели элементов, а также включим в заголовок отчета текущую дату (Дата и время).



В **Верхний колонтитул** не включаем никаких надписей, а в **Нижний колонтитул** включим номер страницы. В **Область данных** размещаем **№ зачетки** и **ФИО**.

Поскольку общий список студентов должен быть разбит по группам, *выполним группировку* по полю **Код группы**. Для этого нужно нажать на кнопку **группировка** на панели инструментов конструктора

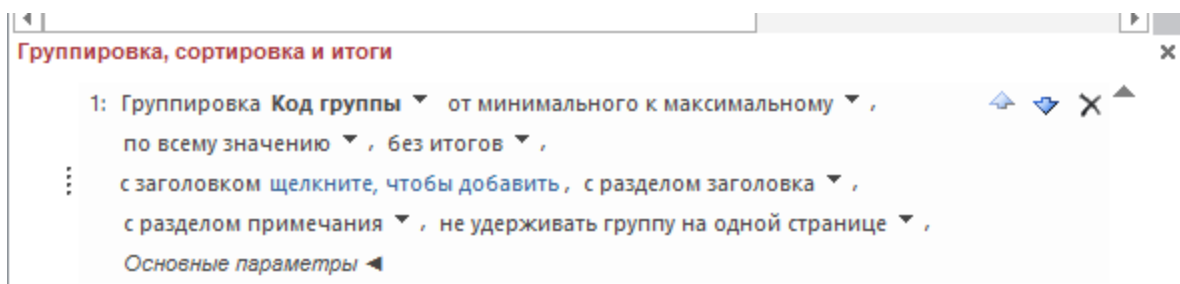


Добавить группировку Добавить сортировку

и заполнить поля открывшегося окна. В окне должно быть представлено поле **Код группы**, по которому определена группировка и поле **ФИО**, по которому определена только сортировка. Для этого необходимо в **Свойствах группы для поля Код группы** выполнить установки: **Заголовок группы - Да, Примечание группы - Да, а для поля ФИО - Заголовок группы - Нет, Примечание группы - Нет.**

The screenshot shows an Excel report window titled "Отчет1" with a grid of 12 columns and multiple rows. The report is structured into several sections:

- Заголовок отчета:** Contains the text "СПИСОК СТУДЕНТОВ НАШЕГО ФАКУЛЬТЕТА" and two rows with formulas "=Дата()" and "=Time()".
- Верхний колонтитул:** An empty row.
- Заголовок группы 'Код группы':** Contains a dropdown menu with "Буквы" selected and a text box with "Курс-номер".
- Область данных:** Contains two columns labeled "№ зачетки" and "ФИО студента".
- Нижний колонтитул:** Contains the formula "=""Страница "" & [Page]".
- Примечание отчета:** An empty row.



Задание: создайте самостоятельно отчеты:

1. Вкладыш к диплому (отпечатать вкладыши к дипломам всем студентам 5-го курса)
2. Экзаменационная ведомость (на форму Успеваемость поместите кнопку Предварительный просмотр отчета; параметры запроса для этого отчета привяжите к полям формы Группа и Предмет).

ЛАБОРАТОРНАЯ РАБОТА №8

Цель работы: завершить разработку информационной системы, связав в единое приложение все разработанные объекты базы данных. Знакомство с диспетчером кнопочных форм. Задание параметров запуска.

Порядок выполнения работы

Все созданные элементы приложения можно запускать отдельно из Окна базы данных, но это очень неудобно для пользователя. Стандартный способ связать воедино разрозненные части единой системы — создание основной формы с кнопками, при помощи которых будут запускаться на выполнение отдельные формы, отчеты, макросы и т.д.

Задание: создайте самостоятельно основную форму приложения, содержащую кнопки:

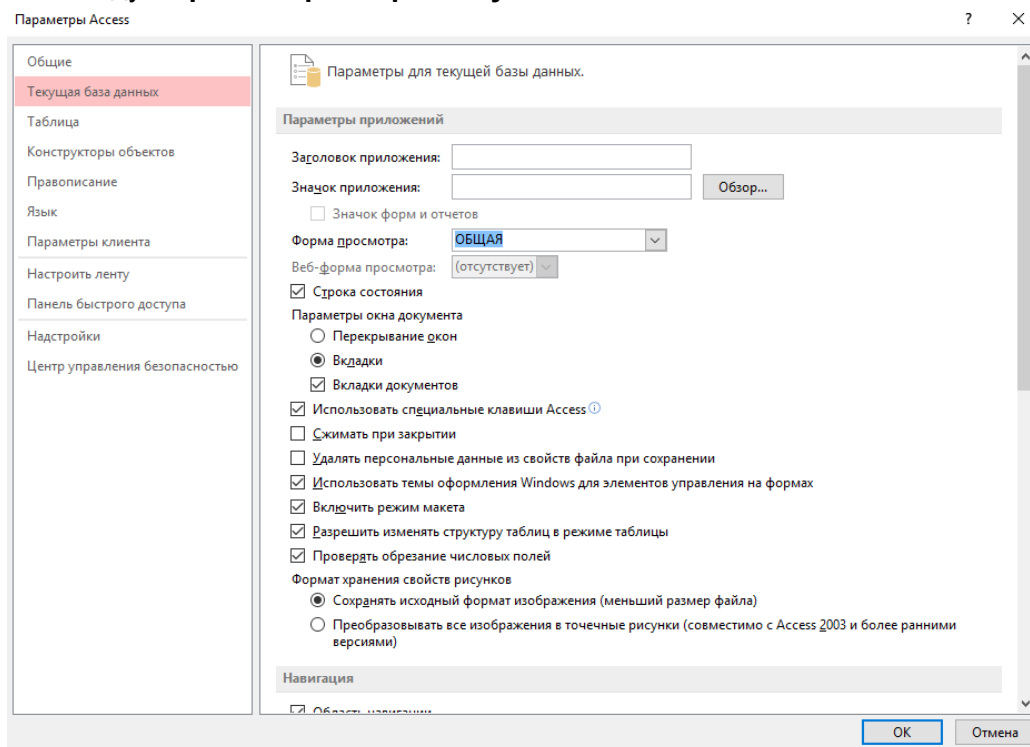
- Специальности (открывается форма Специальности)
- Студенты по группам (форма)
- Студенты по группам (отчет)
- Успеваемость (на ней должна быть кнопка для просмотра отчета Экзаменационная ведомость)
- Перевод на следующий курс (запуск макроса Перевод на следующий курс)
- Вкладыши к диплому
- Выход

Внешний вид формы — на ваше усмотрение. Не забывайте, что это главная форма, с которой начинается работа, поэтому она должна выглядеть красиво и аккуратно.

Основную форму можно создать одним из двух способов:

- Создать обычную форму, не привязанную ни к какому источнику данных, и разместить на ней кнопки и надписи к ним.
- Воспользоваться диспетчером кнопочных форм (**Сервис/Надстройки/Диспетчер кнопочных форм**). Пользоваться диспетчером несложно, предлагается освоить эту работу самостоятельно.

Для того чтобы при запуске файла базы данных на экране сразу появлялась основная форма, выполним команду **Сервис/Параметры запуска**.



ЛИТЕРАТУРА

1. Ульман, Д. Введение в системы баз данных /Д.Ульман, Д.Уидом; Пер. с англ. – М.: Лори , 2000. – 512 с.
2. Базы данных: Учебник для ВУЗов / Под ред. А.Д.Хомоненко — СПб: Корона принт, 2000. – 416 с.
3. Роджер, Д. Использование Microsoft Access 2000. Специальное издание /Д.Роджерс; Пер. с англ. — М.: Издательский дом «Вильямс» , 2000. – 1152 с.
4. Ржеуцкая, С. Программирование баз данных. Рабочая программа, методические указания и контрольные задания / С.Ю.Ржеуцкая, Г.Г.Рапаков — Вологда: ВоГТУ, 2002. –19 с.