**министерство образования и науки Российской Федерации**

Старооскольский технологический институт им. А.А. УГАРОВА

(филиал) федерального государственного автономного образовательного учреждения

высшего образования

«Национальный исследовательский технологический университет «МИСиС»

**ОСКОЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ**

УТВЕРЖДено

НМС опк

пРОТОКОЛ №1

ОТ «01» сентября 2016 г.

**ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ**

***Методические указания для студентов очной формы обучения для выполнения практических заданий***

Специальность 09.02.04 Информационные системы (по отраслям)

Старый Оскол 2016г

|  |  |
| --- | --- |
| *Рассмотрены на заседании П(Ц)К 09.02.04*  *Протокол №*  *от «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016г.*  *Председатель*  *Назарова О.И.* | *Методические указания составлены в соответствии с рабочей программой по дисциплине*  *Основы проектирования баз данных*  *Специальности*  *09.02.04 Информационные системы (по отраслям)*  *Зам .директора по М Р*  *к.п.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_А.М. Степанова* |

***Составитель: Артюхина Д.Д., Коренькова Т.Н.***

***Рецензенты:***

*внутренний: Горюнова М.В.* - преподаватель ОПК СТИ НИТУ МИСиС

*внешний:* Анпилов А.Э. – инженер ООО «КМАЭМ»

СОДЕРЖАНИЕ

[Введение 5](#_Toc466125041)

[Практическая работа №1 6](#_Toc466125042)

[Практическая работа №2 15](#_Toc466125043)

[Практическая работа №3 21](#_Toc466125044)

[Практическая работа №4 29](#_Toc466125045)

[Практическая работа № 5 60](#_Toc466125046)

[Практическая работа №6 72](#_Toc466125047)

[Практическая работа №7 105](#_Toc466125048)

[Практическая работа №8 117](#_Toc466125050)

[Практическая работа №9 125](#_Toc466125052)

[Практическая работа №10 136](#_Toc466125054)

[Практическая работа №11 148](#_Toc466125055)

[Практическая работа №12 163](#_Toc466125056)

[Практическая работа №13 166](#_Toc466125057)

[Практическая работа №14 170](#_Toc466125058)

[Список использованных источников 179](#_Toc466125059)

# Введение

Базы данных являются составной частью информационной системы (ИС), осуществляющей хранение и обработку данных в широком смысле, включая ввод и корректировку данных, удаление данных, выполнение различных запросов, арифметические и логические преобразования данных, составление отчетов, выдачу информации в различных формах на экране и в бумажном виде.

Проектирование баз данных представляет собой достаточно сложный процесс, основывающийся на многих научных дисциплинах таких, как системный анализ, концептуальное моделирование, проектирование баз данных, теории нормализации, проектировании, реляционной алгебры и языков запросов, и многих других, включенных в общую теорию информатики.

В предлагаемых методических указаниях рассматриваются вопросы концептуального моделирования, теории проектирования баз данных на основе проектирования, операций реляционной алгебры и простейших запросов.

# **Практическая работа №1**

Определение структуры новой таблицы и создание таблиц.

Цель работы: научить учащихся создавать файл базы данных Access, определять структуру новой таблицы в режиме конструктора, создавать первичный ключ и сохранять созданные таблицы.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

Рассмотрим последовательность необходимых действий при создании файла и таблиц базы данных на примере проекта базы данных "Учебный процесс"

Для создания файла новой БД необходимо при открытии программы MS Access выбрать команду **Создать – Пустая база данных рабочего стола** (рис. 1).



Рисунок 1. Создание пустой БД рабочего стола

Далее необходимо указать имя БД, как показано на рисунке 2.

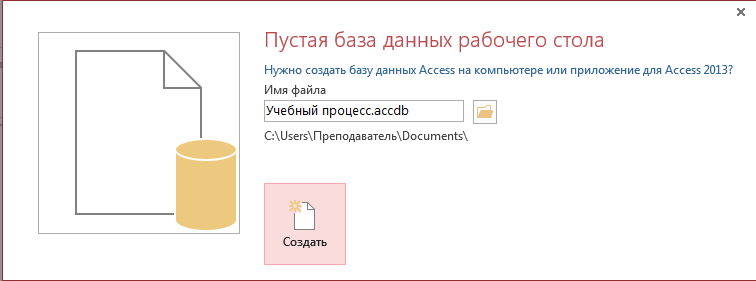


Рисунок 2. Создание пустой БД рабочего стола

Прежде чем нажать на кнопку **Создать**, необходимо определите папку, в которой будет размещен этот файл. Папка должна находиться по адресу **С:\\User\User\Документы\ФАМИЛИЯ, ГРУППА СТУДЕНТА**

**Практическая часть**

**Создание таблицы**

Основные параметры структуры таблицы, соответствующей информационному объекту группа, представлены ниже в таблице 1.

**Создание структуры таблицы**

Начнем создание таблицы группа с определения ее структуры в режиме конструктора таблиц. В окне базы данных будет открыта по умолчанию Таблица 1 (рисунок 3).

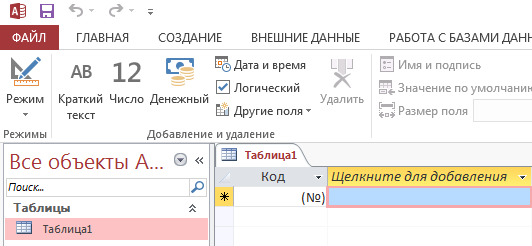


Рисунок 3 – Таблица 1

Для редактирования структуры данной таблицы, необходимо выделить ее и выполнить следующую команду Режим – Конструктор (рисунок 4)

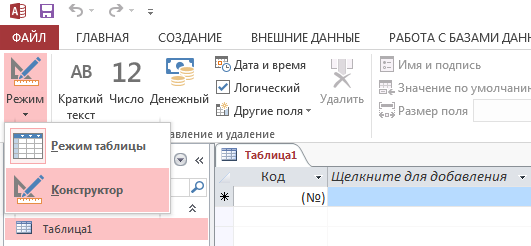


Рисунок 4. Выбор режима Конструктор для таблицы

Далее необходимо задать имя таблицы **ГРУППА.**

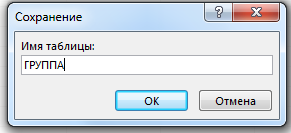


Рисунок 5. Сохранение таблицы

Для определения структуры таблицы группа в соответствии с приведенными в табл. 1 проектными параметрами, выполните в окне конструктора Таблица 1: таблица следующую последовательность действий:

1. В столбец Имя поля введите в нужной последовательности имена полей НГ, КОЛ, ПБАЛЛ.
2. В столбце Тип данных выберите нужный тип данных для каждого поля, использовав кнопку списка.

3. На вкладке Общие задайте свойства полей:

- Размер поля, нажав на нем кнопку списка:

* для текстового поля НГ задайте размер поля, равный 3;
* для числового поля КОЛ выберите значение Байт, определяющее длину целого числа, достаточную для размещения максимального значения поля (40 студентов в группе);
* для числового поля ПБАЛЛ выберите значение Одинарное с плавающей точкой;
* Формат поля (Format) для поля ПБАЛЛ — Фиксированный (Fixed);
* Число десятичных знаков для поля ПБАЛЛ — 2;
* Подпись поля, Условие на значение, Сообщение об ошибке — выберите для каждого из полей значения, как указано в табл. 1.

Таблица 1. Описание свойств полей таблицы ГРУППА

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Ключевое (уникальное) | Обязательное поле | Тип данных | Размер поля | Число дес. знаков | Подпись поля | Правило проверки | Сообщение об ошибке |
| НГ | Да | Да | Текстовый | 3 |  | Ном.  группы |  |  |
| КОЛ |  | Нет | Числовой | Байт |  | Кол-во студ. В группе | >=0 and <=35 | Кол. Студ.  больше  допустимого |
| ПБАЛЛ |  | Нет | числовой | С плав., точкой | 2 | Средний балл | >2 and <5 or 0 | Ошибка в оценке |

Условие на значение, которое заносится в бланк запроса, является выражением, которое может быть сформировано с помощью построителя выражений. Построитель вызывается при нажатии кнопки справа от строки Условие на значение, в которую должна быть введено выражение. (Не допускайте пробелов при вводе выражения. Нужные пробелы построитель введет сам.)

После ввода выражения в окно построителя и нажатия клавиши EnterAccess выполнит синтаксический анализ выражения и поместит его в строке Условие на значение.

Теперь определим первичный ключ таблицы. Выделим поле НГ, щелкнув кнопкой мыши на области маркировки слева от имени поля, и нажмем кнопку Ключевое поле. Признаком установки ключа является изображение ключа слева от имени поля. Определим свойства ключевого поля в соответствии с табл. 1

Сохраним созданную структуру и присвоим имя новой таблице - ГРУППА. Для этого выполним команду Сохранить и введем это имя в строке Сохранение. Таблица ГРУППА появится в списке объектов Таблицы в окне Учебный процесс: база данных. При сохранении таблицы происходит обновление файла базы данных учебный процесс.mdb.

УПРАЖНЕНИЕ. Создайте структуру таблиц КАФЕДРА, ПРЕПОДАВАТЕЛЬ, ПРЕДМЕТ, СТУДЕНТ, ИЗУЧЕНИЕ, УСПЕВАЕМОСТЬ. При создании таблиц используйте проектные параметры их структуры, которые представлены ниже.

При открытии базы данных в первый раз, вы видели пустую таблицу в режиме таблицы, где можно добавить данные. Для того, чтобы создать таблицу в существующей базе данных, необходимо нажать на кнопку **Конструктор таблиц** во вкладке **СОЗДАНИЕ** (рисунок 6)

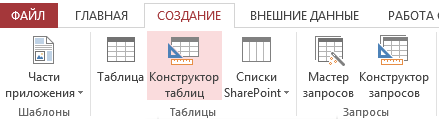


Рисунок 6 – создание таблицы в режиме Конструктор

Описание свойств полей таблицы СТУДЕНТ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Ключевое поле | Уникальное | Обязательное | Тип данных | Размер | Число дес. знаков | Подпись поля |
| НГ | Да | Да | Да | Текстовый | 3 |  | Группа |
| НС | Да | Да | Текстовый | 2 |  | Номер студента в группе |
| ФИО |  |  | Да | Текстовый | 15 |  | ФИО |
| ГОДР |  |  | Нет | Числовой | Целое |  | Год рождения |
| АДРЕС |  |  | Нет | Текстовый | 25 |  |  |
| ПБАЛЛ |  |  | Нет | Числовой | С плав. точкой | 2 | Балл при поступлении |

Описание свойств полей таблицы КАФЕДРА

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Ключевое | Уникальное | Обязательное | Тип данных | Размер | Подпись поля |
| ККАФ | Да | Да | Да | Текстовый | 2 | Код |
| НКАФ |  |  | Нет | Текстовый | 15 | Название |
| ТЕЛ |  |  | Нет | Текстовый | 9 | Телефон |
| ЗАВ |  |  | Нет | Текстовый | 15 | ФИО зав каф |
| ФОТО |  |  | Нет | Поле  объекта OLE |  | Фотография заведующего |

Описание свойств полей таблицы ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Ключевое | Уникальное | Обязательное | Тип данных | Размер | Подпись поля |
| ТАБН | Да | Да | Да | Текстовый | 4 | Табельный номер |
| ФИО |  |  | Да | Текстовый | 30 | ФИО преподавателя |
| СТ |  |  | Нет | Текстовый | 15 | Уч. степень |
| ЗВ |  |  | Нет | Текстовый | 10 | Уч. звание |
| ККАФ |  |  | Да | Текстовый | 2 | Код кафедры |

Описание свойств полей таблицы ПРЕДМЕТ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Ключевое | Уникальное | Обязательное | Тип  данных | Размер | Подпись поля | Условие на значение | Сообщение об ошибке |
| КП | Да | Да | Да | Текстовый | 2 | Код  предмета |  |  |
| НП |  |  | Да | Текстовый | 15 | Название предмета |  |  |
| ЧАСЫ |  |  | Нет | Числовой | Целое | Всего часов | >0 and <300 | Число часов не более 300 |
| ЛЕК |  |  | Нет | Числовой | Целое | Лекции |  |  |
| ПР |  |  | Нет | Числовой | Целое | Практика |  |  |
| ЧС |  |  | Нет | Числовой | Целое | Семестров |  |  |
| ПРОГР |  |  |  | Поле MEMO |  | Программа |  |  |

Описание свойств полей таблицы ИЗУЧЕНИЕ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Ключевое | Уникальное | Обязательное | Тип данных | Размер | Число дес знаков | Подпись  поля |
| НГ | Да | Да | Да | Текстовый | 3 |  | Номер группы |
| КП | Да | Да | Текстовый | 2 |  | Код  Предмета |
| ТАБН | Да | Да | Текстовый | 4 |  | Таб. ном препод |
| ВИДЗ | Да | Да | Текстовый | 3 |  | Вид занятий |
| ЧАСЫ |  |  | Нет | Числовой | Целое | 0 | Часов по занятию |
| СБАЛЛ-ГР |  |  | Нет | Числовой | С пл. точкой | 2 | Ср. балл по предмету |

Описание свойств полей таблицы УСПЕВАЕМОСТЬ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя поля | Ключевое | Уникальное | Обязательное | Тип данных | Размер | Число дес знаков | Подпись поля |
| НГ | Да | Да | Да | Текстовый | 3 |  | Номер группы |
| НС | Да | Да | Текстовый | 2 |  | Номер студента |
| КП | Да | Да | Текстовый | 2 |  | Код  предмета |
| ТАБН | Да | Да | Текстовый | 4 |  | Таб. ном препод. |
| ВИДЗ | Да | Да | Текстовый | 3 |  | Вид занятия |
| ОЦЕНКА |  |  | нет | Числовой | Целое | 0 | Оценка |

Рассмотрим некоторые особенности структуры таблиц кафедра, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ, СТУДЕНТ **базы данных** Учебный процесс.

Использование данных типа объект OLE

В таблице КАФЕДРА предусмотрено поле Фото, которое должно содержать фотографию заведующего, хранящуюся в формате графического редактора Paint в файле с расширением bmp. Тип данных такого поля должен быть определен как Поле объекта OLE. Размещение этого объекта в поле производится на этапе заполнения полей таблицы в режиме таблицы или через форму. Отображение объекта возможно только в форме или отчете. Объект может быть внедренным или связанным.

**Замечание**

OLE (ObjectLinkingandEmbedding, Связывание и внедрение объектов) — это метод передачи информации в виде объектов между приложениями Windows. Поле объекта OLE является средством, позволяющим установить связь с объ­ектами другого приложения или внедрить объект в базу данных. Объектом является часть документа, созданная в другом приложении, сохраняющая формат документа источника и информацию о создавшем его приложении. Объектами могут быть простые и форматированные тексты, рисунки, диаграммы, файлы звукозаписи (WAV), музыка в формате MIDI (Музыкально-инструментальный цифровой интерфейс), файлы анимации (FLI, МММ), видеоклипы (АVI), электронные таблицы и другие элементы из различных приложений, поддерживающих это средство. Access, поддерживая OLE, полностью интегрирован с другими приложениями пакета MicrosoftOffice.

Внедренный объект сохраняется в файле базы данных. Двойным щелчком мыши на ячейке, содержащей внедренный объект, открывается возможность редактирования объекта средствами приложения, в котором объект был создан.

Связанный объект сохраняется в отдельном файле. Файл объекта можно обновлять независимо от базы данных. Последние изменения будут выведены на экран при следующем открытии формы или отчета. При работе с базой данных также можно просматривать и редактировать объект. Отредактированный связанный объект будет сохраняться в файле объекта, а не в файле базы данных. Связывание объекта удобно при работе с большими объектами, которые нежелательно включать в файл БД, а также с объектами, используемыми в различных документах организации, например, логотип. Если связанный файл объекта перемещен, необходимо повторно установить связь.

Использование данных типа MEMO

В таблице предмет предусмотрено поле прогр, которое будет содержать текстовое данное большой длины — краткую программу курса. Для такого поля выбирается тип данного — **Поле MEMO** (Memo). Ввод данных в это поле можно выполнить непосредственно в таблице либо через область ввода, вызываемую нажатием клавиш <Shift>+<F2>.

Если тексты программ по предметам подготовлены в некотором текстовом редакторе, например, MicrosoftWord, и хранятся в отдельных файлах, удобно для этого поля задать тип **Поле объекта**OLE (OLEObject) и при вводе значений в него установить связь с файлами.

Определение составного первичного ключа

В таблице студент в составной первичный (уникальный) ключ входят поля НГ и НС. Для определения этого ключа в режиме конструктора таблиц следует выделить оба эти поля, щелкнув кнопкой мыши на области маркировки при нажатой клавише <Ctrl>, затем нажать на панели инструментов кнопку **Ключевое поле** (PrimaryKey).

Аналогично определяются составные ключи в таблицах ИЗУЧЕНИЕ и УСПЕВАЕМОСТЬ.

**Контрольные вопросы**

1. Как создать таблицу в Access?
2. В каком расширении сохраняется БД Access?
3. Как перейти в режим Конструктора?
4. Что такое первичный ключ?
5. Что такое объектOLE?
6. Где возможно отображение объекта OLE?
7. Что такое внедренный объект?
8. Что такое связанный объект?
9. Что такое поле MEMO?
10. Чем первичный ключ отличается от уникального?

# ПРАКТИЧЕСКАЯ РАБОТА №2

**Открытие, редактирование и пополнение табличного файла.**

Цель работы: Научить студентов вводить исходные данные в таблицу, перемещаться по таблице, добавлять, изменять и удалять записи.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

Теоретическая часть

**Ввод данных в таблицы БД**

**Ввод записей в режиме таблицы**

Введем несколько записей в таблицу **кафедра,** данные для которых представлены в таблице Данные таблицы КАФЕДРА

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Название | Телефон | ФИО зав.каф. |
| 01 | Информатики | 32-32-32 | Игнатьев В.В. |
| 02 | Математики | 32-23-23 | Иванов И.И. |
| 03 | Истории | 42-32-23 | Смирнова И.В. |
| 04 | Иностранного языка | 42-42-42 | Жданова А.Е. |
| 05 | Физкультуры | 42-42-32 | Плетнёв В.А. |
| 06 | Философии | 32-42-32 | Бондаренко В.В. |

В окне учебный процесс: база данных установим курсор на таблице КАФЕДРА и выполняем команду Главная – Режим – Режим таблицы (рисунок 6). Таблица откроется в режиме таблицы. Заполним строки (записи) открывшейся таблицы в соответствии с названиями столбцов (полей).

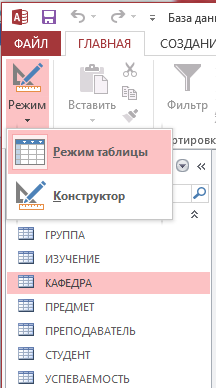


Рисунок 7 – Открытие таблицы в режиме таблицы

Корректность вводимых данных (соответствие заданному типу поля, размеру, и условию на значение, которые определены в свойствах полей в режиме конструктора) проверяется автоматически при их вводе. Отслеживается уникальность значений ключевых полей.

Отменить ввод значения в поле до перехода к другому полю можно, нажав кнопку Отменить (рисунок 7). Переход от одного поля к другому можно выполнить клавишей <Таb> или переводом курсора.

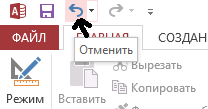


Рисунок 8 - Отмена последней записи

Завершение ввода новых значений записи или редактирования осуществляется при переходе к любой другой записи (при смене текущей записи).

Практическая часть

Добавление записи

Добавление записи в таблицу начинается с заполнения пустой строки, размещенной в конце таблицы и помеченной звездочкой (\*).

Переход к этой записи можно выполнить также по команде Главная – Перейти – Новая запись (рисунок 8) или нажав на кнопку Создать (рисунок 9). Таблица должна находиться в режиме таблицы. Корректируемая запись помечается слева символом карандаша.



Рисунок 9 – Создание новой записи

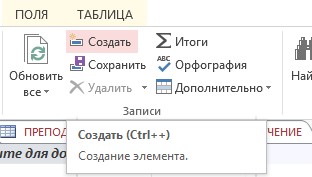
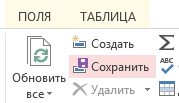


Рисунок 10 – Создание элемента

Замечание

Для создания новой записи должны быть обязательно заполнены ключевые поля, обязательные поля, а также поля связи, если установлены связи таблицы с другими таблицами.

Сохранение новой записи происходит после перехода к другой записи или выполнения нажатия на кнопку Сохранить на ленте  или на панели быстрого доступа .

Изменение и удаление записи

Изменение значений в полях записи осуществляется непосредственно в ячейках таблицы. Введенное значение проверяется Access при попытке перевода курсора в другое поле. Если значение не является допустимым, появ­ляется предупреждающее сообщение. Если данные были введены неверно, можно отменить внесенные изменения. Например, чтобы отменить изменения при редактировании данных в поле, следует нажать кнопку Отменить на панели быстрого доступа или нажать клавишу ESC. Чтобы отменить все изменения, внесенные в запись, нажмите клавишу ESC еще раз. Даже после сохранения изменений или после перехода на другую запись все еще можно отменить изменения, внесенные в предыдущую запись, нажатием кнопки Отменить на панели быстрого доступа. Однако после начала редактирования другой записи, применения или снятия фильтра либо при переключении на другое окно или вкладку документа изменения становятся постоянными.

При удалении записи данные безвозвратно удаляются из таблицы. Таким образом, следует удостовериться, что запись требуется удалить. Если необходимо удалить только некоторые данные, но не всю запись, выделите данные, которые необходимо удалить, и нажмите клавишу DELETE. Для удаления записи необходимо выделить запись или записи, которые необходимо удалить, и нажать на клавишу DEL или вкладке Главная в группе Записи щелкните Удалить 

Чтобы выделить запись, щелкните область выделения записей, расположенную рядом с записью. Чтобы расширить или уменьшить выделение перетащите область выделения записей (если она доступна) либо нажмите клавиши SHIFT+СТРЕЛКА ВНИЗ или SHIFT+СТРЕЛКА ВВЕРХ.

Размещение объекта OLE

Рассмотрим размещение объекта OLE на примере поля Фотография заведующего в таблице КАФЕДРА. Пусть фотографии хранятся в формате графического редактора Paint в файлах с расширением bmp.

Рассмотрим вариант внедрения объекта в файл базы данных. Установим курсор в соответствующем поле таблицы. Нажав на правую кнопку мыши, выберем Вставить объект из выпадающего списка. В открывшемся окне необходимо установить переключатель Создать из файла. Далее окно примет вид, который позволит ввести имя файла с фотографией. Для поиска файла можно воспользоваться кнопкой Обзор, при нажатии которой активизируется диалоговое окно, позволяющее просмотреть диски и папки и выбрать необходимый файл.

Флажок Связь (Link) по умолчанию не установлен и, следовательно, содержимое файла будет введено в поле как внедренный объект. Увидеть содержимое поля можно через форму или отчет. Дальнейшие изменения файла не будут отражаться на встроенном объекте. Для ввода в поле связанного объекта следует установить флажок Связь (Link). Это сэкономит место в базе данных и даст возможность отображать все вносимые в файл изменения.

В результате заполнения таблицы в соответствующем поле указан вид объекта - точечный рисунок. Для просмотра внедрённого объекта достаточно установить курсор в соответствующем поле и дважды щёлкнуть кнопкой мыши.

Для отображения содержимого поля в виде значка, представляющего файл с документом, в окне установите флажок В виде значка. Значок может быть использован для представления связанного объекта.

Ввод логически связанных записей

Введем несколько логически взаимосвязанных записей в таблицы ГРУППА и СТУДЕНТ.

Объекты ГРУППА и СТУДЕНТ связаны одно-многозначными отношениями, но, пока не создана схема данных, в которой устанавливаются связи между таблицами, система не может контролировать логическую взаимосвязь вводимых данных. Поэтому для получения целостной базы, в которой все записи подчиненной таблицы имеют логически связанную с ней главную запись, пользователю необходимо самому отслеживать логические связи записей. При вводе подчиненной записи в таблицу СТУДЕН Тнужно проверять наличие записи в главной таблице ГРУППА с ключом, значение которого совпадает, со значением поля связи (внешнего ключа) вводимой подчиненной записи. То есть при добавлении записи о студенте группы 221 необходимо, чтобы группа с этим номером уже была представлена в таблице ГРУППА.

При непосредственном вводе в таблицу записей, логически связанных с записями другой таблицы, полезно отобразить на экране обе таблицы.

Для одновременного отображения открытых таблиц можно воспользоваться командой Сверху вниз или Слева направо в меню Окно.

Данные таблицы СТУДЕНТ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа | Номер студента в группе | ФИО | Год рождения | Балл при поступлении |
| 101 | 01 | Аристов Р.П. | 1979 | 4,25 |
| 101 | 02 | Бондаренко С.А. | 1978 | 4,50 |
| 101 | 03 | Борисова Е.И. | 1979 | 4,75 |
| 101 | 04 | Макова Н.В. | 1977 | 4,75 |
| 102 | 01 | Боярская Н.П. | 1977 | 4,50 |
| 102 | 02 | Федоров Д.К. | 1977 | 4,25 |
| 102 | 03 | Сидоров И.Р. | 1977 | 4,50 |
| 103 | 01 | Андреев Г.М. | 1978 | 4,25 |
| 103 | 02 | Петров O.K. | 1979 | 4,75 |
| 104 | 01 | Иванов К.К. | 1977 | 4,50 |

Данные таблицы ГРУППА

|  |  |  |
| --- | --- | --- |
| Ном.группы | Кол-во студ. в группе | Средний балл |
| 101 | 30 | 4,50 |
| 102 | 32 | 4,50 |
| 103 | 29 | 4,80 |
| 104 | 35 | 4,40 |
| 105 | 35 | 4,80 |
| 201 | 35 | 3,90 |
| 202 | 30 | 4,00 |
| 203 | 28 | 4,70 |
| 204 | 25 | 4,00 |

Данные таблицы ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Табельный номер | ФИО препод. | Уч. степень | Уч. звание | Код кафедры |
| 101 | Андреев А.П. | Д.т.н. | Профессор | 01 |
| 102 | Апухин И.С. | К.т.н. | Доцент | 01 |
| 103 | Глухов И.Л. | К.т.н. | Доцент | 01 |
| 104 | Сеченов Ю.Б. | К.т.н. | Доцент | 01 |
| 105 | Чернов Л.К. | К.т.н, | Доцент | 01 |
| 201 | Блюмкина И.П. | Д. физ-мат. н. | Профессор | 02 |
| 202 | Львова П.Р. |  | Ассистент | 02 |
| 203 | Шапошников СИ. | Д.т.н. | Профессор | 02 |
| 204 | Новиков П.Н. |  | Ассистент | 02 |
| 301 | Ильясов И.Т. | К.ф.н. | Доцент | 03 |
| 302 | Пустынцев А.П. | К.и.п. | доцент | 03 |
| 303 | Романов Р.А. | К.и.н. | Доцент | 03 |
| 304 | Цветков А.И. | К.и.н. | Доцент | 03 |
| 401 | Сорокина М.Ф. | К.ф.н. | Доцент | 04 |
| 402 | Богомолов П.Р | К.ф.н. | Доцент | 04 |
| 403 | Лысова М.И | К.ф.н. | Доцент | 04 |
| 404 | Шаповалова М.Ф. | К.ф.н. | Доцент | 04 |
| 405 | Кудряшова Г.М. |  | Ассистент | 04 |
| 501 | Жигарева П.Р. | К.п.н. | Доцент | 05 |
| 502 | Егорова Т.И. |  | Ст.преп. | 05 |
| 503 | Ермолин Е.Н. |  | Ассистент | 05 |
| 601 | Логинов A.M. | К.ф.н. | Доцент | 06 |
| 602 | Яковлев П.П. | К.ф.н. | Доцент | 06 |
| 603 | Раков А.В. | К.ф.н. | Доцент | 06 |
| 604 | Соловьев СИ. |  | Ассистент | 06 |

Данные таблицы ПРЕДМЕТ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код предмета | Название предмета | Всего часов | Лекции | Практика | Семестров |
| 01 | Информатика | 200 | 80 | 120 | 4 |
| 02 | Математика | 200 | 100 | 100 | 4 |
| 03 | История | 140 | 90 | 50 | 3 |
| 04 | Ин.язык | 200 | 0 | 200 | 4 |
| 05 | Философия | 100 | 40 | 60 | 2 |
| 06 | Физкультура | 100 | 0 | 100 | 2 |

Данные таблицы ИЗУЧЕНИЕ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер группы | Код предмета | Таб.ном. препод. | Вид занятий | Часов по занятию |
| 101 | 01 | 101 | Лек | 40 |
| 101 | 01 | 102 | Пр | 60 |
| 101 | 02 | 201 | Лек | 50 |
| 101 | 02 | 202 | Пр | 50 |
| 101 | 03 | 301 | Лек | 48 |
| 101 | 03 | 302 | Пр | 20 |
| 101 | 04 | 401 | Пр | 50 |
| 101 | 05 | 501 | Лек | 50 |
| 101 | 05 | 502 | Пр | 50 |
| 101 | 06 | 601 | Лек- | 100 |
| 102 | 01 | 101 | Лек | 100 |
| 102 | 01 | 103 | Пр | 180 |
| 102 | 04 | 401 | Лек | 100 |
| 105 | 01 | 101 | Лек | 100 |
| 201 | 01 | 102 | Пр | 180 |
| 201 | 02 | 201 | Пр | 70 |
| 202 | 04 | 403 | Пр | 100 |
| 203 | 01 | 101 | Лек | 100 |
| 204 | 05 | 503 | Пр | 100 |

Данные таблицы УСПЕВАЕМОСТЬ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер группы | Номер студента | Код предм. | Таб.ном. преп. | Вид занятия | Оценка |
| 101 | 01 | 01 | 101 | Лек | 5 |
| 101 | 01 | 03 | 302 | Пр | 0 |
| 101 | 02 | 01 | 101 | Лек | 5 |
| 101 | 02 | 03 | 302 | Пр | 0 |
| 101 | 03 | 01 | 101 | Лек | 4 |
| 101 | 03 | 03 | 302 | Пр | 0 |
| 101 | 04 | 01 | 101 | Лек | 3 |
| 101 | 04 | 03 | 302 | Пр | 0 |

**Контрольные вопросы**

1. Каким образом можно открыть таблицу БД в Access?
2. Отменить ввод значения в поле до перехода к другому полю можно, нажав клавишу…?
3. Переход от одного поля к другому можно выполнить клавишей…?
4. После перехода к другой записи можно отменить ввод (редактирование) всей записи, выполнив команду…?
5. С чего начинается добавление записи в таблицу?
6. Отмена ввода значения в поле после перехода к другому полю текущей записи (до перехода к другой записи) происходит с помощью клавиши…?
7. Каким символом помечается корректируемая запись?
8. Как внести объект в файл базы данных?
9. Отмена изменения значения производится нажатием кнопки…?
10. Можно ли восстановить файл после подтверждения на удаление?

# Практическая работа №3

**Создание схемы данных. Сортировка, поиск и фильтрация данных.**

**Цель работы:** 1) научить студентов создавать схему данных на примере создания схемы данных базы данных Учебный процесс, включать таблицы в схему данных, устанавливать связи между таблицами.

2) научить студентов производить поиск и замену значений в полях, сортировать записи, создавать и удалять фильтры.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

Рассмотрим процесс создания схемы данных в соответствии с логической структурой базы данных Учебный процесс. При активном окне **Учебный процесс: база данных** нажмем на панели инструментов кнопку **Схема данных** **(Работа с базами данных – Схема данных** (рисунок 10)**)**. В открывшемся окне **Добавление таблицы** выберем вкладку **Таблицы** и, нажимая кнопку **Добавить**, разместим в окне **Схема данных** все ранее созданные таблицы базы данных, которые отображены в окне **Добавление таблицы**. Затем нажмем кнопку **Закрыть**. В результате в окне **Схема данных** будут представлены все таблицы базы данных Учебный процесс со списками своих полей и выделенными жирным шрифтом ключами.

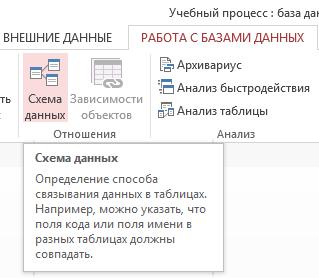


Рисунок 10 – Создание схемы данных

**Практическая часть**

**Ход работы**

**Определение связей по простому ключу**

Установим связь между таблицами ГРУППА и СТУДЕНТ, которые находятся в отношении типа один-ко-многим.

Для этого в окне **Схема данных** установим курсор мыши на ключевом поле НГ главной таблицы группа и перетащим его в область поля НГ в подчиненной таблице студент. В открывшемся окне **Изменение связей** в свойстве **Тип отношения** установится значение один-ко-многим. Установим флажок **Обеспечение целостности данных**. Если таблицы ГРУППА и СТУДЕНТ ранее были заполнены корректными данными, между таблицами будет установлена связь, обозначенная на схеме как 1:. Это свидетельствует о регистрации связи типа 1:М с параметром поддержания целостности. В противном случае появится сообщение о невозможности установить этот тип отношения

Для обеспечения автоматической корректировки данных во взаимосвязанных таблицах установим флажок **каскадное обновление связанных полей** и **каскадное удаление связанных записей**.

Аналогичные действия выполняются для других пар таблиц КАФЕДРА - ПРЕПОДАВАТЕЛЬ (**КЛЮЧ - ККАФ**) , ПРЕДМЕТ - ИЗУЧЕНИЕ (**КЛЮЧ - КП**), ПРЕПОДАВАТЕЛЬ - ИЗУЧЕНИЕ (**КЛЮЧ - ТАБН**), ГРУППА - ИЗУЧЕНИЕ (**КЛЮЧ НГ**) .

**Определение связей по составному ключу**

Определим связи между таблицами СТУДЕНТ - УСПЕВАЕМОСТЬ, которые связаны по составному ключу НГ+НС. Для этого в главной таблице студент выделим оба этих поля, удерживая нажатой клавишу <Ctrl>. Перетащим оба поля в область поля НГ в подчиненной таблице успеваемость.

В окне **Изменение связей** для ключевого поля не главной таблицы **Таблица/запрос** (рисунок 11) выберем соответствующее поле подчиненной таблицы **Связанная таблица/запрос**. В этом же окне установим флажок **Обеспечение целостности данных** и другие параметры связи.

Аналогично определяются связи между парой таблиц ИЗУЧЕНИЕ - УСПЕВАЕМОСТЬ (составной ключ связи: НГ + КП + ТАБН + ВИДЗ).

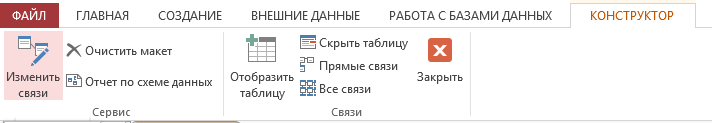


Рисунок 11 – Вызов окна «Изменение связи»

Напомним, что для каждой пары связываемых таблиц по умолчанию выбирается первый способ объединения, обеспечивающий объединение только тех записей, в которых связанные поля обеих таблиц совпадают.

После определения связей таблицы могут перемещаться в пределах рабочего пространства окна схемы данных. Перемещения и изменения размеров таблиц осуществляются принятыми в Windows способами. На рисунке показана схема данных БД Учебный процесс, где таблицы размещены в соответствии с их относительной подчиненностью.

Проверка работоспособности схемы данных, поддержание целостности осуществляется при конструировании форм, запросов, отчетов и их использовании, а также при непосредственной работе с таблицами. Ниже рассматри­вается процедура проверки целостности связей при непосредственной корректировке таблиц.

**Проверка поддержания целостности в базе данных**

На рисунке в созданной схеме данных БД Учебный процесс, все связи отмечены символами 1:∞. Это свидетельствует о том, что таблицы находятся в отношении типа 1:М (по простому или составному ключу) и для них установлен флажок обеспечения целостности данных.

Проверим, как обеспечивается поддержание целостности при внесении изменений в таблицы ГРУППА-СТУДЕНТ, связанные одно-многозначными отношениями.

**Проверка целостности**

**при изменении значений связанных полей в таблицах**

Откроем таблицу ГРУППА в режиме таблицы. Изменим значение ключевого поля нг (номер группы) в одной из записей главной таблицы ГРУППА. Убедимся, что во всех записях подчиненной таблицы СТУДЕНТ автоматически изменится значение поля нг для обучающихся в этой группе студентов. Изменение происходит, т. к. был установлен флажок **каскадное обновление связанных полей.** Причем это изменение осуществляется мгновенно, как только изменяемая запись перестает быть текущей. Для наблюдений за автоматическими изменениями в подчиненной таблице откроем одновременно таблицы ГРУППА и СТУДЕНТ. Если открыть другие подчиненные таблицы, то можно убедиться, что каскадное обновление распространяется также на подчиненные таблицы следующих уровней ИЗУЧЕНИЕ и УСПЕВАЕМОСТЬ.

Изменим значение ключа связи НГ в подчиненной таблице СТУДЕНТ па значение, которое отсутствует в записях таблицы ГРУППА, и убедимся, что такое изменение запрещено, так как при поддержании целостности не может существовать запись подчиненной таблицы с ключом связи, которого нет в главной таблице.

**Проверка при добавлении записей в подчиненную таблицу**

Убедимся, что невозможно включить новую запись в подчиненную таблицу СТУДЕНТ со значением ключа связи НГ, не представленным в таблице ГРУППА.

**Проверка при удалении записи в главной таблице**

Убедимся, что вместе с удалением записи в главной таблице ГРУППА удаляются все подчиненные записи, т. к. был установлен флажок **каскадное удаление связанных записей**.

Заметим, если каскадное удаление не разрешено, невозможно удалить запись главной таблицы при наличии связанных с ней записей в подчиненной.

Если выполнение изменений невозможно, появится сообщение невозможности внесения изменений.

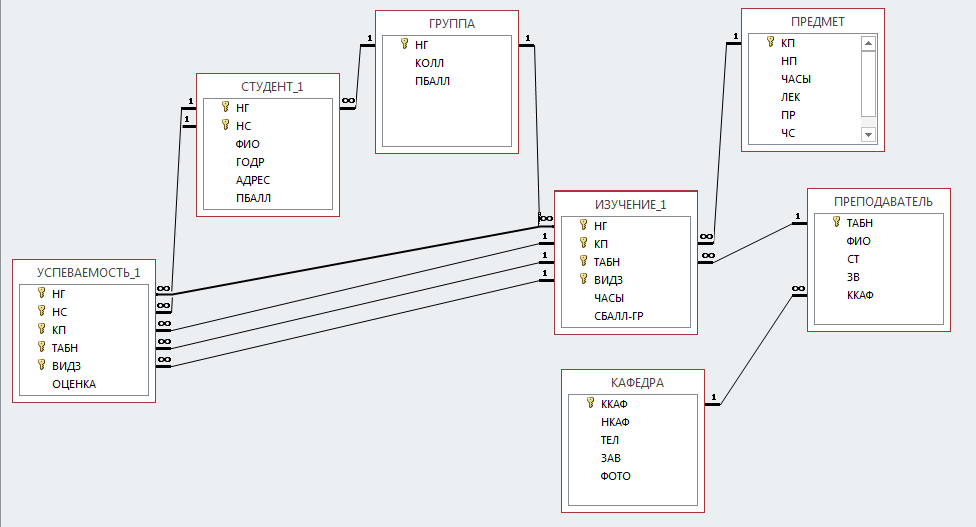


Рисунок 12 – Схема данных базы данных

**Поиск и замена значений в полях**

Для поиска значений в текущем или во всех полях таблицы необходимо в режиме таблицы или формы нажать кнопку панели инструментов **Найти** **.** В появившемся диалоговом окне **Поиск и замена** на вкладке **Поиск** в поле **Образец** вводится значение, которое требуется найти в полях таблицы. В раскрывающемся списке **Поиск в** выбирается область поиска. Если в таблице не выделен ни один из столбцов, этот раскрывающийся список позволяет выбрать в качестве области поиска поле, в котором установлен курсор, или таблицу целиком. Если в таблице выделено несколько столбцов, то этот список неактивен и поиск автоматически производится только в выделенных столбцах. В раскрывающемся списке **Совпадение** можно выбрать степень совпадения значений. **С начала ноля**, **С любой частью поля**, **Поля целиком**.

Раскрывающийся список **Просмотр** в нижней части окна позволяет выбрать направление поиска, которое может принимать значения Все, Вверх, **Вниз**. Если при сравнении значений требуется учиты­вать регистр и формат поля, необходимо, соответственно, установить флажки **С учетом регистра** и **С учетом формата полей**.

После нажатия в диалоговом окне кнопки **Найти далее** отыскивается первое значение, удовлетворяющее условию поиска. Для получения следующего значения, удовлетворяющего условию поиска, необходимо продолжить поиск, нажимая кнопку **Найти далее**. Образец поиска может быть задан полным значением поля или его частью. При этом могут использоваться операторы шаблона. Если выбрать в диалоговом окне вкладку **Замена**, оно дополнится раскрывающимся списком **Заменить на**, позволяющим ввести в найденные поля новое значение. Кроме того, появятся две дополнительных кнопки, позволяющие управлять процессом замены.

**Задание:** 1) Преподаватель Блюмкина И.П. вышла замуж и поменяла свою фамилию на фамилию на Сторожева. Внесите необходимые изменения в базу данных.

**Сортировка записей**

Сортировка записей по возрастанию или убыванию значений одного поля (поля сортировки) может быть выполнена как в режиме таблицы, так и в режиме формы. Для этого курсор устанавливается в поле сортировки и нажимается соответствующая кнопка на панели инструментов **Таблица в режиме таблицы** или **Режим формы**:

* кнопка **Сортировка по возрастанию;**
* кнопка **Сортировка по убыванию**.

Для сортировки записей можно также использовать команду меню **Главная – Сортировка и фильтр - Сортировка по возрастанию/Сортировка по убыванию.**

**Задание:** 1) Отсортируйте записи таблицы ПРЕПОДАВАТЕЛЬ по возрастанию табельного номера. 2) Отсортируйте записи таблицы ИЗУЧЕНИЕ по полю ЧАСЫ по убыванию.

**Фильтр по выделенному**

Существует три основных способа фильтрации.

* **Общие фильтры**    — такие фильтры всегда под рукой: вы можете выбрать нужное значение из фильтруемых данных или использовать готовые фильтры сравнения, например **Содержит**.
* **Фильтрация по выделению**    — такой фильтр подходит, если вы выбрали значение, по которому хотите отфильтровать данные.
* **Фильтр по форме** — позволяет лучше контролировать процесс, если вы хотите одновременно применить несколько фильтров.

Чтобы применить фильтр, щелкните правой кнопкой мыши нужное поле или элемент управления, а затем выберите команду фильтрации в нижней части появившегося меню. Чтобы выполнить фильтрацию по определенному значению, выделите его, щелкните правой кнопкой мыши и выберите команду фильтрации. Параметры фильтрации зависят от типа поля (рисунок 13)

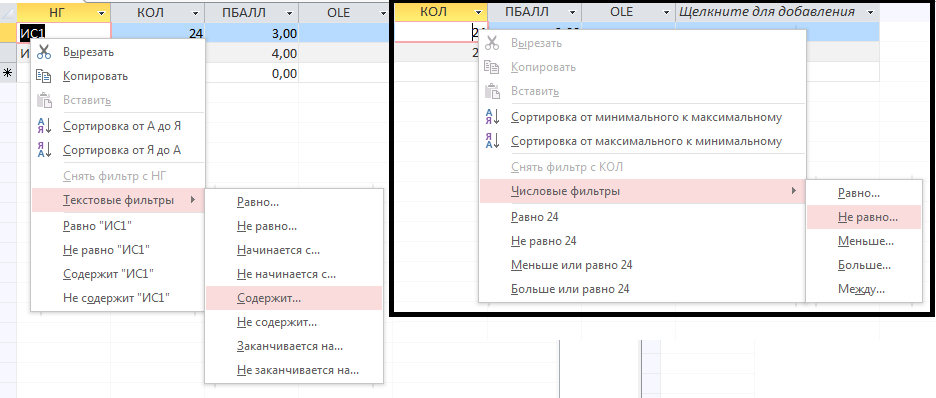


Рисунок 13 – Параметры фильтрации

Фильтры позволяют просмотреть только отдельные записи в форме, отчете, запросе или таблице либо напечатать нужные записи из отчета, таблицы или запроса. С помощью фильтра вы можете ограничить объем выводимых данных, не изменяя макет базовых объектов.

Рассмотрим этот способ фильтрации на примере таблицы студент. Пусть нужно найти все записи о студентах, фамилии которых начинаются с буквы с.

Откроем таблицу студент в режиме таблицы. Выделим букву **С** в поле ФИО одной из записей, где фамилия начинается с этой буквы. Выберем команду меню **Сортировка и фильтры - Фильтр - Фильтр по выделенному** или нажмем соответствующую кнопку на панели инструментов Таблица в режиме таблицы. В результате фильтрации в таблице будут отображены только те записи, в которых фамилия студента начинается на букву с.

На полученном подмножестве записей продолжим фильтрацию по дополнительному условию отбора. Среди найденных данных о студентах выберем родившихся в 1985 году. Выделим в поле Дата рождения (ДАТАР) одной из записей значение 85. После выполнения команды Фильтр по выделенному из всех записей подмножества в таблице останутся только те, в которых ноле ДАТАР имеет значение 1985

**Обычный фильтр**

*Обычный фильтр* вызывается командой **Сортировка и фильтры – Фильтр.** После выполнения данной команды в окне обычного фильтра на экран выводится пустая таблица или форма для активного объекта базы данных. На отрывшейся вкладке в поля фильтра вводятся значения, по которым будут отбираться записи. Значения могут вводиться с клавиатуры или выбираться из списка, как это показано для поля. Ввод значений в несколько полей одной строки фильтра определяет отбор записей, в которых присутствуют все указанные значения. При этом заданные условия рассматриваются как объединяемые логической операцией "И".

Для того чтобы указать альтернативный вариант отбора записей, включаемых в результирующий набор, выбирается вкладка Или (Or) в левом нижнем углу окна фильтра. При этом открывается новое поле фильтра для задания альтернативного варианта. Набор условий, введенных в разные поля, связывается логической операцией "ИЛИ". Следующий альтернативный вариант может быть задан на следующей вкладке Или (Or).

### Фильтрация по выделенному

Чтобы применить фильтр по выделенному значению, откройте вкладку **Главная** на ленте и в группе**Сортировка и фильтр** нажмите кнопку **Выделение**. Раскроется список с параметрами фильтрации. Как и для общих фильтров, доступные варианты зависят от типа данных. Для фильтрации по выделенному значению можно также щелкнуть его правой кнопкой мыши.

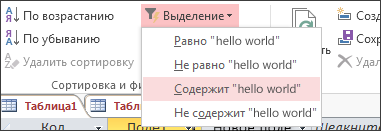


Рисунок 14 – Фильтрация по выделенному

### Фильтр по форме

Чтобы отфильтровать данные, заполнив форму, на вкладке **Главная** в группе **Сортировка и фильтр** нажмите кнопку **Дополнительно** и выберите пункт **Фильтр по форме**. При этом Access создаст пустую форму, аналогичную фильтруемой, и вы сможете заполнить столько полей, сколько захотите. Когда все будет готово, Access найдет записи, соответствующие условиям. Такая возможность удобна для фильтрации по нескольким полям. На рисунке показан простой фильтр по форме:

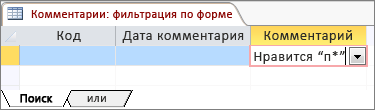


Рисунок 15 – Фильтрация по форме

При использовании фильтра по форме невозможно задавать значения полей следующих типов: многозначное поле, Memo, поле гиперссылки, логическое поле или поле объекта OLE.

**Расширенный фильтр**

*Расширенный фильтр* вызывается командой **Главная – Сортировка и фильтр – Дополнительно - Расширенный фильтр**. При этом в окне расширенного фильтра в верхней части выводится список полей активной таблицы. В нижней части окна выводится *бланк запроса*. В строку бланка запроса **Поле** из списка перетаскиваются мышью поля, по которым необходимо задать условия отбора записей. Условия отбора вводятся в соответствующее поле окна фильтра. Кроме того, бланк запроса позволяет выбрать тип сортировки для одного или нескольких выбранных полей в поле **Сортировка**.

**Сохранение фильтра**

Если к закрываемой таблице, запросу, форме или отчету применен фильтр, то он сохраняется вместе с ними. При следующем открытии объекта выводятся неотфильтрованные данные, но сохраненный фильтр легко применить — просто переключитесь в отфильтрованное представление, щелкнув **Без фильтра** в строке навигатора по записям. Если вы хотите, чтобы фильтр применялся при открытии объекта, установите для свойства **FilterOnLoad** значение **Да**.

Свойство **FilterOnLoad** применяется только при загрузке объекта. Если задать его для объекта в конструкторе и затем переключиться в другое представление, новое значение применено не будет. Чтобы изменения в свойстве **FilterOnLoad** вступили в силу, необходимо закрыть и снова открыть объект.

**Сохранение фильтра в качестве запроса**

Чтобы сохранить фильтр как запрос, а затем применять его к другим объектам, используйте расширенный фильтр: на ленте откройте вкладку **Главная**, в группе **Сортировка и фильтр** нажмите кнопку**Дополнительно** и выберите значение **Расширенный фильтр**. Настройте фильтр, как вы хотите, щелкните правой кнопкой мыши пустое место на вкладке **Фильтр** и выберите команду **Сохранить как запрос (рисунок 14)**.

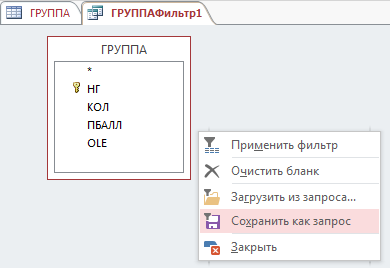


Рисунок 16 – Сохранение фильтра как запрос

## **Очистка фильтра**

Чтобы полностью удалить фильтр, очистите его. После этого фильтр невозможно снова применить с помощью значка **Без фильтра** — его потребуется создать заново. Можно снять один фильтр с отдельного поля или все фильтры из всех полей в представлении.

* **Очистка одного фильтра**    Щелкните отфильтрованный элемент правой кнопкой мыши и выберите команду **Снять фильтр с ИМЯ\_ПОЛЯ**.
* **Очистка всех фильтров**    На вкладке **Главная** в группе **Сортировка и фильтр** нажмите кнопку **Дополнительно** и выберите команду **Очистить все фильтры**.

## **Переключение между отфильтрованным и неотфильтрованным представлениями**

Чтобы просмотреть неотфильтрованные данные, щелкните в строке навигатора по записям надпись **С фильтром (рисунок 15)**.

Переключение фильтра в навигаторе по записям

Рисунок 17 – Строка навигатора

Чтобы вернуться к отфильтрованному представлению, щелкните надпись **Без фильтра**.

**Задание:** Создайте фильтр, который будет отбирать в таблице ПРЕПОДАВАТЕЛЬ всех преподавателей, имеющих ученое звание доцент и работающих на кафедре информатики (код кафедры 01).

**Контрольные вопросы**

1. Что такое схема данных?
2. Какие могут быть связи в схеме данных?
3. Что означает Обеспечение целостности данных?
4. Что используется для обеспечения автоматической корректировки данных во взаимосвязанных таблицах?
5. Сколько вы знаете способов объединения таблиц? Охарактеризуйте каждый способ?
6. Когда осуществляется проверка работоспособности схемы данных?
7. Для поиска значений в текущем или во всех полях таблицы необходимо…?
8. Каким образом можно выполнить сортировку данных в таблицах?
9. Что такое обычный фильтр?
10. Что такое расширенный фильтр?

# Практическая работа №4

Запросы к базе данных

**Цель:** Научить студентов создавать запрос к базе данных, просматривать его инструкции SQL, конструировать различные виды запросов.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

Теоретическая часть

1. ***Создание и назначение запроса***

Запрос – производный от таблиц объект БД, с помощью которого производится отбор информации; это механизм выборки, обновления, удаления и добавления данных, создания новых таблиц на основании данных одной таблицы или нескольких существующих таблиц.

Для чего нужен запрос?

* Отобрать поля. Необязательно включать в запрос все поля таблицы. Например, можно создать запрос, который показывает только фамилии и номера телефонов клиентов и не показывает адреса и прочие сведения.
* Отобрать записи. Пользователь может указать условия, которым должны удовлетворять записи, которые следует включить в динамический набор. Например, можно отобрать только записи со сведениями за определенный месяц.
* Сортировать записи. Пользователь может задать порядок изображения записей. Например, можно упорядочить записи о клиентах по алфавиту по фамилии клиента. Для того, чтобы отобразить отсортированные записи в форме, можно создать запрос, сортирующий эти записи, а затем использовать этот запрос в качестве источника данных для новой формы.
* Задать вопрос о данных, хранящихся в разных таблицах. Пользователь может использовать запрос для получения ответа на вопрос о содержимом нескольких таблиц, а затем представить результаты в одной таблице или форме.
* Выполнять вычисления. С помощью запросов можно создавать новые поля, содержащие результаты вычислений. Такие поля называются вычисляемыми. Например, для вычисления стоимости заказа можно создать поле, которое будет хранить произведение цены одной единицы товара на количество заказанных единиц товара.
* Использовать запрос в качестве источника данных для форм, отчетов и других запросов.
* Изменять табличные данные.

По способу создания различают 2 типа запросов: запросы по образцу (QBE - queries by example) и структурированный язык запросов (SQL - Structured Query Language).

Запрос по образцу (QBE) – это средство отыскания необходимой информации в БД обычно путем занесения данных в таблицу. Система запроса по образцу выводит данные, имеющие отношение к вводимым. Запрос формируется не на специальном языке, а путем заполнения бланка запроса в окне Конструктора запросов. Указываются критерии выборки записей в исходной таблице. С точки зрения пользователя, конструирование запроса – это просто пометка нужных элементов в окне Конструктора запросов.

SQL-запросы. С помощью структурированного языка формируются запросы любой сложности. Запрос составляется из последовательности SQL-инструкций, которые задают, что надо сделать с входным набором данных для генерации выходного набора. Работа с SQL-запросом рассчитана на специалиста-программиста. Пользователю, как правило, не приходится обращаться к языку SQL напрямую.

Различают запросы выборки и запросы-изменения. Существует 4 типа запросов-изменений:

· Запрос на создание таблицы. Создает новую таблицу, полностью или частично копируя структуру и данные из одной или нескольких таблиц.

· Запрос на удаление записей. Удаляет записи из одной или нескольких таблиц.

· Запрос на добавление записей. Добавляет группу записей из одной или нескольких таблиц в одну или несколько других таблиц.

· Запрос на обновление записей. Изменяет данные в группе записей.

· Запросы-выборки

Создание запроса - с помощью мастера и конструктором запросов

Последовательность создания запроса с помощью Конструктора:

1) В окне БД выбрать вкладку «Запросы» и щелкнуть кнопку «Создать»; при этом откроется окно диалога «Новый запрос».

2) Выбрать опцию «Конструктор» и нажать кнопку «ОК»; при этом открывается окно диалога «Добавление таблицы». В нем нужно выбрать имена таблиц-источников, на основе которых будет производиться выбор данных, и щелкнуть кнопку «Добавить.

3) Закрыть окно диалога «Добавление таблицы» кнопкой «Закрыть», окно Конструктора «Запрос на выборку» станет активным.

С помощью инструментария окна Конструктора формируем запрос:

1) выбираем таблицу-источник, из которой производится выборка записей;

2) выбираем поля результатов запроса (перетаскиванием в нижнюю часть окна Конструктора);

3) указываем критерии для выборки, группирования и упорядочения данных.

В верхней части окна Конструктора запроса размещается схема данных запроса, которая содержит список таблиц-источников и отражает связь между ними.

В нижней части окна находится Бланк запроса, в котором каждая строка выполняет конкретную функцию:

«Поле» - указываются имена полей, которые участвуют в запросе;

«Имя таблицы» - имя таблицы, из которой выбрано это поле;

«Сортировка» - указывается тип сортировки;

«Вывод на экран» - устанавливается флажок просмотра поля на экране;

«Условие отбора» - задаются критерии поиска.

Практическая часть

**Схема данных запроса**

В окне конструктора запроса отображаются выбранные таблицы со списком полей и одно-многозначные связи между ними, имеющиеся и схеме БД. Первая строка в списке полей, отмеченная звездочкой (\*), обозначает все множество полей таблицы. Кроме того, если таблицы имеют поля с одинаковыми именами и типами данных, Access автоматически устанавливает связи для объединения таких таблиц, даже когда связи не были установлены в схеме данных. Пользователь может самостоятельно установить неустановленные автоматически связи, переместив с помощью мыши задействованные в связи поля из одного списка полей в другой.

При использовании в запросе других запросом или таблиц, не представленных в схеме данных базы, с ними также могут быть установлены связи - объединения.

**Бланк запроса по образцу**

Бланк запроса по образцу представлен в виде таблицы на нижней панели окна запроса. Такая таблица предназначена для конструирования структуры таблицы результата запроса и условий выборки данных из исходных таблиц. Первоначально эта таблица пуста.

Каждый столбец бланка относится к одному запрашиваемому полю. Поля могут использоваться для включения в таблицу запроса, для задания сортировки, для задания условий отбора записей, а также для выполнения вычислений в записях таблицы.

При заполнении бланка запроса:

* в строку **Поле** включаются имена используемых в запросе полей;
* в раскрывающемся списке **Сортировка** выбирается порядок сортировки записей результата;
* в строке **Вывод на экран** устанавливаются флажки для полей, которые должны быть включены в результирующую таблицу;
* в строке **Условие отбора** задаются условия отбора записей;
* в строке **или** задаются альтернативные условия отбора записей.

В ряде случаев в бланке запроса наряду с именем поля необходимо отображать имя соответствующей таблицы, например, когда поля имеют одинаковые имена в разных таблицах. Для отображения имен таблиц в строке бланка следует выбрать команду **Вид | Имена таблиц** или нажать соответствующую кнопку на панели конструктора запросов. В результате выполнения команды в бланке появится строка **Имя таблицы**.

**Поля бланка запроса**

Каждый столбец бланка запроса соответствует одному из полей таблиц, на которых строится запрос. Кроме того, здесь может размещаться вычисляемое поле, значение которого вычисляется на основе значений других полей записи результата, или итоговое поле для групп записей, использующее одну из встроенных групповых функций Access. Для включения требуемых полей таблиц в соответствующие столбцы запроса можно воспользоваться следующими приемами:

* в первой строке бланка запроса **Поле** щелчком мыши вызвать появление кнопки списка и выбрать из списка нужное поле. Список содержит все поля таблиц, представленных в бланке запроса;
* переместить с помощью мыши требуемое поле из списка полей таблицы в схеме данных запроса в первую строку бланка запроса;
* дважды щелкнуть на имени поля таблицы в схеме данных запроса;
* для включения в запрос всех полей таблицы можно переместить с помощью мыши в соответствующую строку бланка запроса все поля из списка полей таблицы в схеме данных запроса или дважды щелкнуть на символе \* (звездочка) в этом списке.

**Конструирование однотабличного запроса на выборку**

Рассмотрим процесс конструирования однотабличного запроса на выборку на примере получения информации из таблицы ПРЕДМЕТ базы данных «Учебный процесс».

**Использование логических операций в условии отбора**

Пусть надо выбрать предметы, по которым общее число изучения не более 100, и есть лекции, а также выбрать предметы, по которым общее число часов больше 150 и число семестров изучения не более двух.

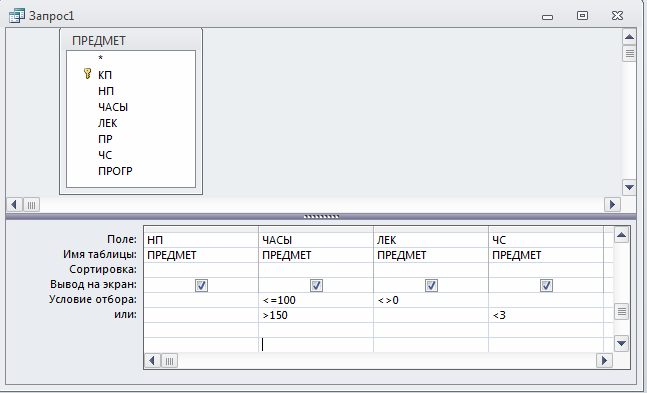
Результат должен содержать наименование предмета (НП), общее число часов по предмету (ЧАСЫ), количество лекционных часов (ЛЕК) и число семестров (ЧС).

Для создания запроса в режиме конструктора выберем вкладку **Создание** на панели быстрого доступа и нажмем кнопку **Конструктор запросов**.

После нажатия кнопки появляется окно запроса на выборку в режиме конструктора **Запрос1** (рисунок 18) и диалоговое окно **Добавление таблицы**. В диалоговом окне выберем таблицу ПРЕДМЕТ и нажмем кнопку **Добавить**.

Выбранная таблица будет отображена в области схемы данных запроса. Закроем окно **Добавление таблицы**.

В окне конструктора (рисунок 18) перетащим из списка полей таблицы ПРЕД МЕТ поля НП, ЧАСЫ, ЛЕК и ЧС в столбцы бланка запроса в строку **Поле**.



**Рисунок 18 -** Окно конструктора запроса на выборку с логическими операциями в условии отбора.

Сформулированные в задаче условия требуют формирования следующего логического выражения:

(ЧАСЫ<=100 AND ЛЕК<>0) OR (ЧАСЫ>150 AND ЧС<3)

Здесь ЛЕК<>0 (число лекций не равно нулю), соответствует заданному в зада- че условию выбрать предметы, в которых есть лекции.

Условия из первых скобок запишем в соответствующих полях ЧАСЫ и ЛЕК первой строки **Условия отбора**. Между условиями в разных полях одной строки выполняется логическая операция. Условия из вторых скобок запишем в соответствующих полях ЧАСЫ и ЧС второй строки **Условие отбора**. Между условиями, записанными в разных строках, выполняется логическая операция.

Выполним запрос, нажав на панели конструктора запросов кнопку **Выполнить**. На экране появится окно запроса в режиме таблицы с записями из таблицы ПРЕДМЕТ, отвечающими заданным условиям отбора.

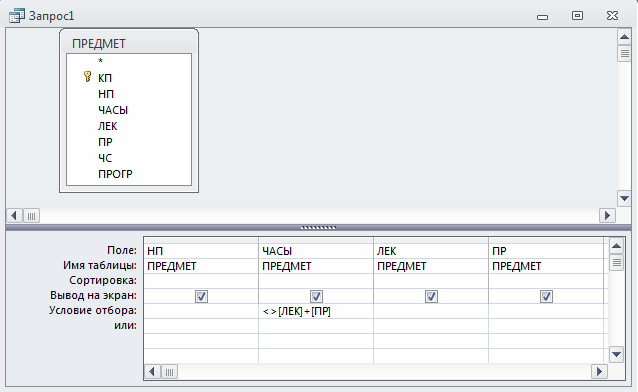
Сохраним запрос, нажав кнопку **Сохранить** на вкладке **Файл** и задав нужное имя запроса. Закроем текущий запрос нажав кнопку окна запроса **Закрыть**. Сохраненный запрос можно выполнить, выделив запрос в окне **Все объекты Access**, нажатием правой кнопки мыши и выбрав кнопку **Открыть**.

**Использование в условии отбора выражений с именами полей**

В предыдущем примере в условии отбора в качестве операндов использовались только значения для отбора по конкретным полям. Создадим запрос, в условии отбора которого сравниваются значения в разных полях.

Пусть необходимо проверить правильность задания общих часов в таблице ПРЕДМЕТ. По запросу должны отбираться только те записи, в которых значение в поле ЧАСЫ не равно значению, получаемому при сложении значений полей ПР и ЛЕК.

Такое условие записывается в бланке запроса в столбце ЧАСЫ и в нем используются имена полей [ПР] и [ЛЕК], как показано в бланке запроса на рисунке 19.



**Рисунок 19 -** Окно запроса на выборку из таблицы ПРЕДМЕТ записей, в которых количество часов не равно сумме часов лекций и практики

## Конструирование многотабличного запроса на выборку

Рассмотрим технологию конструирования многотабличного запроса на выборку на примере получения информации об успеваемости студентов из БД «Учебный процесс».

**Запрос на основе нескольких взаимосвязанных таблиц**

Пусть необходимо получить информацию об оценках полученных студентами по всем предметам. Результат должен содержать фамилию студента, наименования сданных предметов и оценки.

Для создания запроса на панели быстрого доступа выберем вкладку **Создание**

и нажмем кнопку **Конструктор запросов**.

**Формирование схемы данных запроса**

В окне Добавление таблицы выберем таблицы:

* СТУДЕНТ – для выборки фамилии студента из поля ФИО
* УСПЕВАЕМОСТЬ - для определения кодов предметов (поле КП), по которым студент сдал экзамены, выборки оценок по предмету (из поля ОЦЕНКА).
* ПРЕДМЕТ- для выборки наименования предмета (из поля НП), представленного кодом КП в таблице УСПЕВАЕМОСТЬ.

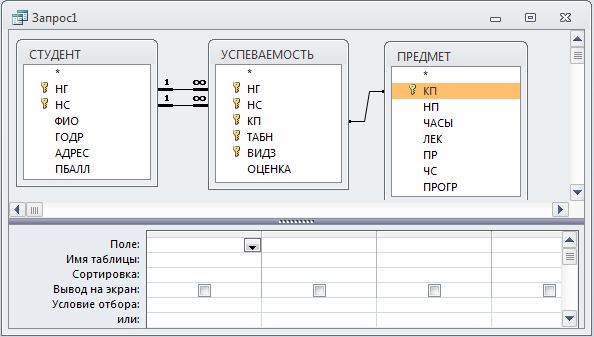
Закроем окно **Добавление таблицы**.

В окне конструктора запросов (рисунок 20) представлена схема данных запроса, содержащая выбранные таблицы. Между таблицами автоматически установлены необходимые связи:

* *Одно-многозначная* связь между таблицами СТУДЕНТ и УСПЕВАЕМОСТЬ по составному ключу НГ+НС в соответствии с построенной ранее схемой данных.

●*ь Связь объединение* между УСПЕВАЕМОСТЬ и ПРЕДМЕТ поскольку эти

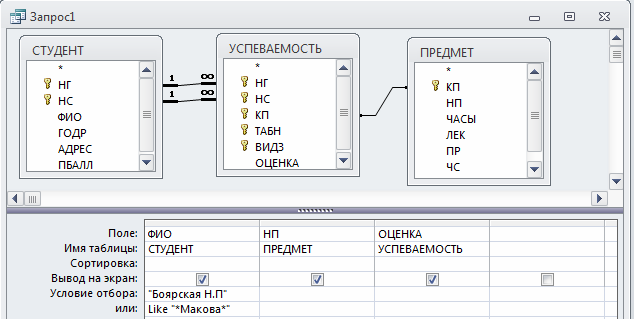
таблицы имеют поля с одинаковым именем КП и одинаковым типом данных.



**Рисунок 20 -**  Окно запроса об успеваемости студента с созданной схемой данных.

**Подготовка бланка запроса**

Поскольку в запросе используется несколько таблиц, в бланке запроса удобно видеть имя таблицы наряду с именем поля. Для отображения имен таблиц в бланке запроса (рисунок 21) нажмем кнопку **Имена таблиц** на панели инструментов конструктора или нажмем соответствующую кнопку в контекстном меню, который вызовем правой кнопкой мыши.



**Рисунок 21 -** Запрос на получение информации о сдаче экзаменов студентками Боярской Н.П. и Маковой.

Перетащим с помощью мыши поля, включаемые в результат выполнения запроса, в строку бланка запроса **Поле**:

* ФИО- из таблицы СТУДЕНТ
* НП- из таблицы ПРЕДМЕТ
* ОЦЕНКА- из таблицы УСПЕВАЕМОСТЬ

**Ввод значений в условия отбора записей**

Пусть необходимо получить информацию об успеваемости конкретных студентов: Боярской Н.П. и Маковой.

Зададим в строке **Условие отбора** их фамилии. Запишем фамилии студентов в разных строках бланка запроса, поскольку необходимо выбрать записи со значением в поле ФИО- Боярская или Макова. Поскольку инициалы студентки Маковой неизвестны, ее фамилию зададим с использованием символа шаблона «\*». Заметим, что фамилия с инициалами содержит точки, поэтому ее надо брать в кавычки. После ввода фамилии с символом шаблона система сама вставляет оператор Like, определяющий поиск по образцу. Заполненный бланк запроса представлен на рисунке 21.

Выполним запрос, нажав на панели конструктора запросов кнопку **Выполнить**.

**Замечание.** Записи о заданном студенте появятся в результирующей таблице запроса только в том случае, если запись об этом студенте содержится в таблице СТУДЕНТ, а в таблице УСПЕВАЕМОСТЬ имеются записи, связанные с записью о студенте.

**Формирование записей результата при выполнении запроса**

По заданной фамилии студента- Боярская Н.П.- в таблице СТУДЕНТ отыскивается запись. По значению ключа связи НГ+НС осуществляется выборка подчиненных записей из таблицы УСПЕВАЕМОСТЬ с оценками данного студента по разным предметам (в поле ОЦЕНКА). Для каждой из этих записей по значению ключа связи КП выбирается одна запись с наименованием предмета (НП) из таблицы ПРЕДМЕТ. Таким образом, таблица с результатом запроса будет содержать по одной записи о каждом предмете, сданном студентом. Аналогично формируются записи для второго заданного в запросе студента- Маковой.

**Ввод параметров в запрос**

В предыдущем примере для задания фамилии конкретного студента необходимо было корректировать бланк запроса. Чтобы избежать этого, целесообразно использовать в запросе параметры. При этом Access перед выполнением запроса через диалоговое окно будет запрашивать у пользователя конкретные значения параметров и введет их в условия отбора.

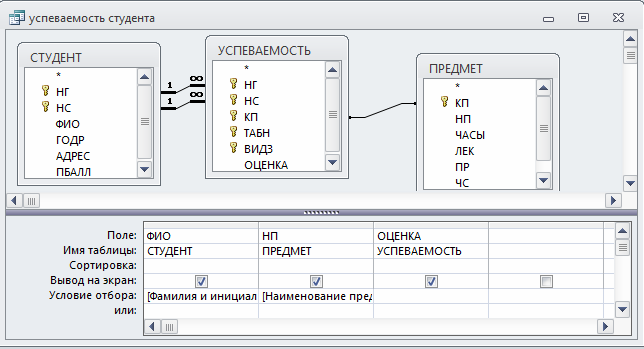
Пусть необходимо получить информацию об оценке студента по заданному предмету.

В условие отбора поля ФИО вместо конкретной фамилии введем название параметра, по которому будет запрашиваться фамилия при выполнении запроса. Название параметра введем как текст, заключенный в квадратные скобки:

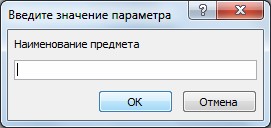
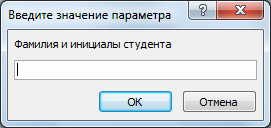
[ [Фамилия и инициалы студента]

Этот текст Access воспринимает как имя параметра (рисунок 22). В условие отбора поля НП введем второй параметр запроса:

[Наименование предмета]



**Рисунок 22 -** Бланк запроса с параметрами для ввода ФИО и НП.



**Рисунок 23 -** Диалоговые окна для ввода параметров запроса.

При выполнении запроса Access выведет диалоговые окна, представленные на рисунке 23, в которые пользователь сможет ввести нужные значения параметров.

**Использование имен полей различных таблиц в условии отбора**

Пусть необходимо выбрать записи из таблицы ИЗУЧЕНИЕ, в которых часы практических занятий по информатике не соответствуют равномерному распределению по семестрам всех часов практики.

Для решения этой задачи необходимо использовать таблицы:

* ИЗУЧЕНИЕ, в которой содержатся сведения о плановых занятиях в группах (в текущем семестре), в том числе о продолжительности (поле ЧАСЫ) каждого вида занятия (поле ВИД3).
* ПРЕДМЕТ, в которой содержатся сведения о наименовании (поле НП), общей продолжительности изучения предмета (поле ЧАСЫ), числа часов практики (ПР) и числе семестров изучения (ЧС).

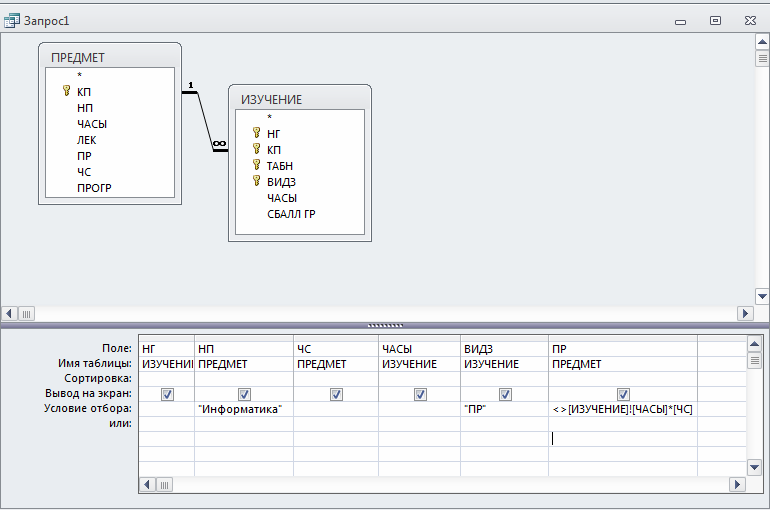
Для отбора записей о практических занятиях по информатике из таблицы ИЗУЧЕНИЕ надо в строке **Условие отбора** для поля НП (ТАБЛИЦА ПРЕДМЕТ) задать значение «Информатика», а для поля ВИД3 (таблицы ИЗУЧЕНИЕ) задать значение «пр» (практическое занятие).

При равномерном распределении практики по семестрам общее число часов практических занятий по предмету (ПР) должно равняться произведению ча- сов практики (ЧАСЫ) из таблицы ИЗУЧЕНИЕ на число семестров (ЧС) из таблицы ПРЕДМЕТ. Для решения рассматриваемой задачи надо включить в результат только те записи, для которых число часов не соответствует этому произведению. Для этого запишем в **Условие отбора** поля ПР (таблицы ПРЕДМЕТ) выражение:

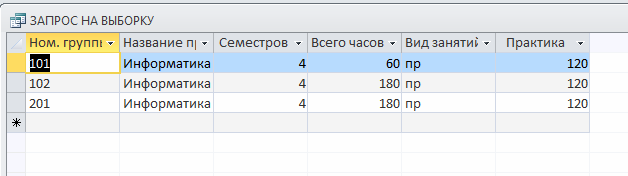
[ИЗУЧЕНИЕ] ! [ЧАСЫ]\*[ЧС]

[**Замечание.** Указывать таблицу ИЗУЧЕНИЕ для поля ЧАСЫ обязательно, потому что поле с таким же именем имеется и в таблице ПРЕДМЕТ. Запрос на выборку с условиями отбора записей, удовлетворяющих сформулированной задаче, приведен на рисунке 24. Результаты выполнения запроса приведены на рисунке 25.

**Замечание.** Если результат выполнения запроса не содержит записей, то это означает, что для заданного предмета в каждой из студенческих групп ча- сы практических занятий соответствуют равномерному распределению по се- местрам всех часов практики.



**Рисунок 24 -** Запрос с использованием в условиях отбора имен полей из разных таблиц.



**Рисунок 25 -** Записи о практических занятиях по информатике, в которых число часов не отвечает заданным условиям.

**Создание вычисляемых полей в запросах**

Вычисляемое поле, включенное в запрос, позволяет получить новое поле с результатами вычисления, отображаемыми только в таблице запроса, и не создает полей в исходных таблицах базы данных.

Рассмотрим технологию создания запроса с вычисляемым полем на примере таблицы ПРЕДМЕТ.

Пусть необходимо найти записи о предметах, в которых общее число часов по предмету не совпадает с суммой часов лекций и практики. Для решения этой задачи рассчитаем разность между общим числом часов по предмету (поле ЧАСЫ) и суммой часов лекций (поле ЛЕК) и практики (поле ПР). в ответ включим только те записи, для которых эта разность не равна нулю.

Создадим запрос на выборку для таблицы ПРЕДМЕТ. Перетащим в бланк запроса поля НП, ПР, ЛЕК, ЧАСЫ (рисунок 30).

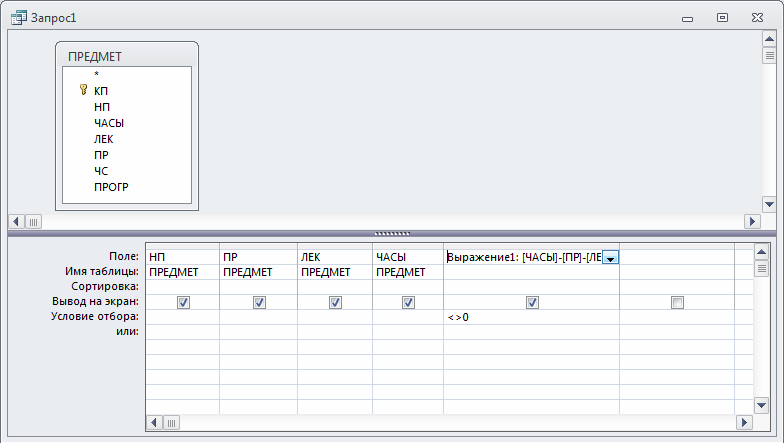
**Создание вычисляемого поля**

Для получения разности создадим вычисляемое поле в пустой ячейке строки

**Поле**, записав туда выражение:

[ЧАСЫ] - [ПР] - [ЛЕК]

Для отбора записей с ненулевым значением разности в вычисляемом поле в строку **Условие отбора** введем <>0 (не равно 0).



**Рисунок 30 -**  Запрос с вычисляемым полем и условием отбора по его значению.

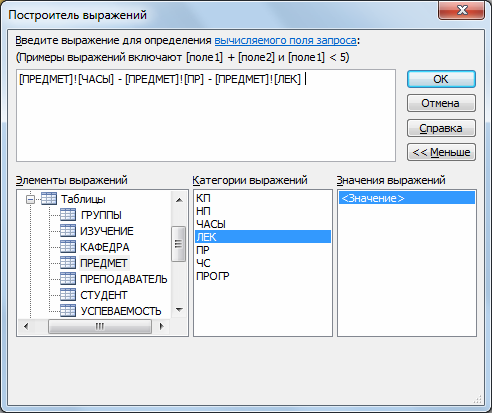
После ввода выражения система формирует имя вычисляемого поля по умолчанию - «Выражение 1». Это имя вставится перед выражением. Для изменения имени установим курсор мыши в вычисляемом поле бланка запроса и нажмем правую кнопку мыши. В контекстно- зависимом меню выберем **Свойства** поля, а в строку **Подпись** введем новое имя поля- «ЧАСЫ не равны ПР+ЛЕК». Имя поля может быть исправлено также непосредственно в бланке запроса.

**Использование построителя выражений**

Для формирования сложного выражения в вычисляемом поле целесообразно использовать построитель выражений. Построитель позволяет выбрать необходимые имена полей из таблиц, запросов, форм, знаки операций, функции.

Вызовем построитель выражений, нажав команду **Построить** в контекстно- зависимом меню (курсор мыши должен быть установлен на строке **Поле** вычисляемого поля).

В левой части окна **Построитель выражений** (рисунок 31) выберем таблицу ПРЕДМЕТ, на которой построен запрос. Справа отобразится список ее полей. Последовательно выберем нужные поля, добавляя их двойным щелчком мыши, знаки операций вводятся с клавиатуры. При этом в верхней части окна сформируется выражение.



**Рисунок 31 -**  Окно построителя выражений при формировании вычисляемого поля.

Сохраним запрос под именем «Разность часов по предмету». Сохранить. Сохраненный запрос можно выполнить, выделив запрос в окне **Все объекты Access**, нажатием правой кнопки мыши и выбрав кнопку **Открыть**.

Построенный запрос может быть использован для проверки правильности заполнения поля ЧАСЫ в таблице ПРЕДМЕТ.

**Использование групповых операций в запросах** **Назначение групповых операций**

Групповые операции позволяют выделить группы записей с одинаковыми значениями в указанных полях и использовать для этих групп одну из статистических функций. В Access предусмотрено девять статистических функций:

●- Sum сумма значений некоторого поля для группы

* + - Avg - среднее от всех значений поля в группе
    - Max, Min - максимальное, минимальное значение поля в группе
    - Count - число значений поля в группе без учета пустых значений
    - Stdev - среднеквадратичное отклонение от среднего значения поля в группе
    - Var - дисперсия значений поля в группе
    - First и Last - значение поля из первой или последней записи в группе Результат запроса с использованием групповых операций содержит по одной записи для каждой группы. В запрос включаются поля, по которым производится группировка, и поля, для которых выполняются групповые функции.

**Порядок создания запроса с использованием групповых операций** Для создания запроса с использованием групповых операций формируется запрос на выборку. В бланк запроса включаются поля, по которым надо произвести группировку, и поля, по которым надо произвести статистические вычисления.

Выполните команду **Создать/Конструктор запросов** и на панели инструментов конструктора запросов нажмите кнопку **Итоги**.

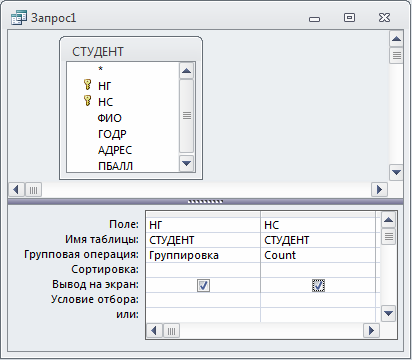
Для групповых вычислений по некоторому полю нужно заменить в нем слово **Группировка** на нужную статистическую функцию. Выбрать нужную функцию можно через раскрывающийся в поле список.

**Конструирование однотабличного запроса с групповой операцией** Рассмотрим технологию конструирования однотабличного запроса с групповой операцией на примере таблицы СТУДЕНТ.

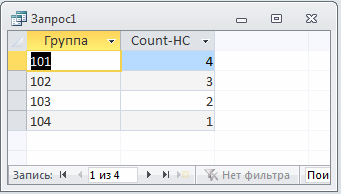
**Запрос с функцией Count**

Определим фактическое число студентов в группе. Создадим запрос на выборку из таблицы СТУДЕНТ. Из списка таблицы СТУДЕНТЫ перетащим в бланк запроса поле НГ (номер группы). Таким образом мы укажем, что по этому полю должна производиться группировка. Перетащим в бланк запроса поле НС, по которому будет вычисляться функция Count для подсчета числа студентов в группе.

Нажмем кнопку **Итоги**. Заменим слово "Группировка" в столбце НС на функцию Count. Для этого вызовем список и выберем эту функцию. Бланк запроса примет вид, показанный на рисунке 33. Результат запроса показан на рисунке 34.

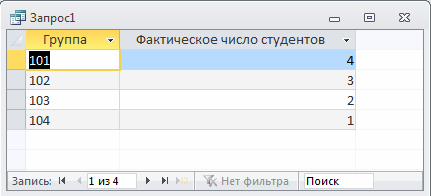


**Рисунок 33 -** Запрос с групповой операцией подсчета числа студентов в группе



**Рисунок 34 -** Результат подсчета числа студентов в группе

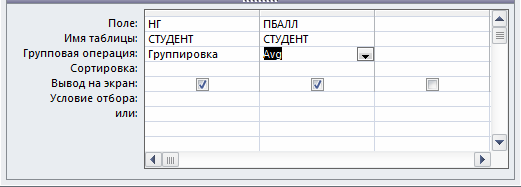
Подпись поля "Count\_HC" можно заменить на "Фактическое число студентов". Для ввода этой подписи в бланке запроса установим на поле НС курсор мыши и нажмем правую кнопку. В контекстно-зависимом меню выберем ко- манду **Свойства** . В **Окне свойств** наберем в строке **Подпись** "Фактическое число студентов". Таблица результата после доработки запроса показана на рисунке 35. Сохраним запрос под именем "Число студентов в группах".



**Рисунок 35 -** Таблица результата с измененной подписью поля

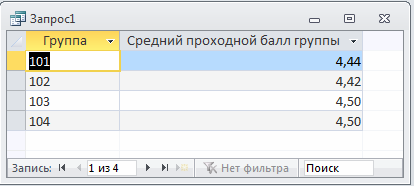
**Запрос с функцией Avg**

Подсчитаем средний проходной балл в группе. Сформируем запрос на выборку для таблицы СТУДЕНТ с функцией Avg для поля ПБАЛЛ (проходной балл студента). В бланке запроса заполним поля, как показано на рисунке 36



**Рисунок 36 -** Запрос с групповой операцией подсчета среднего значения

Для ограничения точности результата двумя знаками выберем в **Окне свойств** в строке **Формат поля** значение **Фиксированный**. Результат выполнения запроса представлен на рисунке 37.



**Рисунок 37 -** Результат подсчета среднего значения

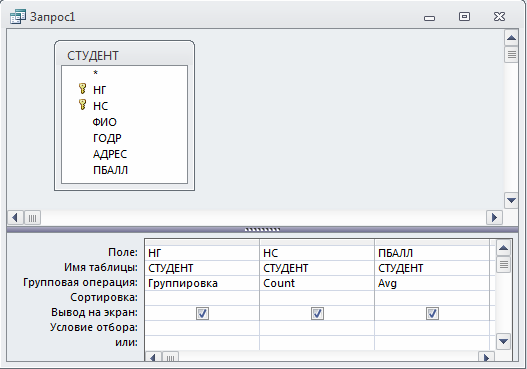
Сохраним этот запрос под именем "Средний проходной балл группы"

**Запрос с несколькими групповыми функциями**

Выполним расчет числа студентов и среднего проходного балла в группе в одном запросе. Это возможно, т. к. группы записей в обоих случаях формируются одинаково (рисунок 38). Сохраним этот запрос под именем "Число студентов и средний ПБАЛЛ группы".

**Задание условий отбора в запросах с групповыми операциями**

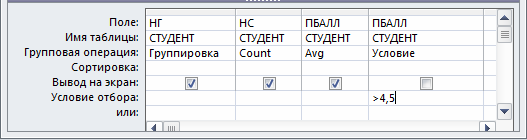
В запрос с групповыми операциями можно включать поля для задания условий отбора записей из таблиц.



**Рисунок 38 -** Запрос с одновременным использованием двух групповых функций

Подсчитаем число студентов в каждой из групп с проходным баллом больше 4,7.

Для этого в запрос **Число студентов и средний ПБАЛЛ группы** вторично включим поле ПБАЛЛ и в строке **Групповые операции** заменим значение **Группировка** на значение **Условие**, выбрав его из списка. После этого введем в строку **Условие отбора** ">4,5" (рисунок 39).



**Рисунок 39 -** Запрос с групповыми операциями и полем, введенным для определения условий отбора записей из таблицы СТУДЕНТ

Заметим, что средний балл в этом запросе также вычисляется только для студентов с проходным баллом, превосходящим 4,7.

Условие отбора, заданное в поле, по которому проводится группировка, или в поле, где записана функция группировки, позволяет отобрать только нужные группы записей, например, группы студентов с заданным номером или с за- данным средним проходным баллом.

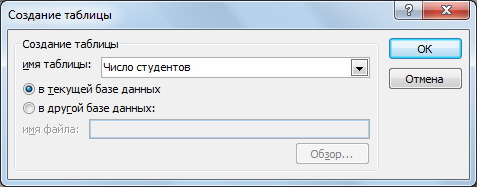
**Конструирование запроса на создание таблицы**

Запрос на *создание таблицы* используется для сохранения результата запроса. Этот вид запроса основан на *запросе на выборку*, но, в отличие от него, сохраняет таблицу с результатами запроса.

Необходимость в сохранении результатов запроса возникает, например, когда невозможно построить запрос непосредственно на другом запросе. К этому случаю относится, в частности, построение запроса на обновление полей на основе запроса с операцией группировки.

Сформируем запрос на создание таблицы на примере ранее полученного запроса на выборку с групповыми вычислениями **Число студентов в группах** (см. рис. 33-35).

В области навигации вызовем названный запрос в режиме конструктора запросов. Преобразуем этот запрос в запрос на создание таблицы, выбрав тип запроса на панели конструктора **Создание таблицы**. В окне **Создание таблицы** введем имя создаваемой таблицы "Число студентов" (рисунок 40).



**Рисунок 40 -** Определение имени таблицы, создаваемой в запросе

Для того, чтобы просмотреть, какие записи будут помещены в новую таблицу, щелкните по кнопке панели инструментов **Выполнить**. Выполните запрос, чтобы таблица ЧИСЛО СТУДЕНТОВ была сохранена в базе данных. Теперь эту таблицу можно увидеть в списке таблиц окна БД.

**Упражнение.** Преобразуйте запрос на выборку Средний проходной балл группы (см. рис. 36, 7.45) в запрос на создание таблицы, а создаваемую таблицу назовите “СРБАЛЛ”.

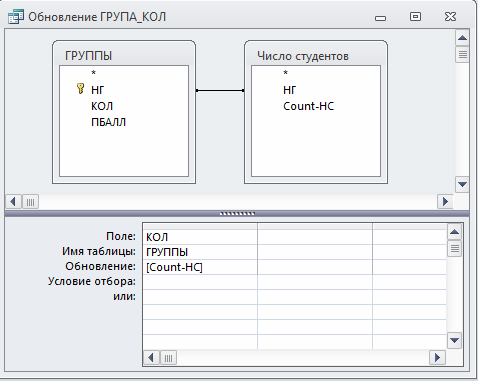
**Конструирование запроса на обновление**

**Обновление полей значениями, рассчитанными с использованием групповых операций**

Рассмотрим технологию создания запроса на обновление на примере обновления поля КОЛ (количество студентов группы в таблице ГРУППА).

Количество студентов в группах ранее было подсчитано в запросе на выборку **Число студентов в группах** (см. рис. 33-35) с использованием статистической функции Count. Запрос на обновление непосредственно на таком запросе построить нельзя. Поэтому используем для обновления не сам запрос, а таблицу ЧИСЛО СТУДЕНТОВ, полученную по запросу на создание таблицы в предыдущем пункте.

Для формирования запроса на обновление сначала создадим запрос на выбор- ку на основе двух таблиц: обновляемой таблицы ГРУППА и таблицы ЧИСЛО СТУДЕНТОВ, содержащей данные для обновления. Заметим, что в подсхеме данных запроса автоматически устанавливается связь этих таблиц по полю с именем НГ. Для преобразования запроса на выборку в запрос на обновление (рисунок 41) выберем на панели конструктора тип запроса **Обновление**.



**Рисунок 41 -** Запрос на обновление таблицы ГРУППА

Заполним бланк запроса. Перетащим обновляемое поле КОЛ из списка таблицы ГРУППА. В строке **Обновление** введем имя поля "Count\_HC" (таблицы ЧИСЛО СТУДЕНТОВ), из которого выбираются значения для обновления. Имя поля вводится в квадратных скобках.

Запрос можно выполнить, не выходя из режима конструктора. Содержимое обновляемого поля КОЛ можно просмотреть в режиме таблицы до и после выполнения запроса. Для последующего использования подготовленного запроса сохраним его под именем "Обновление ГРУППА\_КОЛ".

**Упражнение**

1. Произведите обновление поля ПБАЛЛ – средний проходной балл в таблице ГРУППА значениями из ранее созданной таблицы СРБАЛЛ.
2. Произведите обновление поля СРБАЛЛ-ГР – средняя оценка в группе по предмету в таблице ИЗУЧЕНИЕ. Для выполнения задания:
   * создайте запрос к таблице УСПЕВАЕМОСТЬ для расчета средней оценки в группе по предмету и сохраните результат в таблице, для чего группировку произведите по двум полям: НГ – номер группы и КП – код предмета:
   * обновите поле СРБАЛЛ-ГР в таблице ИЗУЧЕНИЕ, используя со- храненный результат.

**Использование выражений в запросе на обновление**

Рассмотрим формирование запроса на обновление с использованием выраже- ния на примере заполнения поля ЧАСЫ для лекционных занятий в таблице ИЗУЧЕНИЕ. Пусть поле ЧАСЫ должно обновляться данными, вычисляемы- ми на основе полей ЛЕК (часы лекций) и ЧС (число семестров) из таблицы ПРЕДМЕТ. Расчетное число часов по лекциям определим по формуле ЛЕК/ЧС.

В соответствии с задачей в записях лекционных занятий таблицы ИЗУЧЕНИЕ необходимо обновить поле ЧАСЫ расчетным числом часов. Записи о лекци- онных занятиях можно выбрать по значению поля ВИДЗ этой таблицы, т. к. в нем указан вид занятия. Данные для расчета среднего числа часов содержатся в таблице ПРЕДМЕТ. Таким образом запрос должен строиться на основе таб- лиц ИЗУЧЕНИЕ и ПРЕДМЕТ.

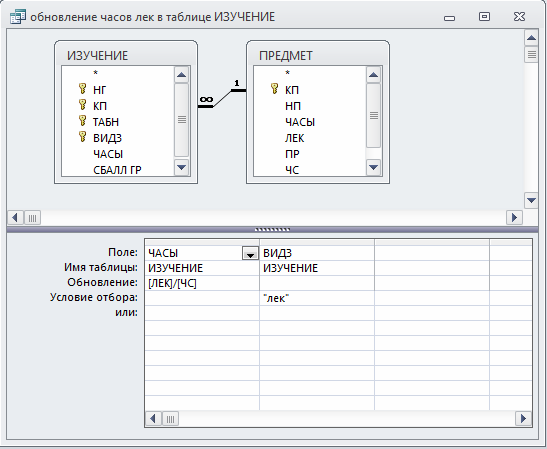
Создадим сначала запрос на выборку на основе таблиц ИЗУЧЕНИЕ и ПРЕД- МЕТ. Затем преобразуем его в запрос на обновление, нажав соответствующую кнопку панели инструментов.

Включим в бланк запроса обновляемое поле ЧАСЫ таблицы ИЗУЧЕНИЕ. В строке **Обновление** для этого поля введем выражение [ЛЕК]/[ЧС]. Для отбора в таблице ИЗУЧЕНИЕ обновляемых записей о лекционных занятиях в бланк запроса включим поле ВИДЗ и укажем в поле **Условия отбора** значение "лек".

Окончательно сформированный запрос показан на рисунке 42.

Выполним запрос, нажав кнопку **Выполнить.** В диалоговом окне появится сообщение о числе обновляемых записей.

Чтобы видеть результаты обновления в таблице ИЗУЧЕНИЕ, откройте ее одновременно с запросом.



**Рисунок 42 -** Запрос на обновление поля значениями, вычисляемыми по данным из другой таблицы

**Конструирование перекрестного запроса**

Создание перекрестного запроса, который позволяет получить данные в форме, подобной электронной таблице, с помощью мастера было рассмотрено выше. Однако такой запрос несложно построить полностью в режиме конструктора.

Построение запроса начинается как обычно, например, выбрать вкладку **Создание/ Конструктор запросов** в окне базы данных. В окне конструктора начинается создание запроса на выборку. В любой момент строящийся запрос на выборку может быть преобразован в перекрестный запрос. Для этого надо выбрать тип запроса **Перекрестный** на панели.

Ниже рассмотрим действия пользователя при работе с перекрестным запросом в режиме конструктора на примере.

Воспользуемся в качестве примера перекрестным запросом **Изучение предметов группами**, полученным мастером перекрестных запросов. Для большей информативности полученной перекрестной таблицы заменим в ней коды предметов их наименованиями. Полю, содержащему результат суммирования по строкам, дадим пользовательское имя "Всего часов".

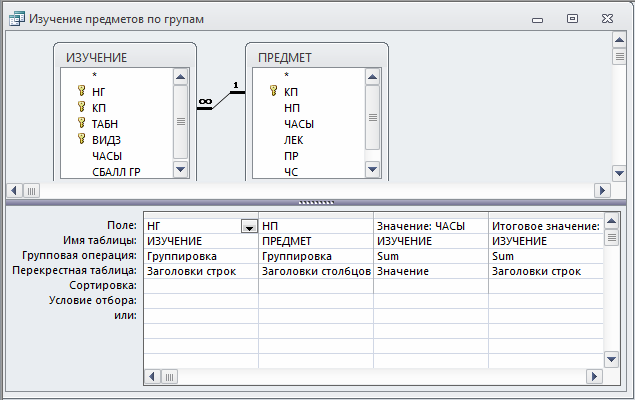
Для выполнения перечисленных преобразований откроем перекрестный запрос **Изучение предметов группами** в режиме конструктора.

Поле НП (наименование предмета) размещено в таблице ПРЕДМЕТ, поэтому ее нужно добавить к разрабатываемому запросу. Для этого, находясь в окне конструктора, нажмем кнопку **Отобразить таблицу**.

Теперь схема данных запроса состоит из таблиц ПРЕДМЕТ и ИЗУЧЕНИЕ, связанных по полю КП (код предмета) отношением один-ко-многим (рисунок 43).

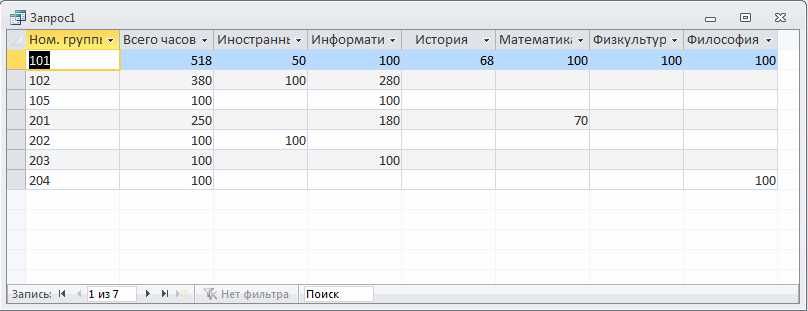
Заменим в бланке запроса поле КП на поле НП таблицы ПРЕДМЕТ. Для этого щелкнем правой кнопкой мыши на области отображения полей таблиц и выберем **Имена таблиц**, чтобы получить в бланке информацию о принадлежности поля к таблице. Далее в поле КП в строке **Имя таблицы** нажмем кнопку списка и выберем поле ПРЕДМЕТ, а в строке **Поле** - поле НП.

Для изменения подписи поля ИТОГОВОЕ ЗНАЧЕНИЕ, содержащего сумму по строкам, щелкнем правой кнопкой мыши, находясь в зоне этого поля. В от- крывшемся контекстно-зависимом меню выберем пункт **Свойства**. В окне **Свойства** введем в строку **Подпись** "Всего часов". Окончательно сформированный перекрестный запрос приведен на рисунке 44.



**Рисунок 43 -** Перекрестный запрос в режиме конструктора

Результат выполнения полученного перекрестного запроса приведен на рисунке 44.



**Рисунок 44 -** Результат выполнения перекрестного запроса

**Решение задач на основе нескольких запросов**

Выше рассматривались примеры простых задач, решение которых осуществлялось выполнением одного запроса. В этом случае запросом реализуется весь алгоритм формирования результата на основе входных данных из таблиц базы и параметров задачи.



**Рисунок 45 -** Функционально-технологическая схема задачи из двух по- следовательно выполняемых запросов

Для описания алгоритма задач, реализуемых одним запросом, обычно достаточно словесного описания действий. Целесообразно также использовать функционально-технологическую схему, на которой указываются входные и выходные таблицы данных.

Более сложные задачи требуют последовательного выполнения нескольких запросов. Каждый из запросов имеет свои входные и выходные данные. В простейшем случае выходные данные предшествующего запроса являются входными для следующего построенного на нем запроса, и, только выполнив последний запрос в цепочке запросов построенных друг на друге, вы инициируете последовательное выполнение всех запросов цепочки и полное решение задачи. Функционально-технологическая схема задачи, решаемой с помощью двух последовательно выполняемых запросов, в общем виде приведена на рисунке 45.

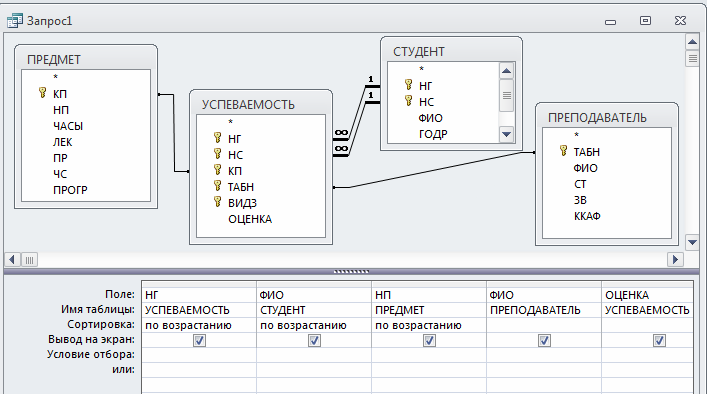
Запросы в Access являются мощным средством решения различных задач. При этом возможно построение сложных запросов, в том числе построенных на других запросах. Ниже рассматривается построение таких запросов, а также реализация задачи последовательно выполняющимися запросами.

**Построение запроса на основе другого запроса**

Выполним анализ оценок, полученных студентами по различным предметам. Например, подсчитаем число оценок (2,3,4,5) по каждому из предметов.

Создадим сначала многотабличный запрос на выборку на основе таблиц СТУДЕНТ, УСПЕВАЕМОСТЬ, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ, формирующий сведения об оценках, полученных студентами по различным предметам. Для этого в режиме конструктора создадим схему данных запроса и бланк, как показано на рисунке 46. Сохраним этот запрос с именем "оценки".

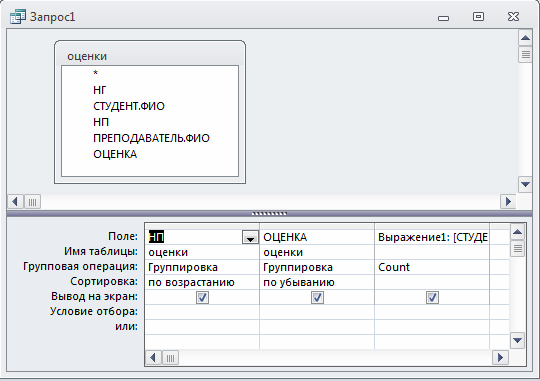
В результате выполнения этого запроса будет получена таблица, источником записей которой является таблица УСПЕВАЕМОСТЬ, а расшифровывающие данные выбираются из таблиц: ПРЕДМЕТ, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ. Таким образом, каждая строка результата будет содержать информацию об одной оценке, полученной студентом по указанному в строке предмету. Число строк в таблице запроса будет равно числу строк в таблице УСПЕВАЕМОСТЬ.



**Рисунок 46 -** Многотабличный запрос об оценках студентов

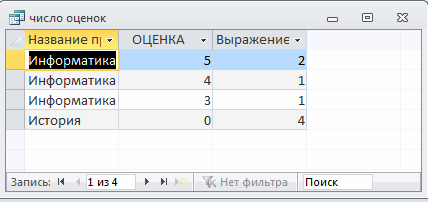
Для подсчета числа различных оценок (2,3,4,5) по каждому из предметов на основе этого запроса создадим новый запрос - **Число оценок**. При создании нового запроса в окне **Отразить таблицу/Добавление таблицы** на вкладке **Запросы** выберем из списка запрос **оценки**. Заполним бланк запроса, как показано на рисунке 47.

Результат выполнения запроса **Число оценок** приведен на рисунке 48, где в столбце **Выражение 1** отображено количество оценок, полученных по каждому предмету.



**Рисунок 47 -** Построение запроса на основе запроса **оценки**

**Замечание.** Нет необходимости предварительно выполнять запрос (**оценки**), на основе которого выполняется другой запрос (**Число оценок**). Выполнение вложенного запроса инициируется системой при выполнении запроса, построенного на нем.



**Рисунок 48 -** Результаты запроса, построенного на другом запросе

**Упражнения**

* + - * Создайте на основе запроса оценки, приведенного на рисунке 46, запрос для анализа оценок, выставленных каждым из преподавателей. Результат запроса должен содержать количество оценок (2,3,4,5), выставленных каждым преподавателем. Подпись столбца с результатами выполнения групповой операции Count Выражение1 замените на Количество оценок
      * Создайте на основе запроса оценки запрос для определения числа студентов, получивших 2,3,4 или 5 по предмету, задаваемому в диалоге с пользователем
      * Создайте на основе запроса оценки запрос для подсчета числа студентов в группе, получивших 2 (или другую заданную оценку) по каждому предмету. Предусмотрите ввод номера группы и оценки в диалоге с пользователем
      * Создайте на основе запроса оценки запрос для подсчета средней оценки в группе по каждому предмету; средней величины оценок, выставленных преподавателем; средней успеваемости по каждому предмету

**Решение задачи, требующей выполнения нескольких запросов и сохранения промежуточных результатов**

Пусть необходимо определить среднюю нагрузку преподавателя кафедры в текущем семестре. Для этого необходимо подсчитать число преподавателей кафедры, затем общее количество часов занятий, проводимых кафедрой, и завершить решение задачи расчетом средней нагрузки преподавателя.

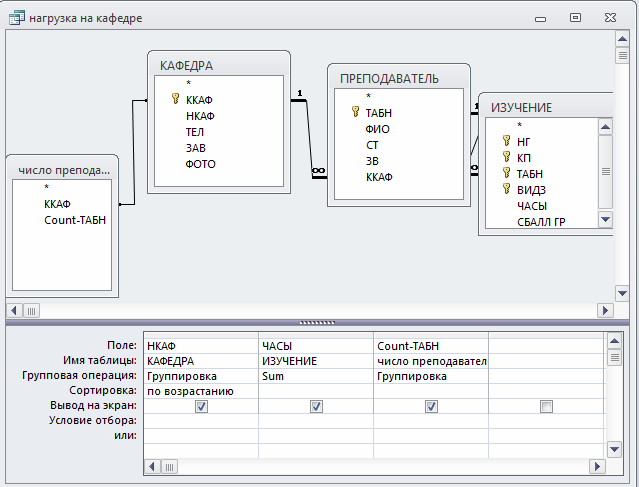
Подготовим и последовательно выполним соответствующие запросы.

**Первый запрос**. Создадим первый запрос на выборку, в котором по таблице ПРЕПОДАВАТЕЛЬ с помощью функции Count подсчитаем число преподавателей по кафедрам. Сохраним запрос под именем "Число преподавателей кафедры".

**Второй запрос**. Подготовим второй запрос на выборку для подсчета общего числа часов занятий, проводимых каждой кафедрой.

Этот запрос построим на базе таблиц ПРЕПОДАВАТЕЛЬ, ИЗУЧЕНИЕ, КА- ФЕДРА и запроса **Число преподавателей кафедры** (рисунок 47). Таблицы ПРЕПОДАВАТЕЛЬ и ИЗУЧЕНИЕ нужны для суммирования числа часов занятий, проводимых преподавателями каждой кафедры. Таблица КАФЕДРА необходима для включения в результат наименования кафедры, а запрос **Число преподавателей кафедры** - для включения в результат числа преподавателей на кафедре.

Записи этого запроса формируются на основе записей таблицы ИЗУЧЕНИЕ, причем число записей до проведения группировки равно числу записей в этой таблице. В результате запроса к каждой записи добавляется наименование кафедры НКАФ, по которому и производится группировка. Число преподавателей кафедры Count\_TAБH никак не нарушает требуемого объединения записей в группы, поскольку для каждой кафедры является единственным. Число записей в таблице результата запроса равно числу кафедр.

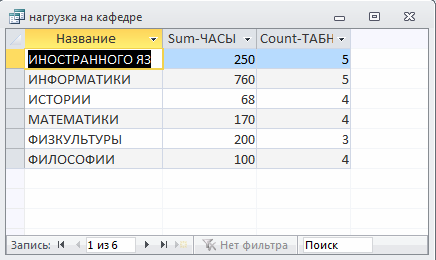


**Рисунок 49 -** Запрос для подсчета общего числа часов занятий, проводимых каждой кафедрой

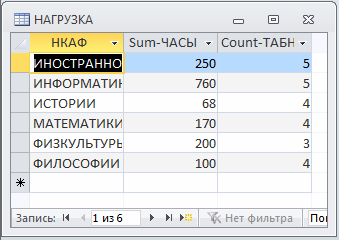
В рамках данного запроса нельзя сразу вычислить среднюю нагрузку преподавателя, поскольку число преподавателей кафедры является результатом выполнения групповой операции. Использование результатов выполнения запроса с групповой операцией не допускается в вычисляемых полях. Поэтому необходимо сохранить результаты выполнения второго запроса в новой таблице и на ее основе построить следующий запрос, в котором будет произведен расчет средней нагрузки преподавателя.

Чтобы сохранить полученные результаты, преобразуем второй запрос на выборку в запрос на создание таблицы. Таблице, которая будет создана запросом, присвоим имя "Нагрузка". Запрос сохраним под именем " Нагрузка на кафедре" (см. рисунок 49).

Второй запрос в режиме таблицы приведен на рисунке 50. Таблица НАГРУЗКА, сохраняемая в базе данных, содержащая результат решения этого запроса, приведена на рисунке 51.



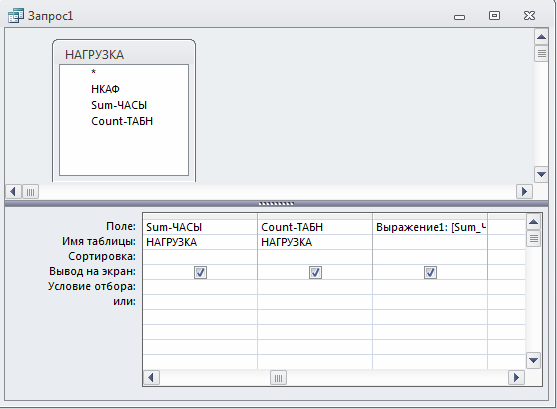
**Рисунок 50 -** Запрос **Нагрузка на кафедре** в режиме таблицы



**Рисунок 51 -** Сохраненная таблица НАГРУЗКА с результатами второго запроса

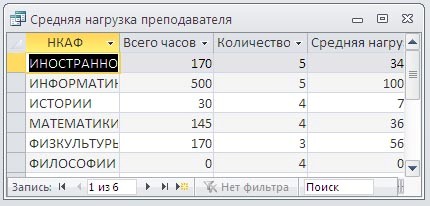
**Третий запрос.** Для окончательного решения задачи расчета средней нагрузки преподавателя кафедры подготовим на базе таблицы НАГРУЗКА третий запрос на выборку с вычисляемым полем. Для создания вычисляемого поля, рассчитывающего среднюю нагрузку преподавателя, в строку **Поле** пустого столбца введем выражение [Sum\_ЧАСЫ]/[Count\_ТАБН]. Третий запрос на выборку с вычисляемым полем представлен на рисунке 52.

В таблице результата следует изменить заголовок столбца **Выражение1**, формируемый по умолчанию для вычисляемого поля, и его формат (для получения результата с округлением до целого). Для этого вызовем свойства поля с помощью контекстно-зависимого меню. Зададим в качестве подписи поля значение "Средняя нагрузка преподавателя", формат поля определим как фиксированный, а параметру **Число десятичных знаков** присвоим значение "0"



**Рисунок 52 -** Запрос, завершающий задачу расчета средней нагрузки

Результаты выполнения запроса после изменения подписей полей в свойствах приведены на рисунке 53.



**Рисунок 53 -** Результаты расчета средней нагрузки преподавателей по кафедрам

**Процесс решения задачи**. Для решения поставленной задачи необходимо по- следовательно выполнить второй запрос - **Нагрузка на кафедре** (см. рисунок 47), а затем третий запрос - (см. рисунок 53). Первый запрос **Число преподавателей кафедры** выполнится автоматически при выполнении запроса **Нагрузка на кафедре**.

В этом разделе на конкретных примерах рассматривалась технология конструирования запросов различного вида. Дано описание конкретных действий пользователя в процессе конструирования запросов.

Подробно описан процесс конструирования однотабличного и многотабличного запроса, показано формирование вычисляемых полей, использование групповых операций и параметров запроса.

Рассмотрено обновление таблиц с помощью запроса. Кроме того, в конце раздела приведен пример решения задачи на основе нескольких запросов.

**Контрольные вопросы**

1. Что такое запрос?
2. Как создать запрос в БД Access?
3. Из чего состоит окно конструктора запросов?
4. Что такое запрос на выборку?
5. Как создать запрос с вводом параметров?
6. Как создать с помощью запроса новую таблицу БД?
7. Что такое запрос на обновление?
8. Что такое запрос на удаление?

# Практическая работа № 5

**Проектирование и создание однотабличной формы.**

**Цель работы:** Рассмотреть на примере базы данных «Учебный процесс» технологию

загрузки базы данных с использованием форм. Научиться проектировать однотабличные формы. Научиться создавать и редактировать однотабличные формы в режиме конструктора.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

**Определение технологии загрузки базы данных на примере.**

Рассмотрим технологию загрузки на примере базы данных **Учебный процесс,** в которой связи между таблицами соответствуют созданной ранее схеме данных.

Таблицы-объекты загрузки

Определим объекты загрузки — подмножества взаимосвязанных таблиц, подлежащих загрузке из одного документа:

1. Таблицы группа — СТУДЕНТ. Загрузка записей этих таблиц будет производиться одновременно из одного документа Список студентов группы, что обеспечит формирование связей между записями о студентах с соответствующей записью о группе. При этом обеспечивается однократный ввод реквизитов группы и значений НГ для всех студентов группы.

2. Таблицы КАФЕДРА — ПРЕПОДАВАТЕЛЬ. Загрузка записей этих таблиц также будет производиться одновременно из одного документа Список преподавателей кафедры, что позволит сформировать связи между записями о преподавателях с соответствующей записью о кафедре. При этом также обеспечивается однократный ввод реквизитов КАФЕДРЫ и значений ККАФ для всех преподавателей кафедры.

3. Таблица ПРЕДМЕТ. Эта таблица может быть загружена из документа, содержащего перечень изучаемых предметов.

1. Таблица ИЗУЧЕНИЕ. Загрузка этой подчиненной таблицы будет производиться из плана проведения занятий в группах. Для этого достаточно ввести составной идентификатор занятия, не вводя наименование предмета и фамилию преподавателя.
2. Таблица УСПЕВАЕМОСТЬ. Таблица будет загружена из заполненной экзаменационной ведомости группы. Для выполнения загрузки достаточно вводить составной идентификатор, включающий идентификатор занятия и студента, не вводя фамилию студента, наименование предмета и фамилию преподавателя.

**Практическая часть**

**Последовательность загрузки таблиц**

Загрузка первых трех объектов загрузки ГРУППА — СТУДЕНТ, КАФЕДРА — ПРЕПОДАВАТЕЛЬ, ПРЕДМЕТ может производиться в любой последовательности, т. к. эти группы таблиц не находятся в подчиненных отношениях в схеме данных.

Загрузка таблицы ИЗУЧЕНИЕ должна производиться после загрузки таблиц ГРУППА, ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, т. к. она подчинена этим таблицам в схеме данных.

Загрузка таблицы УСПЕВАЕМОСТЬ может производиться только после загрузки таблиц СТУДЕНТ и ИЗУЧЕНИЕ ПО причине подчиненности этим таблицам в схеме данных.

Технология загрузки базы данных Учебный процесс может быть представлена в виде таблицы (табл. 1). Таким образом, в результате определена последовательность этапов загрузки базы данных Учебный процесс, а также объекты загрузки на отдельных этапах и соответствующие документы-источники данных.

***Таблица 1. Технология загрузки базы данных*** *Учебный процесс*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Таблицы БД — объекты загрузки** | **Документы-источники** | **Вид информации** | **Этап загрузки** | **Приме­чание** |
| ПРЕДМЕТ | Учебный план, программа | Справочная | **I** | Незави­симые  этапы |
| ГРУППА-СТУДЕНТ | Список студентов группы | Справочная | **I** |
| КАФЕДРА-  ПРЕПОЦАВАТЕЛЬ | Список преподава­телей кафедры | Справочная | **I** |
| ИЗУЧЕНИЕ | План занятий группы | Учетная | **II** |  |
| УСПЕВАЕМОСТЬ | Экзаменационная ведомость | Учетная | III |  |

После определения этапов загрузки базы данных можно приступить к определению подсхемы данных для каждого этапа загрузки, проектированию макета форм и их созданию средствами Access.

Технология разработки однотабличной формы

Рассмотрим технологию разработки однотабличной формы, которая включает процесс проектирования формы и конструирование в среде Access.

На этапе проектирования однотабличной формы определяются требования к форме в соответствии с изложенным выше и далее проектируется макет экранной формы с размещением в нем полей таблицы и других элементов. После проектирования формы осуществляется конструирование экранной формы средствами Access.

Рассмотрим технологию разработки однотабличной формы для ввода и корректировки данных на примере таблицы предмет базы данных Учебный процесс в соответствии с этапами загрузки, определенными выше.

**Проектирование однотабличной формы**

**Определение подсхемы**

В результате загрузки данных об изучаемых предметах должны формироваться записи только таблицы ПРЕДМЕТ (объект загрузки). Эта таблица не подчинена другим таблицам. Поэтому в процессе загрузки записей таблицы предмет не должны устанавливаться связи с записями других таблиц, а в подсхему для формы достаточно включить только одну эту таблицу.

**Определение структуры формы**

Определим структуру экранной формы, которую назовем ПРЕДМЕТ-ПРОГРАММА. *Макет* формы для ввода данных в таблицу ПРЕДМЕТ должен содержать: ключевое поле КП и все описательные поля, в том числе поле типа **MEMO**для размещения большого текста программы по предмету.

Таким образом, структуру формы ПРЕДМЕТ-ПРОГРАММА определяют:

* тип формы — однотабличная (простая);
* источник записей формы — таблица ПРЕДМЕТ;
* наличие большого поля для текста программы.

Все поля таблицы ПРЕДМЕТ будут размещаться в области данных формы. Ширина поля текста программы должна быть достаточной для отображения абзаца текста.

**Создание однотабличной формы**

Форма на основе таблицы может быть построена как самостоятельная для загрузки, просмотра и корректировки таблиц, а также как вспомогательная для включения в какую-либо составную форму.

Любая форма, с помощью которой можно просматривать, вводить или редактировать записи таблиц БД, должна быть предварительно спроектирована и далее сконструирована средствами Access.

Для создания формы могут быть использованы мастера Access.Однако, точное формирование макета формы в соответствии с требованиями, выработанными пользователем в процессе ее проектирования, обеспечивается средствами конструирования форм. Ниже рассматриваются основные понятия и техника конструирования однотабличных форм.

**Конструирование формы**

Для конструирования форм в Access используется Конструктор форм. При конструировании однотабличной формы определяется таблица БД, на основе которой создается форма, выбираются поля таблицы, которые должны быть представлены в форме, осуществляется их размещение в макете формы, создаются вычисляемые поля и другие графические элементы: кнопки, выключатели, элементы оформления, поясняющий текст, рисунки. Для настройки различных элементов форм используется типовой набор их свойств.

**Области и элементы формы в режиме конструктора**

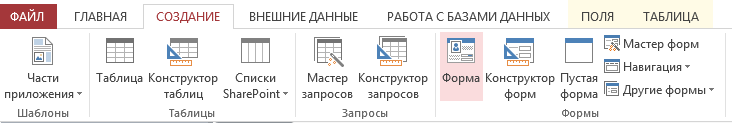
Форма в режиме Конструктора форм имеет три области: **Область данных, Заголовок формы**, и **Примечание формы**, которые могут быть образованы по команде меню **Вид - Заголовок**/**примечание формы**. Области формы наполняются различными графическими объектами.

**Элементы или графические объекты.** Графические объекты, связанные с записями таблиц и предназначенные для отображения данных некоторого поля, называются *элементами управления*. Основными типами элементами управления: **Поле, Поле со списком, Список**. Тип элемента управления, выбираемый для поля *по умолчанию*, определяется в свойствах поля таблицы базы данных, с которым связано поле формы. Задается это свойство при определенных типах данных поля в режиме конструктора таблиц на вкладке **Подстановка.**

Графические объекты, не связанные с таблицами или запросами, предназначены прежде всего для создания макета формы и содержат *надписи* полей (пользовательские названия реквизитов).

**Создание однотабличной формы с помощью инструмента «Форма»**

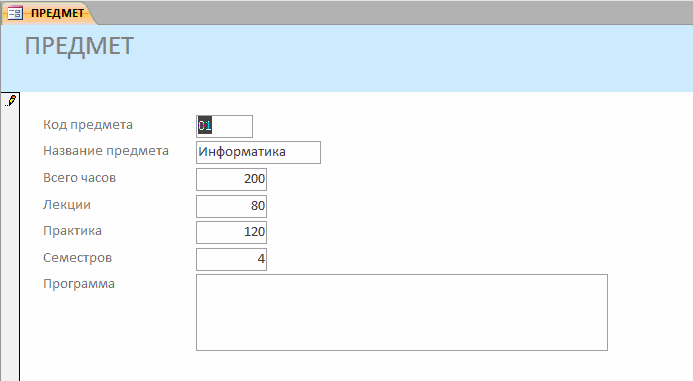
В области переходов щелкните таблицу или запрос с данными, которые должны отображаться в форме. На вкладке **Создать** в группе **Формы** нажмите кнопку **Форма.** (рисунок 54)



**Рисунок 54 -** Кнопка **Форма** на вкладке **Создать** в группе **Формы**

Будет создана новая форма и отображена в режиме макета. В режиме макета можно внести изменения в структуру формы при одновременном отображении данных. Например, можно настроить размер полей в соответствии с данными.

Работа команды «**Форма»** завершается отображением формы (рисунок 55). При этом появляется панель инструментов **Режим формы**. Кнопки этой панели по назначению аналогичны кнопкам панели **Таблица в режиме таблицы**.



**Рисунок 55 -** Результат работы команды **«Форма»** по таблице ПРЕДМЕТ

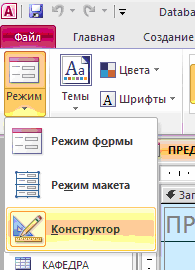
**Внимание.**

Подписи полей в форме соответствуют заданным в их свойствах при определении структуры таблицы. Подпись формы соответствует имени таблицы источника, при сохранении формы можно подтвердить это имя или изменить его.

**Редактирование формы**

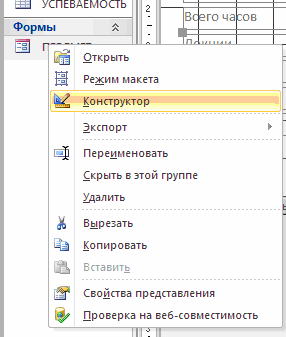
**Переход в режим конструктора формы.** Для уточнения текста надписей, местоположения, размера, шрифта и других параметров отображения элементов формы необходимо перейти в режим конструктора форм.

При открытой форме переход в режим конструктора можно осуществить нажатием в вкладке **Режимы** кнопки **Режим** (рисунок 56). Кнопка **Режим** обеспечена списком, развернув который можно выбрать необходимый режим представления формы.



**Рисунок 56 -** Список режимов кнопки **Режим**

Переход в режим конструктора можно осуществить также с помощью **Контекстного меню** данной формы (рисунок 57)



**Рис**унок 57- Переход в режим конструктора

После перехода в режим конструктора созданная форма откроется в окне конструктора форм (рисунок 58).

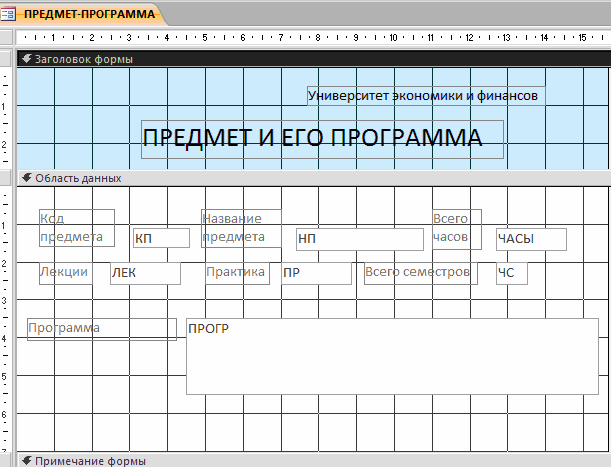


**Рисунок 58 -** Режим конструктора формы, полученной командой «**Форма»**

по таблице ПРЕДМЕТ

После выбора режима конструктора в окне Access появляются панель **Конструктор форм** и **Панель элементов**. Панель форматирования **Формат (Форма/ Отчет)** может быть вызвана при активном окне формы по команде меню **Вид - Панели инструментов - Формат (Форма/Отчет**).

**Создание заголовка.** Для ввода текста заголовка в полученную форму в окне конструктора (рисунок 59) расширим область заголовка формы, установив курсор мыши на границу области данных и перетаскивая эту границу на нужное расстояние.



**Рисунок 59 -** Форма ПРЕДМЕТ-ПРОГРАММА в режиме конструктора форм

Для ввода текста заголовка надо создать графический элемент **Надпись**. Начинается создание элемента щелчком мыши на кнопке панели элементов **Надпись**. Теперь на панели форматирования можно выбрать нужный шрифт и другие параметры оформления. Переместим курсор мыши на место начала текста. Нажмем кнопку мыши и, не отпуская ее, растянем рамку текста до нужного размера. Введем текст. Создание элемента **Надпись** завершается нажатием клавиши <Enter> или щелчком мыши вне рамки элемента.

Форматирование элемента **Надпись** может быть выполнено в любой момент. Для этого элемент надо выделить щелчком мыши внутри его рамки. Для изменения текста надписи курсор должен быть переведен на текст, при этом сам элемент остается невыделенным, а команды форматирования недоступны.

Элемент **Надписи** может быть перемещен в пределах области заголовка, могут быть также изменены размеры рамки элемента. Рамка помеченного элемента может быть растянута или сжата при размещение курсора на специальных точках рамки, в которых появляется двунаправленная стрелка. Перемещение возможно при появлении изображения руки.

**Замечание**

При установке курсора мыши на любой границе рамки курсор отображается в виде раскрытой ладони и тогда возможно перемещать элемента и за пределы области. При установке курсора в левом верхнем углу курсор отображается в виде указательного пальца и перемещение возможно только в пределах области, которая при этом может автоматически расширяться.

Для удаления элемента его надо выделить и нажать клавишу <Del>.

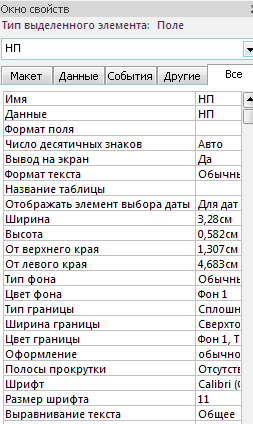
**Изменение надписей и отображения значений полей.** При редактировании связанных элементов **Поле** и **Надпись** , если между ними установлена связь, или аналогичной пары элементов, полученной с помощью кнопки **Список полей** на панели конструктора форм, следует иметь в виду, что независимое перемещение поля и его надписи возможно, только если курсор примет вид указательного пальца. В противном случае оба элемента перемещаются синхронно. Остальные действия по внесению изменений в эти элементы осуществляется аналогично рассмотренному при формировании элемента в области заголовка.

**Изменение свойств.** Редактирование формы и ее элементов может быть выполнено не только графическими средствами, но и путем изменения их свойств. Для этого необходимо с помощью двойного щелчка открыть **Окно свойств** необходимого элемента. А также его можно открыть с помощью контекстного меню. На рисунке 60 показана вкладка **Макет** в окне свойств **Поле: НП** .

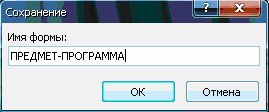
**Сохранение формы после редактирования.** По завершении редактирования формы она может быть сохранена. Если редактируемая форма еще не сохранялась, выполняется команда меню **Файл - Сохранить** или нажимается кнопка панели инструментов **Сохранить**. Можно сохранить форму и при ее закрытии командой **Файл - Закрыть** или нажатием кнопки (Закрыть) окна формы. Далее надо подтвердить необходимость их сохранения, и в диалоговом окне **Сохранение** ввести название (ПРЕДМЕТ-ПРОГРАММА) в текстовом поле **Имя формы** (рисунок 61).

Если редактируемая форма была ранее сохранена, то для сохранения измененной формы под новым именем над выполнить команду меню **Файл - Сохранить как**.

Окончательный вид отредактированной формы в режиме конструктора представлен на рисунке 59.



**Рисунок 60 -** Свойства для поля НП в форме, установленные мастером при создании формы для таблицы ПРЕДМЕТ



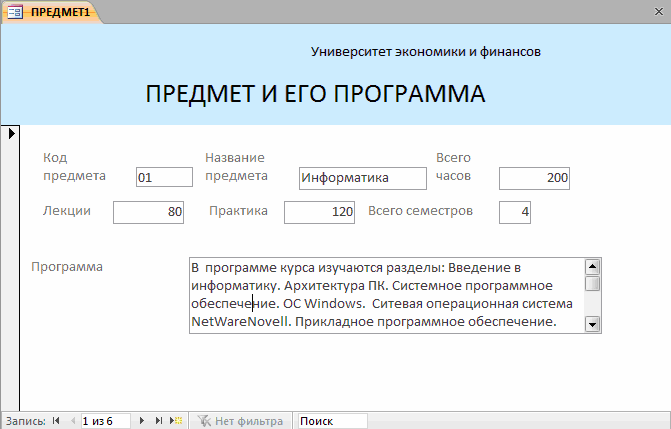
**Рисунок 61 -** Ввод имени формы при ее сохранении

**Работа с данными таблицы в режиме формы**

Завершив редактирование формы, приступим к работе с таблицей ПРЕДМЕТ через форму. Для перехода в режим формы из режима конструктора нужно нажать в вкладке **Режимы** кнопки **Режим** и выбрать **Режим формы**.

Для загрузки, просмотра и корректировки данных таблицы ПРЕДМЕТ через ранее сохраненную форму в окне: **база данных** в группе **Объекты** надо перейти к строке **Формы** и, выделив в рабочей области название формы ПРЕДМЕТ-ПРОГРАММА, нажать кнопку **Открыть.**

На рисунке 62 приводится форма ПРЕДМЕТ-ПРОГРАММА в режиме формы, в которой отображены данные из одной строки таблицы ПРЕДМЕТ.



**Рисунок 62 -** Форма ввода-вывода для работы с данными таблицы ПРЕДМЕТ

Данные для загрузки таблицы ПРЕДМЕТ приведены в соответствующей таблице приложения, как описано в практической работе №2.

Значения, вводимые в поля формы, должны соответствовать типам данных и их свойствам, заданным при определении структуры таблицы.

Для завершения создания (редактирования) записи таблицы ПРЕДМЕТ достаточно перейти к другой записи в поле номера записи внизу формы.

УПРАЖНЕНИЕ. Создайте однотабличную форму для таблицы кафедра с помощью мастера, отредактируйте ее в конструкторе и произведите добавление нескольких записей.

**Контрольные вопросы**

1. Что такое форма?
2. Сколько есть способов создания форм? Перечислите их?
3. В каком порядке необходимо загружать таблицы при создании форм?
4. Что определяется на этапе проектирования однотабличной формы?
5. Что должен содержать макет формы для ввода данных в таблицу?
6. Что должно включать в себя окно конструктора форм?
7. Перечислите основные элементы панели элементов в режиме конструктора форм?
8. При редактировании связанных элементов — Поле и Надпись, когда между ним установлена связь, следует иметь в виду, что…?
9. Через что проще всего создавать форму? Почему?
10. Через что можно отредактировать элементы формы?

# Практическая работа №6

Создание многотабличных форм

**Цели работы**: научиться проектировать многотабличные формы, создавать многотабличные формы с помощью мастера и дорабатывать их в режиме конструктора, включать подчиненные формы, компоновать поля форм для удобного интерфейса.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

**Технология разработки формы для загрузки двух взаимосвязанных таблиц**

Рассмотрим технологию разработки составной формы для загрузки двух таблиц, связанных одно – многозначными отношениями.

Для проектирования макета составной формы определим:

* Подсхему данных для формы;
* общую структуру экранной формы, т. е. ее макет в соответствии со структурой документа - источника загрузки и под схемой данных;
* состав и размещение полей, надписей и других элементов для каждой из частей составной формы.

На основе результатов проектирования произведем конструирование двухтабличной формы средствами Access.**Проектирование формы на основе двух таблиц**

Спроектируем форму для загрузки, просмотра и редактирования данных двух таблиц — ГРУППА и СТУДЕНТ. Документом – источником такой формы является Список студентов группы. Этот документ будет использован для одновременной загрузки двух таблиц: ГРУППА и СТУДЕНТ, которые в совокупности образуют *объект загрузки.*

**Практическая часть**

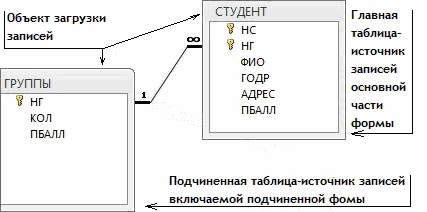
**I часть**

**Определение подсхемы данных для составной формы**

Поскольку объект загрузки: ГРУППА → СТУДЕНТ не подчиняется в схеме данных другим таблицам, подсхема, необходимая для построения формы, не должна включать других таблиц. Такая подсхема приведена на рисунке 63.

**Определение общей структуры составной формы**

В соответствии с приведенной подсхемой определим общую структуру составной формы, которую назовем СПИСОК ГРУППЫ.



**Рисунок 63 -** Подсхема данных для конструирования формы на базе таблиц ГРУППА → СТУДЕНТ

Для того чтобы обеспечить удобный ввод данных с документа, в форме предусмотрим основную часть с реквизитами группы и подчиненную с записями о студентах группы. Подчиненную форму назовем СПИСОК СТУДЕНТОВ.

Таким образом, составную форму СПИСОК ГРУППЫ определяют:

* *Тип формы* – многотабличная
* *Источник записей для основной части формы* – таблица ГРУППА
* *Включаемая подчиненная форма* – СПИСОК СТУДЕНТОВ Подчиненную форму СПИСОК СТУДЕНТОВ определяют:
* *Тип формы* – подчиненная, многозаписевая
* *Источник записей* – таблица СТУДЕНТ

На подсхеме (рис.5.1) показано назначение таблиц при создании формы.

**Размещение реквизитов основной и подчиненной формы**

Размещение реквизитов в основной части формы и подчиненной форме должно соответствовать входному документу «Список студентов группы».

В основной части составной формы СПИСОК ГРУППЫ вверху разместим реквизиты, соответствующие полям таблицы ГРУППА:

* Номер группы (НГ - ключ)
* Количество студентов (КОЛ)
* Средний проходной бал в группе (ПБАЛЛ)

В подчиненной форме СПИСОК СТУДЕНТОВ разместим в качестве заголовков столбцов многозаписевой формы названия реквизитов соответствующих полей таблицы СТУДЕНТ:

* Номер студента в группе (НС)
* Фамилия И. О. (ФИО)
* Год рождения (ГОДР)
* Адрес (АДРЕС)
* Средний балл при поступлении (ПБАЛЛ)

Заметим, что ключевое поле НГ не включено в подчиненную форму, т. к. поле связи НГ включено в основную часть формы.

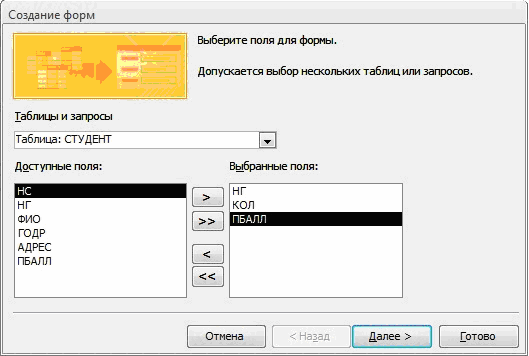
**Создание формы для двух таблиц с помощью мастера**

Осуществим средствами Access конструирование формы для одновременной загрузки и корректировки двух таблиц ГРУППА и СТУДЕНТ в базе данных

«Учебный процесс».

**Определение таблиц и полей для основной и включаемой частей формы**

В окне: **базы данных** выбираем в группе **Объекты** строку **Формы** и нажима ем кнопку **Создать**.



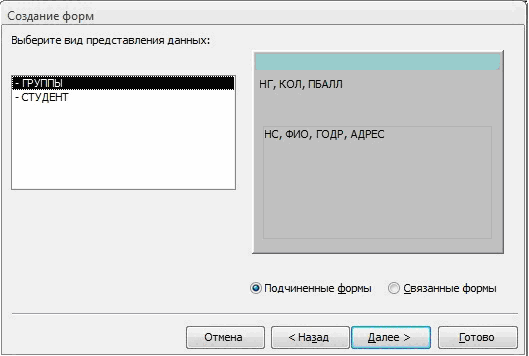
**Рисунок 64 -** Выбор полей для формы в окне мастера форм

В диалоговом окне **Новая форма** выбираем режим создания **Мастер форм** и таблицу ГРУППА, которая будет служить источником данных для основной части, создаваемой *многотабличной формы*.

В открывшемся окне **Создание форм** в списке **Таблицы/Запросы** будет уже отражена ранее выбранная таблица ГРУППА. Выберем для нее в списке **Доступные поля** те поля, которые вошли в спроектированный макет формы, перемещая их в область **Выбранные поля**. Выберем далее таблицу СТУДЕНТ и ее поля (рисунок 64). Эта таблица будет источником записей подчиненной формы, связанных с записью отображаемой в основной части формы.

**Выбор типа формы**

В следующем сеансе окна **Создание форм** отображается макет формы с перечнем полей в основной части формы и в подчиненной форме (рисунок 65). В этом окне уже будет выделена таблица ГРУППА – источник записей основной части формы. Таблица СТУДЕНТ – источник записей подчиненной формы. Для непосредственного включения подчиненной формы выберем первый вариант – **Подчиненные формы**.



**Рисунок 65 -** Выбор варианта явного включения подчиненной формы

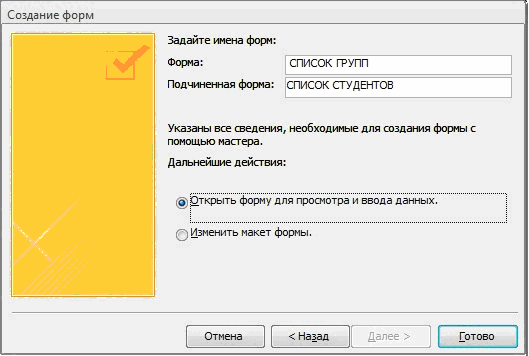
В следующем сеансе диалогового окна мастера выбираем внешний вид под-

чиненной формы **ленточный** для получения многозаписевой подчиненной формы и вывода в ней подписей полей (определенных в свойствах таблиц). В очередном сеансе диалогового окна выберем стиль оформления **Стандартный** с утопленными полями.

**Присвоение имени форме и ее открытие**

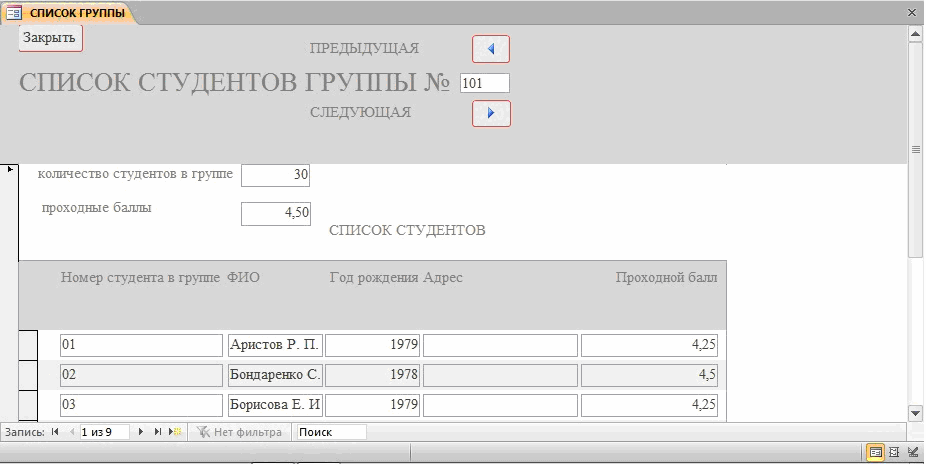
В последнем сеансе окна **Создание форм** (рисунок 66) введем имена (заголовки)

составной формы – СПИСОК ГРУППЫ и подчиненной формы – СПИСОК СТУДЕНТОВ. Выберем также дальнейшие действия мастера – **Открытие формы для просмотра и ввода данных**.



**Рисунок 66 -** Окно ввода имен форм и выбора дальнейших действий мастера

В соответствии с выбранными действиями после завершения работы мастера выводиться форма с данными из таблиц базы данных.



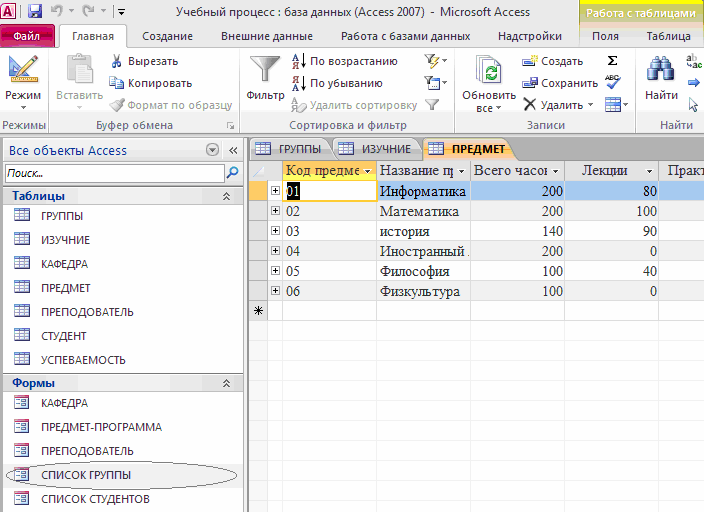
**Рисунок 67 -** Многотабличная форма, созданная мастером на основе таблиц ГРУППА и СТУДЕНТ

При этом в подчиненной форме выводятся те записи таблицы СТУДЕНТ, которые связаны с текущей записью таблицы ГРУППА, данные которой отображаются в основной части формы (рисунок 68).

Замечание. При работе с формой в случае необходимости можно отобразить данные подчиненной формы в виде таблицы с именами полей таблицы БД. Для этого в режиме формы выполняется команда **Вид - Таблица подчиненной формы**, работающая как переключатель. Курсор предварительно устанавливается на подчиненной форме.

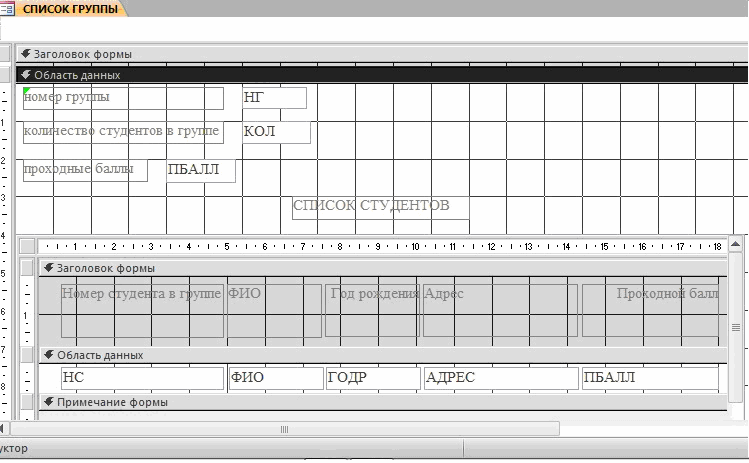
**Редактирование формы в режиме конструктора**

**Изменение надписей и размещения полей.** В окне: **базы данных** (рисунок 68) в области **Объекты** выделим строку **Формы**. Выберем для редактирования созданную ранее многотабличную форму СПИСОК ГРУППЫ и нажмем кнопку **Конструктор**. Если форма была открыта ранее в режиме просмотра, то для перехода в режим конструктора достаточно нажать кнопку **Вид** на панели конструктора форм.



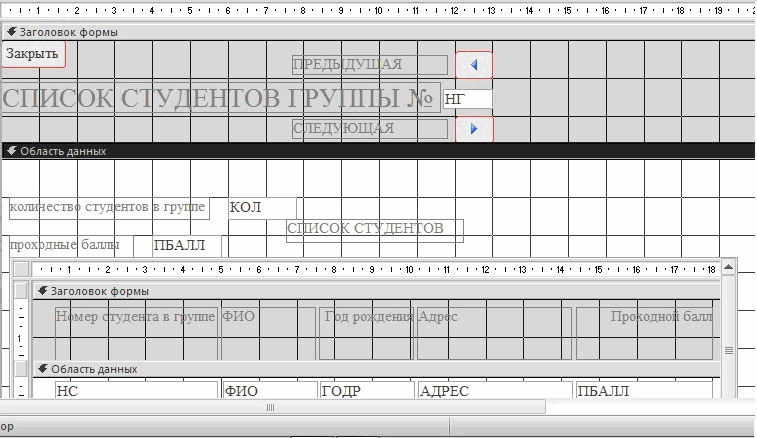
**Рисунок 68 -** Выбор формы в окне базы данных

В окне конструктора форм (рисунок 69) указано имя формы, СПИСОК ГРУППЫ, представлены поля с надписями основной части формы, размещенные в области данных, а также поля и надписи подчиненной формы СПИСОК СТУДЕНТОВ в рамке, созданной для нее мастером.



**Рисунок 69 -** Форма для двух таблиц ГРУППА и СТУДЕНТ в режиме конструктора

Произведем доработку формы СПИСОК ГРУППЫ, используя технику редактирования. Введем в область заголовка полное название формы СПИСОК СТУДЕНТОВ ГРУППЫ №, которое будет выводиться при распечатке формы и будет соответствовать макету документа. Формирование текста в области заголовка подробно было рассмотрено выше при конструировании однотабличной формы.



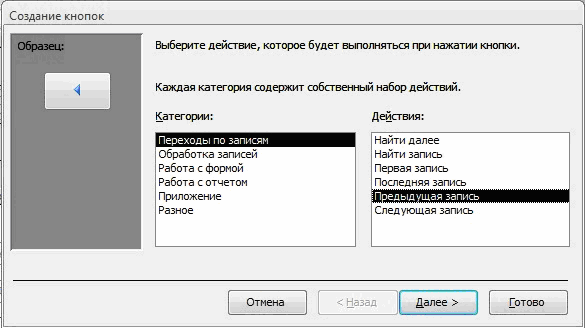
**Рисунок 70 -** Размещение элементов основной части формы после редактирования

Отмечая курсором мыши и перетаскивая отмеченные элементы, разместим поля так, как это показано на рисунке 70. Уточним подписи полей, шрифт и размеры полей и подписей, заголовок формы. Можно изменить размер любого элемента, перемещая границы его рамки. Ширину и высоту подписи в соответствии с размером и шрифтом текста можно задать, используя кнопку. **По размеру данных** на панели конструктора форм. Выравнивание предварительно выделенных элементов, например, по горизонтали выполняется командой меню **Формат – Выравнять - по нижнему краю** .

**Создание кнопок для перехода к другой записи**

Для того чтобы в форме можно было переходить к следующей и предыдущей записей записи таблицы ГРУППА, создадим соответствующие *кнопки управления* в основной части формы.

Нажмем на **Панели элементов** Access кнопку **Мастер элементов**, а затем используем инструмент **Кнопка**. После нажатия, переноса кнопки курсором мыши в нужное место и вычерчивания ее рамки запустится мастер кнопок **Создание кнопок** (рисунок 71).



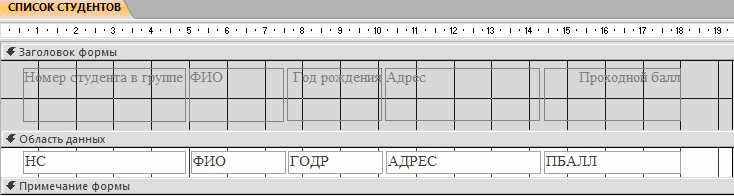
**Рисунок 71 -** Окно мастера кнопок

В окне мастера кнопок выберем действие, которое необходимо выполнять при нажатии кнопки. В группе **Категории** выберем строку **Переходы по записям**, в группе Действия выберем строку **Предыдущая запись**. В следующем сеансе диалогового окна можно выбрать вид кнопки: **Текст** или **Рисунок** и выбрать его из списка. Отметим флажки **Рисунок** и **Показать все рисунки**. Далее выбираем подходящий рисунок из списка, например, **Стрелка вверх** (синяя). После нажатия кнопки **Готово** кнопка с выбранным рисунком встраивается в форму. Аналогичные действия выполняются для встраивания кнопки перехода к последующей записи таблицы. При этом выбираются, соответственно, в группе **Действия** – строку **Следующая запись** и рисунок **Стрелка вниз** (синяя). Отредактируем размер в надписи кнопок для перехода к записи другой группы, записав – «ПРЕДЫДУЩАЯ», «СЛЕДУЮЩАЯ».

Для создания кнопки закрытия формы в группе **Категории** надо выбрать строку **Работа с формой**, а в группе **Действия** – **Закрыть форму**. После формирования кнопки заменим название ее название на «ЗАКРЫТЬ».

**Редактирование подчиненной формы**

Аналогичные действия по доработке выполним для подчиненной формы СПИСОК СТУДЕНТОВ. Перейдем к редактированию подчиненной формы, переводя курсор в область подчиненной формы (рисунок 72) или открывая подчиненную форму в окне базы данных.

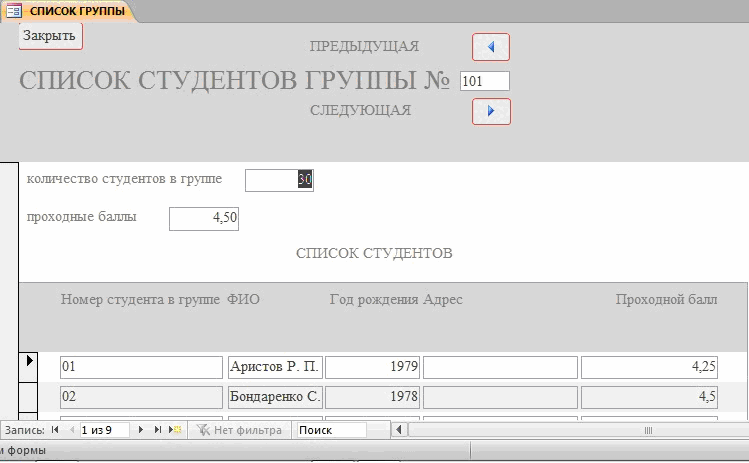


**Рисунок 72 -** Подчиненная форма СПИСОК СТУДЕНТОВ в режиме конструктора после доработки

Используя технику редактирования формы, удалим поле НГ, отображающее номер группы, т. к. это поле является полем связи и его достаточно сохранить в основной части формы. В подчиненной форме это поле имело бы одно и то же повторяющееся значение во всех строках. Уточним подписи полей- столбцов в заголовке формы, а также шрифт, размеры полей и подписей. После редактирования формы сохраним ее, нажав кнопку панели инструментов **Сохранить**. На рисунке 72 приведена подчиненная форма СПИСОК СТУДЕНТОВ после редактирования.

**Переход в режим формы и загрузка таблиц**

Завершив редактирование формы, перейдем из режима конструктора в *режим формы*, выбрав его на панели конструктора форм или выполнив команду меню **Вид - Режим формы**. На рисунке 73 показана окончательно отредактированная форма в режиме просмотра.



**Рисунок 73 -** Форма ввода-вывода для работы с данными двух взаимосвязанных таблиц

Если после редактирования форма была закрыта, то для начала сеанса работы с данными через форму необходимо в окне **Учебный процесс: база данных** (рисунок 68) в группе **Объекты** перейти к строке **Формы**, выделить имя созданной многотабличной формы СПИСОК ГРУППЫ и нажать кнопку **Открыть**.

Полученная многотабличная форма СПИСОК СТУДЕНТОВ ГРУППЫ обеспечивает одновременную загрузку и работу с данными таблиц ГРУППА и СТУДЕНТ. Загрузим эти таблицы данными в соответствии с Приложением А.

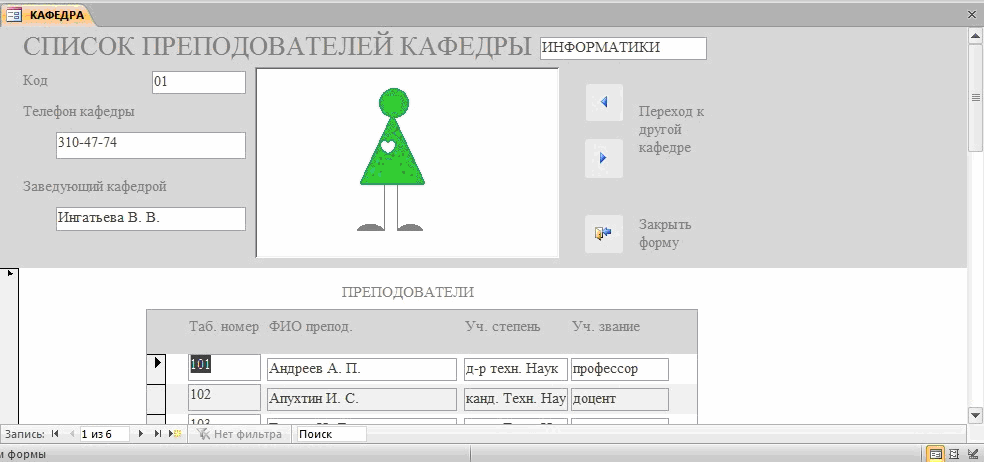
В процессе загрузки сначала вводятся значения реквизитов группы: номер группы, количество студентов и средний балл группы.

**Замечание**. Ввод ключевого поля НГ всегда обязателен для создания записи в таблице ГРУППА. Причем эта запись создается независимо от того, будет ли введен сразу список студентов группы. Другие два реквизита группы могут вводиться и позже при просмотре данных через форму, если не введены дополнительные ограничения на их значения в свойствах при конструировании таблицы.

Далее вводятся реквизиты студентов в область подчиненной формы. При этом ввод номера студента в группе всегда обязателен для образования записи в таблице СТУДЕНТ, который вместе с введенным в основную часть формы номер группы образует уникальный ключ в этой таблице. Запись о студенте сохраняется при переходе к очередной строке в подчиненной форме.

Для перехода к записи другой группы можно использовать созданные кнопки со стрелками вверх (вниз), для перемещения по записям студентов – стандартные кнопки перехода в *поле номера записи* в нижней части подчиненной формы. Для завершения работы с формой используется созданная в форме кнопка **ЗАКРЫТЬ** или стандартной кнопкой окна в Windows.

**Упражнение**. Для одновременной загрузки таблиц КАФЕДРА и ПРЕПОДАВАТЕЛЬ создайте форму, показанную на рисунке 74. При создании формы выполните действия, аналогичные рассмотренным в примере для таблиц ГРУППА и СТУДЕНТ.



**Рисунок 74 -** Форма для загрузки и работы с таблицами КАФЕДРА → ПРЕПОДАВАТЕЛЬ

1. В соответствии с технологией загрузки базы данных, рассмотренной в данной практической работе, осуществите проектирование формы для загрузки данных в таблицы КАФЕДРА и ПРЕПОДАВАТЕЛЬ из документа-источника «Список преподавателей кафедры»:
   * Определите подсхему данных для составной формы
   * В соответствии с подсхемой определите общую структуру составной формы

Спланируйте размещение реквизитов в макете формы так, чтобы обеспечить удобный ввод данных из документа «Список преподавателей кафедры».

1. Создайте форму средствами мастера форм и отредактируйте в конструкторе форм.
2. Загрузите данные через построенную форму в таблице КАФЕДРА и ПРЕПОДАВАТЕЛЬ.

**II часть**

**Технология разработки многотабличной формы для загрузки подчиненной таблицы**

В соответствии с этапами загрузки базы данных "Учебный процесс", определенными выше (см. табл. 4.1), загрузка записей о занятиях текущего семестра в таблицу ИЗУЧЕНИЕ должна выполняться после загрузки таблиц со справочными данными, что обеспечит установление связей загружаемых записей с соответствующими записями этих таблиц.

*На этапе проектирования* определим все необходимые требования к создаваемой форме и ее макет*.* Процесс конструирования сложной формы средствами Access осуществим далее в соответствии с результатами этой работы.

При определении требований к форме рассмотрим особенности назначения и работы с формой, составим подсхему данных для создания формы, определим общую структуру формы и размещение реквизитов в соответствии со структурой входного документа и подсхемой данных.

**Проектирование формы**

Выполним проектирование формы, обеспечивающей загрузку в таблицу ИЗУЧЕНИЕ данных о занятиях текущего семестра в каждой группе, и определим требования к форме, на основе которых можно перейти к ее созданию.

Документом внемашинной сферы, содержащим необходимые данные для загрузки таблицы ИЗУЧЕНИЕ, может служить "План проведения занятий в группе". Поэтому загружаемые через форму записи целесообразно группировать в соответствии с их подчиненностью записям другой таблицы - ГРУППА.

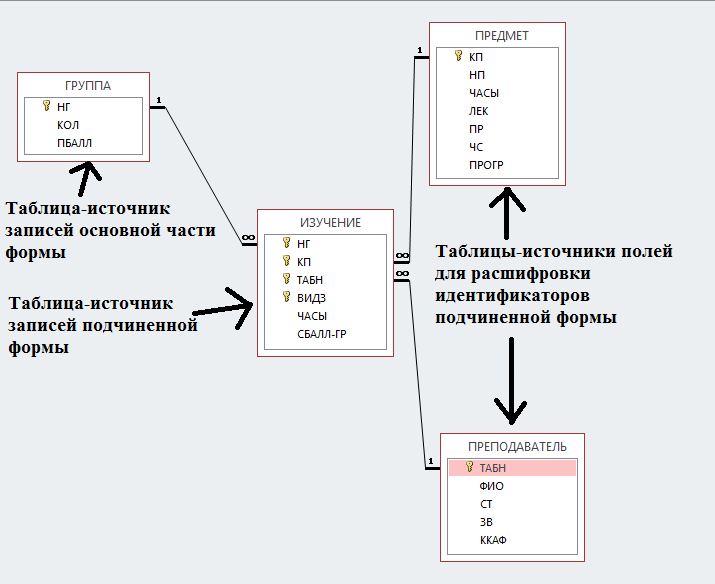
В таблицу ИЗУЧЕНИЕ в соответствии с ее структурой наряду с номером группы и видом занятия нужно вводить идентификаторы предмета и преподавателя. В документе-источнике указаны наименование предмета и фамилия преподавателя. Для того чтобы при загрузке правильно вводить только идентификаторы, предусмотрим отображение в форме расшифровывающей информации: *наименования предмета* (НП) и *фамилии преподавателя* (ФИО) из таблиц ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, которым подчинена загружаемая таблица.

Поскольку форма служит не только для загрузки, но и для просмотра, включим в форму и другие описательные реквизиты из таблиц ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ для их отображения.

**Определение подсхемы данных**

В результате загрузки данных о занятиях в группах должны формироваться только записи таблицы ИЗУЧЕНИЕ *(объект загрузки),* которую надо включить в подсхему данных для формы ввода-вывода данных о занятиях в группах (рисунок 75).

Поскольку загрузку и просмотр данных удобно производить по каждой группе в отдельности, в подсхему включим таблицу ГРУППА. Выберем по этой причине таблицу ГРУППА в качестве *источника основной части составной формы*. Заметим, что данные таблицы ГРУППА должны только отображаться в форме, но не вводить в нее. В подсхему (рисунок 75) включим также таблицы ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, т.к. в форме необходимо предусмотреть отображение описательных данных о предмете и преподавателе.

**Рисунок 75 -** Подсхема для формы ввода-вывода данных в таблицу ИЗУЧЕНИЕ

**Определение общей структуры формы**

На основе полученной подсхемы (рисунок 75) определим общую структуру формы, которую назовем ПЛАН ЗАНЯТИЙ.

**Основная часть формы**. Многотабличная форма ПЛАН ЗАНЯТИЙ будет содержать основную часть на основе таблицы ГРУППА для группировки вводимых данных о занятиях по каждой группе студентов. Для ввода данных в таблицу ИЗУЧЕНИЕ предусмотрим *непосредственное включение подчиненной формы* ИЗУЧЕНИЕ.

Таким образом, форму ПЛАН ЗАНЯТИЙ определяют:

* *Тип формы* – многотабличная
* *Источник записей* для основной части формы – таблица ГРУППА
* Включаемая подчиненная форма ИЗУЧЕНИЕ с *источником записей* - таблица ИЗУЧЕНИЕ

**Подчиненная форма, включаемая в основную**. Для расшифровки идентификаторов предмета и преподавателя предусмотрим включение в подчиненную форму ИЗУЧЕНИЕ полей из таблиц ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, являющихся главными относительно таблицы, на которой строится форма ИЗУЧЕНИЕ.

Таким образом, подчиненную форму ИЗУЧЕНИЕ определяют:

* *Тип формы – подчиненная*
* *Источник записей для основной части формы ИЗУЧЕНИЕ – таблица ИЗУЧЕНИЕ*
* *Источники отображаемых полей формы – таблицы ПРЕДМЕТ и*

*ПРЕПОДАВАТЕЛЕЙ*

**Размещение реквизитов формы**

**Основная часть формы.** В основной части составной формы ПЛАН ЗАНЯТИЙ вверху будем размещать поля таблицы ГРУППА:

* *НГ – номер группы (уникальный ключ)*
* *КОЛ – количество студентов*
* *ПБАЛЛ – средний балл в группе при поступлении*

Доступ к перечисленным полям должен быть ограничен *только чтением*, т.к. значения этих полей не должны вводиться и корректироваться из документа

«План занятий».

**Подчиненная форма ИЗУЧЕНИЕ.** В подчиненной форме ИЗУЧЕНИЕ разместим:

Все поля загружаемой таблицы ИЗУЧЕНИЕ, кроме ключевого реквизита НГ (номер группы), включенного в основную часть формы, что обеспечивает однократное отображение одинаковых номеров группы по форме :

* *КП – код предмета*
* *ТАБН – идентификатор преподавателя*
* *ВИДЗ – вид занятий*
* *ЧАСЫ – число часов занятий*
* *СБАЛЛ-ГР – средний балл по предмету в груп*пе

Поля из таблиц ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, позволяющие отобразить

справочную информацию о предмете и преподавателе, ведущем занятие: поля таблицы ПРЕДМЕТ:

* *НП – название предмета*
* *ЧАСЫ – всего часов*
* *ЛЕК – часов лекций*
* *ПР – часов практики*

Поля таблицы ПРЕПОДАВАТЕЛЬ:

* *ФИО – фамилия преподавателя*
* *СТ – ученая степень*
* *УЗ – ученое звание*

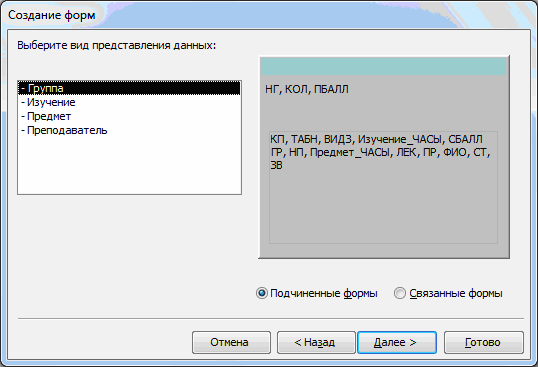
**Создание многотабличной формы с подчиненной формой средствами мастера**

Используя мастера форм, создадим в соответствии с результатами проектирования составную форму ПЛАН ЗАНЯТИЙ, включающую подчиненную форму ИЗУЧЕНИЕ.

В окне базы данных выберем вкладку **Создание,** в группе **Формы** нажмем кнопку **Мастер форм**. В качестве таблицы, которая будет источником записей основной части сложной формы, выберем таблицу ГРУППА.

В первом сеансе диалоговом окна **Создание форм** выберем включаемые в

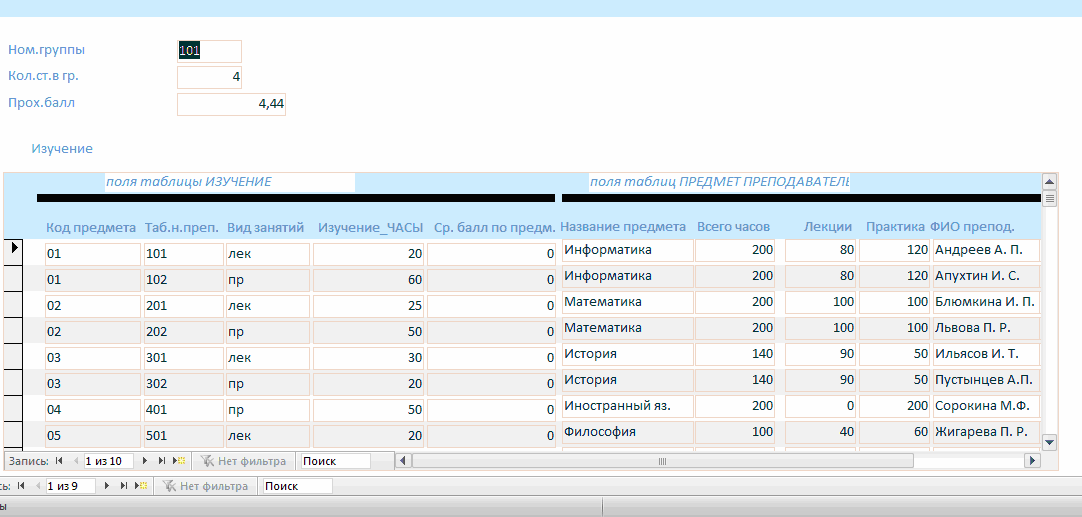
форму поля таблицы ГРУППА, а затем поля таблиц ИЗУЧЕНИЕ, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ. В следующем окне (рисунок 76) уже будет выделена таблица ГРУППА, которая выбрана для создания основной части формы. Поскольку эта таблица является главной в схеме данных относительно используемой в форме таблицы ИЗУЧЕНИЕ, представляется возможность выбрать тип формы – Подчиненные формы.



**Рисунок 76 -** Окно мастера форм при выборе типа включения подчиненной формы

Для получения многозаписевой подчиненной формы выберем в следующем сеансе окна мастера вид формы **Ленточный**  и стиль оформления **Стандартный**. В последнем сеансе окна **Создание форм** зададим имя составной формы – ПЛАН ЗАНЯТИЙ, а также имя подчиненной формы – ИЗУЧЕНИЕ. Выберем также дальнейшие действия мастера – **Открыть форму для просмотра и ввода данных**.

По завершению работы мастера выводится форма с данными из таблиц, которые были заданы пользователем в диалоге с мастером. В подчиненной форме выводятся все записи о занятиях для одной группы (рисунок 77). При этом запись о занятии содержит, кроме полей из таблицы ИЗУЧЕНИЕ, также поля из главных по отношению к ней таблиц ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ. Мастер, проанализировав схему данных, сам формирует подчиненную форму ИЗУЧЕНИЕ, и включает поля этих таблиц в качестве расшифровывающих.



**Рисунок 77 -** Форма с подчиненной формой, созданная мастером на основе

четырех таблиц

Заметим, что запись в подчиненной форме имеет очень много полей, причем лишь часть этих полей служит для загрузки данных в подчиненную таблицу ИЗУЧЕНИЕ. Другие поля должны отображаться из ранее загруженных таблиц ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ. Такое размещение полей неудобно для работы пользователя. Форму целесообразно доработать средствами конструктора.

**Разработка формы, обеспечивающей удобный интерфейс пользователя для загрузки подчиненной таблицы**

Рассмотренные выше два варианта построения формы для загрузки данных о занятиях недостаточно удобны для работы пользователя. В первом варианте данные о каждом занятии были представлены компактно в виде *одиночной записи,* но не обеспечивался удобный доступ к занятиям каждой группы. Вариант формы с подчиненной таблицей также имел недостатки, поскольку данные, вводимые в таблицу ИЗУЧЕНИЕ, входили в одну длинную запись вместе с отображаемыми справочными данными.

Выполним разработку формы, более удобной пользователю для просмотра занятий группы и загрузки новых занятий, и защитим в ней поля справочных данных.

**Требования к создаваемой форме.** Обеспечим возможность компактного

отображения на экране всей информации об одном занятии и сохраним в то же время возможность объединения записей о занятиях по группам, как в форме на рисунке 77. Кроме того, целесообразно в форме в одну группу объединить поля, в которые вводятся значения при загрузке таблицы ИЗУЧЕНИЕ, а в другую - поля, которые содержат только справочную информацию о предмете и преподавателе, которая отображается для расшифровки идентификаторов занятия.

В процессе конструирования обеспечим защиту справочных данных в таблицах ГРУППА, ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ от случайных изменений при загрузке данных о занятиях в таблицу ИЗУЧЕНИЕ. Предусмотрим для удобства пользователя кнопки перехода к просмотру занятий для другой группы и кнопку закрытия формы. Для визуальной проверки правильности вводимых идентификаторов преподавателя и предмета используем поля со списком.

В соответствии с перечисленными требованиями для первоначального размещения полей и создания подчиненной формы можно воспользоваться формой ПЛАН ЗАНЯТИЙ, полученной мастером (рисунок 77). Откроем эту форму в режиме конструктора. Для этого в **Границе области переходов (область навигации)** выберем для редактирования созданную ранее многотабличную форму ПЛАН ЗАНЯТИЙ. Если форма была открыта ранее в режиме просмотра или в режиме макета, то для перехода в режим конструктора достаточно выбрать соответствующий тип представления в группе **Режимы** на вкладке **Главная.**

**Редактирование основной части формы.** В основной части формы разместим и отредактируем поля таблицы ГРУППА так, как это показано на рисунке 78. Уточним текст подписей полей, шрифт и размеры полей и подписей, введем текст в заголовок формы. Удалим элемент с подписью подчиненной формы. Удалим разделительные линии между разделами формы: заголовком, областью данных и примечания. Для этого в свойствах формы на вкладке **Макет** в строке **Разделительные липни** выберем **Нет**. Уберем область выделения записи, проставив в свойствах формы в соответствующей строке "Нет" Создадим две кнопки для перехода к следующей или предыдущей группе, а также кнопку для закрытия формы.



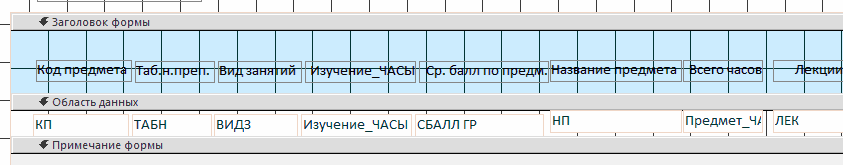
**Рисунок 78 -** Основная часть многотабличной формы после редактирования в режиме конструктора

**Ограничение доступа к полям таблицы-источника основной части формы.** Защитим данные записей таблицы ГРУППА от непроизвольных изменений при работе с формой, т.к. они должны использоваться только для отображения. Это все поля основной части формы. Для защиты поля выделим рамку поля и с помощью контекстно-зависимого меню вызовем свойства поля. В окне свойств на вкладке **Данные** в строке **Блокировка** выберем **Да**. После установки этого свойства поле доступно только для чтения.

Сохраним форму под новым именем – «ПЛАН ЗАНЯТИЙ В ГРУППЕ», используя команду **Сохранить как**.

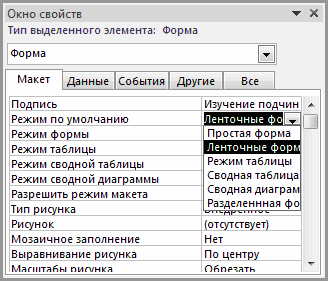
**Редактирование подчиненной формы ИЗУЧЕНИЕ**.

Ранее мастером было получена подчиненная ленточная форма, которая в режиме конструктора показана на рисунке 79.



**Рисунок 79 -** Подчиненная форма ИЗУЧЕНИЕ в режиме конструктора

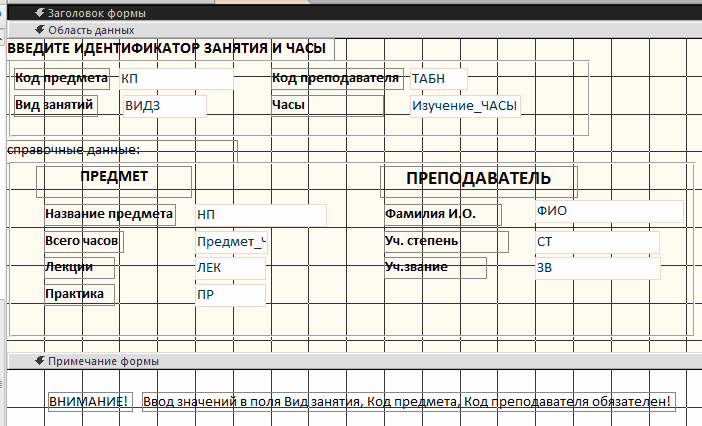
Для изменения вида подчиненной формы вызовем ее свойства. На вкладке **Макет** в окне **Окно свойств** (прав.кнопка мыши - пункт «свойства формы») (рисунок 80) заменим в строке **Режим по умолчанию** значение **Ленточная форма** на **Простая форма**. Это позволит отображать в подчиненной форме одну запись о занятии. Вид формы в конструкторе останется прежним.



**Рисунок 79 -** Отображение списка режимов по умолчанию в окне свойств подчиненной формы

В подчиненной форме разместим поля так, как это показано на рисунке 81. После перемещения всех подписей полей из заголовка в область данных можно сократить его размер до нуля перемещением границы заголовка и области данных. Поля таблицы ИЗУЧЕНИЕ, в которой надо вводить данные из документа

«План занятий», разместим в верхней части области данных. В нижней части области данных разместим поля, в которые будут автоматически выводиться справочные данные из таблиц ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ для расшифровки вводимых идентификаторов занятия. Эти поля служат только для отображения сведений о предмете и преподавателе. Для создания рамок используем кнопку панели элементов управления **Прямоугольник** .



**Рисунок 81 -** Подчиненная форма ИЗУЧЕНИЕ в режиме конструктора форм после редактирования

Уточним текст подписей полей, шрифт и размеры полей и подписей. Выполним относительное выравнивание надписей и полей с помощью команды **Главная - Форматирование текста - Выровнять.**

В область примечаний формы введем инструкцию пользователю, требующую обязательного ввода данных в поля, идентифицирующие занятие: код предмета — КП, номер преподавателя — ТАБН и вид занятия — ВИДЗ. Без этого не может быть создана запись в таблице ИЗУЧЕНИЕ.

**Защита справочных данных от изменений.** Защитим поля НП, ЧАСЫ, ЛЕК, таблицы ПРЕДМЕТ и поля ФИО, СТ, 3В таблицы ПРЕПОДАВАТЕЛЬ от случайных изменений при работе с формой. Для зашиты поля выделим рамку поля и с помощью контекстно-зависимого меню вызовем свойства по- ля. В окне свойств на вкладке **Данные** в строке **Блокировка** выберем **Да**. После установки этого свойства поле доступно только для чтения.

Для визуального контроля правильности ввода идентификаторов занятия: КП и ТАБН можно использовать **Поле со списком**. Процесс создания такого поля рассматривается ниже.

**Создание полей со списком**

При вводе идентификационных данных через форму в Access имеется возможность получить справочную информацию из ранее загруженных таблиц, что позволяет выбрать уже имеющиеся значения в базе и тем самым повысить достоверность вводимой информации. Отображение данных из справочных таблиц при вводе идентификатора свидетельствует о наличии в базе данных главных записей для загружаемой подчиненной записи, что необходимо для успешного завершения ввода при установленном параметре целостности в схеме данных.

Поле со списком объединяет поле формы, в которое нужно ввести данные, и список. Список содержит записи из связанной главной таблицы. В списке можно выбрать из соответствующего поля нужное значение и ввести его в поле формы (пример на рисунке 82).

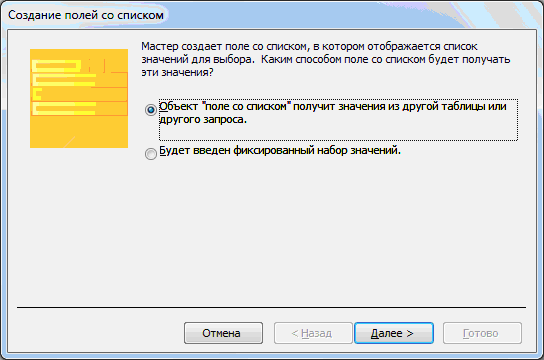


**Рисунок 82 -** Отображение списка

**Создание поля со списком с помощью мастера**

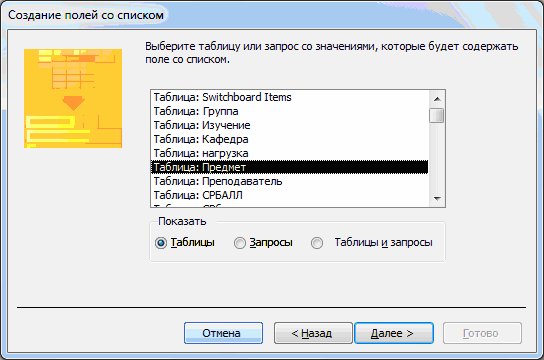
Создадим поле со списком для ввода значений кода предмета – КП в таблицу ИЗУЧЕНИЕ. Это позволит просматривать и вводить значения, которые уже имеются в главной таблице ПРЕДМЕТ, а также проверять соответствие кода и наименования предмета, имеющихся в документе-источнике загрузки «План занятий».

Выберем кнопку **Поле со списком** в группе **Элементы управления** на вкладке **Конструктор,** установим курсор мыши в нужное место, нажмем кнопку мыши и, не отпуская ее, вычертим рамку элемента. После отпускания кнопки мыши запустится мастер и откроет диалоговое окно **Создание полей со списком.** В этом окне определим способ, которым список поля получает свои значения. Для формирования списка из связанной таблицы выберем вариант **Таблица или запрос содержат значения, которые использует поле со списком** (рисунок 83).



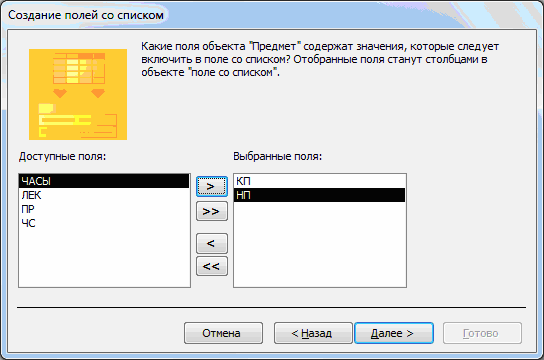
**Рисунок 83 -** Выбор способа форматирования списка

В следующем окне выберем таблицу ПРЕДМЕТ, которая будет поставлять значения в список поля (рисунок 84).



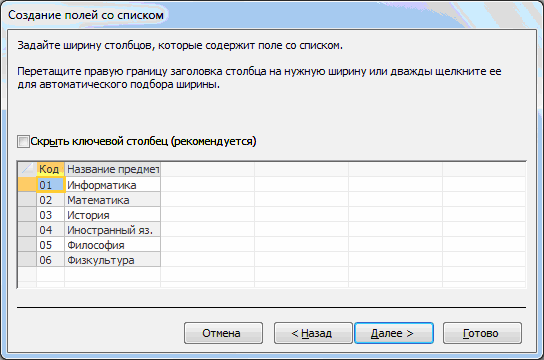
**Рисунок 84 -** Выбор источника данных для формирования списка

Затем вберем поле КП, а также поле НП для расшифровки кода КП (рисунок 85). Эти поля образуют записи списка.



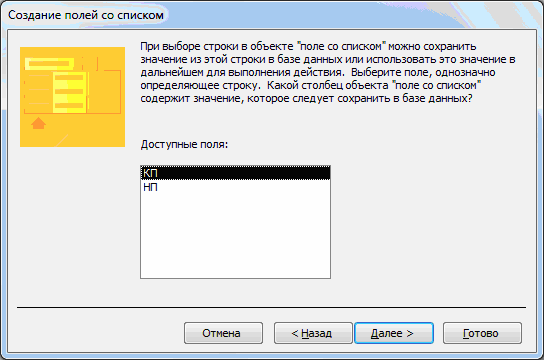
**Рисунок 85 -** Выбор полей для формирования записи списка

Далее в появившейся таблице определим ширину столбцов списка в соответствии с размером значений. Для этого курсор мыши установим на линию, разделяющую имена столбцов, и переместим ее в нужное место (рисунок 86).



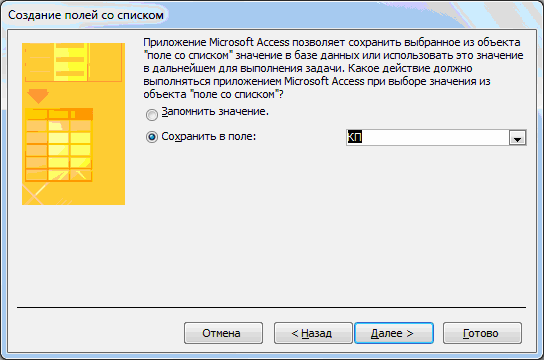
**Рисунок 86 -** Окно для оформления столбцов списка

Далее выберем поле списка КП, являющееся ключом связанной таблицы ПРЕДМЕТ. Из этого поля будет выбираться значение для ввода в поле фор мы (рисунок 87).



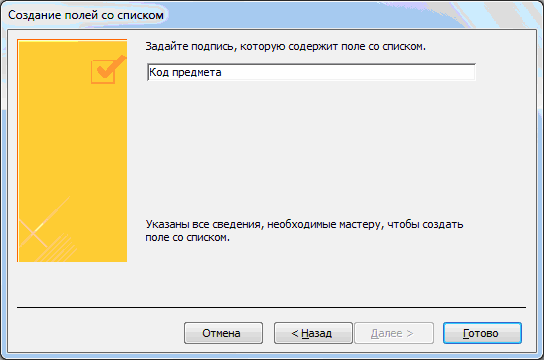
**Рисунок 87 -** Окно для выбора поля источника значений для поля формы

В следующем окне отметим переключатель **Сохранить в поле** и выберем поле формы КП (поле таблицы ИЗУЧЕНИЕ ), в кото-рое будут вводиться значения из списка (рисунок 88)



**Рисунок 88 -** Выбор поля формы, в которое вводится значение из списка

Далее введем подпись поля со списком – «Код предмета» (рисунок 89).



**Рисунок 89 -** Определение подписи поля со списком

Нажмем кнопку **Готово**. В результате получим поле КП со списком, которое содержится в окончательной форме, представленной на рисунке 92. Использование поля со списком предметов возможно в режиме формы. Для удобства поиска нужного значения в списке можно воспользоваться операциями поиска и сортировки. Доступ к этим операциям возможен при помощи кнопок **Найти (**группа **Найти**), **Сортировка по возрастанию**, **Сортировка по убыванию** группы **Сортировка и фильтр**.

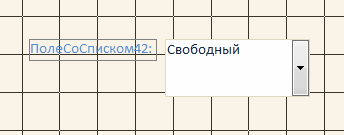
**Создание поля со списком без использования мастера**

Создадим поле со списком для ввода значений идентификатора преподавателя ТАБН. Это позволит просматривать и вводить значения, которые уже имеются в главной таблице ПРЕПОДАВАТЕЛЬ, а также проверять соответствие номера и фамилии преподавателя, имеющихся в документе-источнике загрузки.

Нажмем на вкладке **Конструктор** в группе **Элементы управления** кнопку

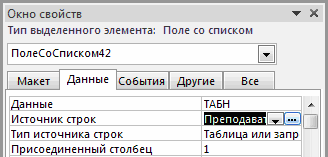
**Поле со списком**.

Установим курсор мыши на появившийся элемент **Свободный** (рисунок 90).



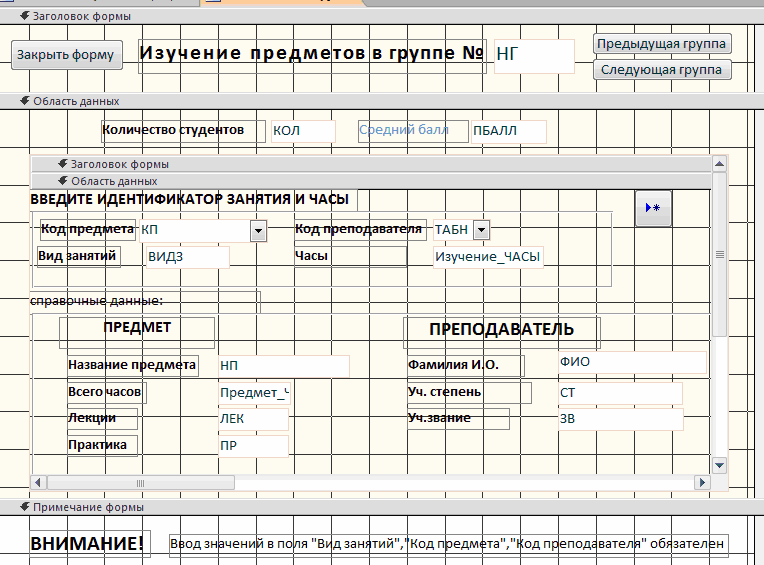
**Рисунок 90 -** Элемент формы **ПолеСоСписком**

Нажмем правую кнопку мыши, чтобы вызвать контекстно зависимое меню. Выберем пункт **Свойства**, затем – вкладку **Данные**. В строке **Данные** выберем поле ТАБН, которое необходимо заполнять через форму в таблице ИЗУЧЕНИЕ (рисунок 91). В строке **Тип источника строк** выберем элемент **Таблица - Запрос**, а в строке **Источник строк** – таблицу ПРЕПОДАВАТЕЛЬ. Поля, включаемые в список, и их порядок определяются в построителе, который вызывается в этой же строке нажатием кнопки .



**Рисунок 91 -** Окно определения свойств данных поля со списком

Построитель выводит бланк запросов, в который перетащим из таблицы ПРЕПОДАВАТЕЛЬ поля ТАБН и ФИО.

Для того чтобы в списке выводились два поля, на вкладке **Макет** необходимо в строке **Число столбцов** указать «2». Для настройки ширины столбцов списка в строках **Ширина списка** и **Ширина столбцов** зададим подходящие значения. Закроем окно свойств **Поле со списком**. В результате получим поле ТАБН со списком, которое содержится в окончательной форме, представленной на рисунке 92.

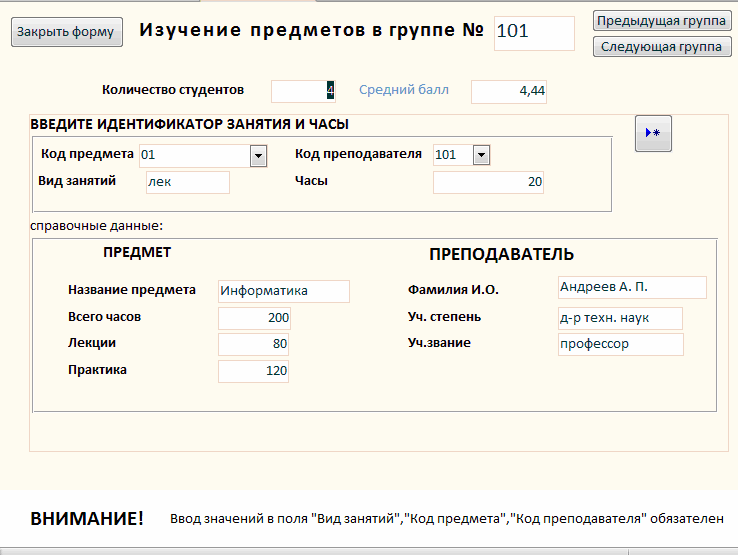
**Рисунок 92 -** Форма ПЛАН ЗАНЯТИЙ в режиме конструктора

**Замечание** Если необходимо преобразовать обычное поле в поле со списком , надо в контекстно-зависимом меню поля выбрать **Преобразовать элемент в** и далее выбрать строку **Поле со списком** При включенной кнопке **Мастера элементов** элемент будет преобразован мастером. При выключенной кнопке необходимо для получения поля со списком установить свойства этого поля, как описано выше.

**Загрузка данных в подчиненную таблицу через форму**

Для загрузки данных в подчиненную таблицу ИЗУЧЕНИЕ через форму ПЛАН ЗАНЯТИЙ можно сразу перейти из режима конструктора в **Режим формы.**

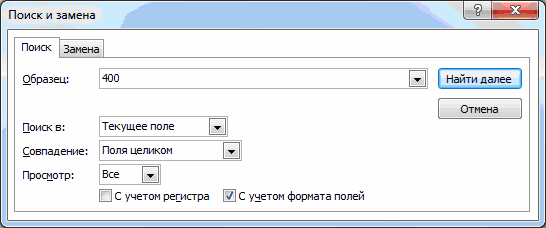
Для этого на вкладке **Главная** в списке кнопки **Режим** выбирается **Режим формы**. Если форма была закрыта, необходимо в окне **Область навигации** выбрать форму ПЛАН ЗАНЯТИЙ. На рисунке 93 приводится форма ПЛАН ЗАНЯТИЙ в режиме формы, обеспечивающая загрузку и корректировку данных о занятиях в таблице ИЗУЧЕНИЕ, подчиненной таблицам ГРУППА, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ, данные из которых только отображаются в форме.



**Рисунок 93 -** Форма ПЛАН ЗАНЯТИЙ в режиме формы

**Загрузка подчиненной таблицы.** Для загрузки записи нового занятия в таблицу ИЗУЧЕНИЕ через вызванную форму необходимо сделать текущим номер группы, для которой вводятся данные о занятиях из документа «План занятий». Это можно сделать путем просмотра записей групп при помощи кнопок **Предыдущая группа** и **Следующая группа** (рисунок 93).

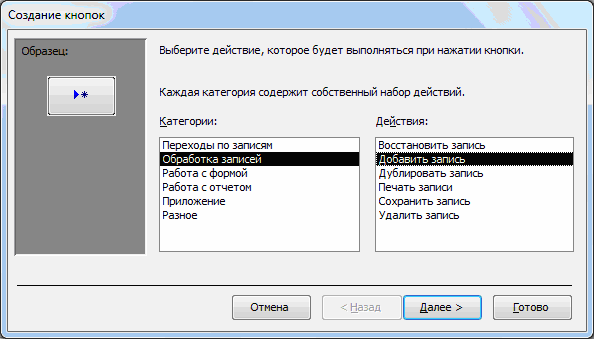
При большом числе групп целесообразно использовать функцию поиска нужной записи. Чтобы осуществить поиск, установим курсор в поле номера группы, нажмем на вкладке **Главная** в группе **Найти** кнопку **Найти** и в открывшемся окне **Поиск и замена** (рисунок 94) зададим в качестве образца поиска номер нужной группы. После нажатия в этом окне кнопки **Найти далее** в форме отобразятся данные занятиях заданной группы.



**Рисунок 94 -** Диалоговое окно поиска записи

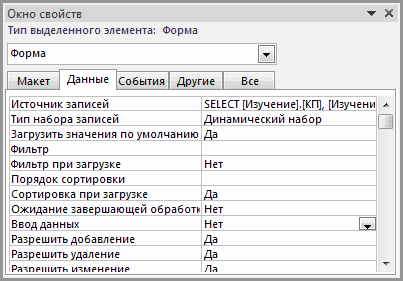
Для ввода новых записей должен быть обеспечен переход в режим добавления новой записи в подчиненной форме. Для этого может быть создана специальная кнопка перехода к пустой записи **Добавить запись** (рисунок 93).

Для формирования такой кнопки используем мастер кнопок, в диалоговом окне которого **Создание кнопок** (рисунок 95) надо выбрать соответствующую категорию **Обработка записей** и действие **Добавить запись**, которые обеспечат формирование нужной процедуры обработки события.



**Рисунок 95 -** Выбор параметров при создании кнопки для добавления за- писи

Установка свойств **Разрешить добавление**, **Разрешить удаление, Разрешить изменение**, обеспечивающих возможность добавления, удаления и изменения записей при загрузке и корректировке записей таблицы ИЗУЧЕНИЕ – источника записей, показана в окне свойств подчиненной формы (рисунок 96).

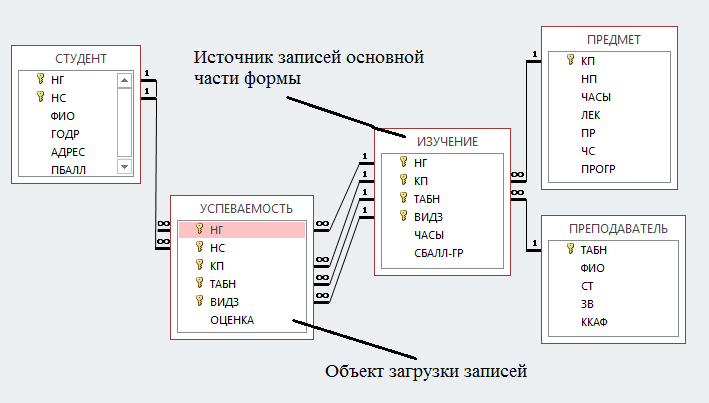


**Рисунок 96 -** Установка свойств подчиненной формы ИЗУЧЕНИЕ

**Упражнение.** Создайте многотабличную форму, для загрузки результатов сдачи экзаменов в таблицу УСПЕВАЕМОСТЬ и их просмотра. При создании формы произведите действия, аналогичные рассмотренным для таблицы ИЗУЧЕНИЕ.

В соответствии с технологией загрузки базы данных, рассмотренной в начале настоящей главы, осуществите проектирование формы для загрузки данных в подчиненную таблицу УСПЕВАЕМОСТЬ из документа-источника «Экзаменационная ведомость».

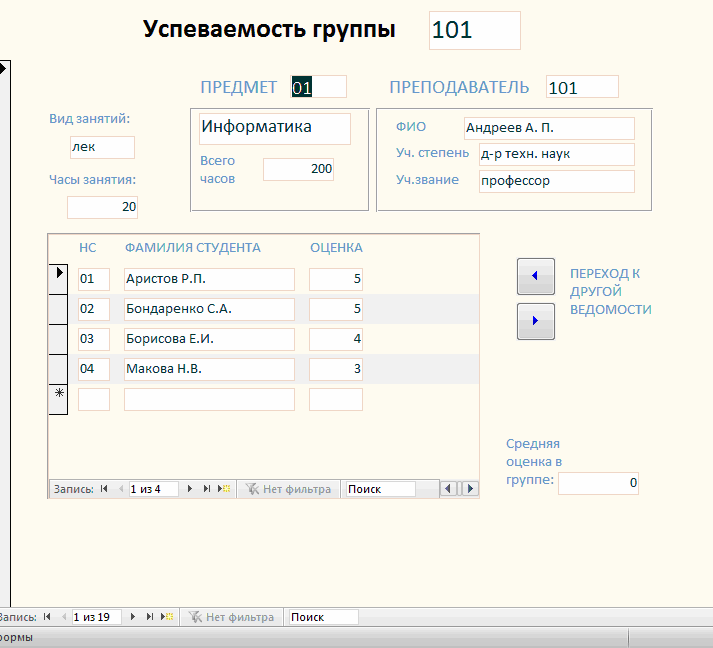
Определите общую структуру составной формы для ввода (просмотра) данных об оценках студентов группы по предмету в соответствии с подсхемой данных для составной формы, приведенной на рисунке 97.

**Рисунок 97 -** Подсхема для составной формы ВЕДОМОСТЬ УСПЕВАЕМОСТИ

В результате загрузки в БД данных об оценках студентов группы по предмету в БД должны формироваться только записи таблицы УСПЕВАЕМОСТЬ (объект загрузки). Загрузку и просмотр этих данных удобно производить по каждому проведенному в группе занятию в отдельности из соответствующей «Экзаменационной ведомости». Поэтому в подсхему для формы ввода включается таблица ИЗУЧЕНИЕ, которую по этой причине целесообразно выбрать в качестве источника основной части составной формы. Кроме того, в форме предусмотрен вывод (отображение) данных о предмете и преподавателе, проводящем занятие, а в списке студентов необходимо предусмотреть отображение его фамилии. Для этого в подсхему включены таблицы СТУДЕНТ, ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ.

Спланируйте размещение реквизитов в макете формы так, чтобы обеспечить удобный ввод данных в таблицу УСПЕВАЕМОСТЬ из документа «Экзаменационная ведомость», а также отображение справочной информации о студенте, предмете и преподавателе, в списке студентов предусмотрите вывод фамилии студента.

Выполните конструирование экранной формы (рисунок 98), через которую будет осуществляться ввод, добавление и изменение записей таблицы базы данных УСПЕВАЕМОСТЬ.



**Рисунок 98 -** Форма ввода и просмотра данных об успеваемости студентов по теории и практике

Загрузите через построенную форму данные из документа «Экзаменационная ведомость» в таблицу УСПЕВАЕМОСТЬ.

# Практическая работа №7

# Создание однотабличного отчета в режиме конструктора

**Цель работы:** научить студентов создавать отчет для одной таблицы в режиме конструктора, группировать и сортировать данные отчета, размещать данные в разделах отчета, оформлять отчет, просматривать отчет и отправлять его на печать.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

**Создание отчета для одной таблицы**

Рассмотрим технологию создания однотабличного отчета на примере получения списков студентов по группам. Пусть в результате проектирования макета отчета списки СТУДЕНТОВ определены перечисленные ниже требования к отчету.

* Вид макета формируемого отчета должен соответствовать приведенному на рис. 1. На макете показано оформление списка студентов для одной группы. В отчете должны последовательно, со своими заголовками, отображаться списки студентов для каждой группы. Вывод записей списка группы должен осуществляться в порядке возрастания номера студента в группе.
* При формировании отчета необходимо рассчитать средний проходной балл для каждой группы и отобразить его в отчете
* Вывод названия отчета должен производиться на каждой странице отчета



Рис. 1. Проект макета для создания однотабличного отчета со списками студентов по группам

**Практическая часть**

**Создание однотабличного отчета в режиме конструктора**

Выберите в окне базы данных объект **Отчеты** (Repoгts). Начать конструирование отчета можно разными способами. Можно сразу перейти в режим конструирования, выбрав в окне базы данных строку **Создание отчета в режиме конструктора** (Create гepoгt in Design view). В этом случае сразу открывается не связанный с источником отчет.

Можно начать конструирование, нажав кнопку**Создать** (New). В открывающемся диалоговом окне (рис. 2) **Новый отчет** (New Repoгt), позволяющем выбирать разные режимы, в данном случае необходимо выбрать вари­ант **Конструктор** (Design). Если в этом окне в качестве источника данных отчета выбрать таблицу (запрос), то одновременно с открытием нового отчета в режиме конструктора откроется дополнительное окно со списком полей источника.

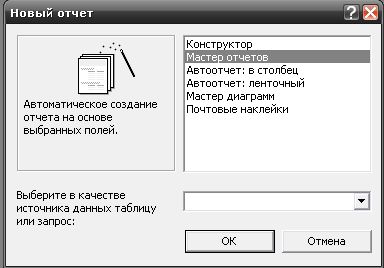


Рис. 2. Окно выбора варианта создания отчета, таблицы-источника и окно доступных полей

Выберем в качестве источника для рассматриваемого примера та6лицу СТУДЕНТ. Появится окно со списком доступных полей источника отчета (рис. 2).

В открывшемся окне конструктора отображается макет создаваемого отчета. Если раздел **Заголовок отчета** (Report Header) отсутствует, включим его с помощью команды меню **Вид | Заголовок / примечание отчета** (View I Report Header /Footer) или соответствующей кнопки панели инструментов конструктора отчетов.

Для удобства работы можно на панели конструктора добавить кнопку **заголовок | примечание отчета** (Report Header l Footer), при использовании которой добавляются соответствующие разделы.

**Группировка и сортировка данных отчета**

Для выполнения требований к группировке и сортировке отображаемых в отчете данных можно использовать кнопку **Сортировка и группировка** (Sоrtiпg апd Grouping) на панели инструментов конструктора. Далее задаются необходимые параметры в открывающемся диалоговом окне **Сортировка и группировка** (Sorting апd Grouping). Для рассматриваемого примера в этом окне необходимо задать группировку записей по полю нг и сортировку по полю нс (рис. 3).

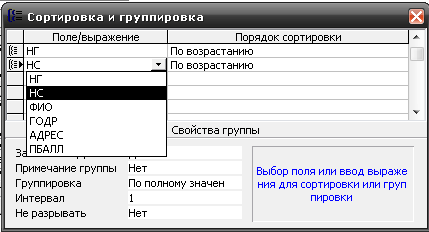


Рис. З. Окно определения групп записей по полю НГ и сортировки по полю НС

**Группировка по полю**

В окне **Сортировка и группировка** (Sorting and Grouping) в столбце **Поле/выражение** (Field/Competition) необходимо записать имя поля для группировки или выбрать это поле из раскрывающегося списка полей источника. В области **Свойcтвa группы** (Group Properties) в строках **Заголовок группы** (Group Header) и **Примечание группы** (GroupFooter) следует выбратьзначение Да (Yes).

Поскольку общий список студентов в соответствии с проектом отчета должен быть разбит по группам, выберем для группировки поле НГ рассматриваемого примера.

После определения группировки в окне конструктора отчета (рис. 4) появляются дополнительные разделы **Заголовок группы НГ(НГ Header), Примечание группы НГ (НГ Footer).**

**Сортировка по полю**

Для задания сортировки в следующей строке диалогового окна достаточно записать (выбрать) имясоответствующего поля. В области **Свойства группы** этого поля в строках **Заголовок группы НР(НГ Healler)** и **Примечание группы НГ (НГ Footer)** по умолчанию установлены значения Нет (No), что и устанавливает по этому полю только сортировку.

В рассматриваемом при мере для вывода отсортированного списка студентов в каждой группе зададим сортировку по полю номера студента НС. Для этого в диалоговом окне выберем во второй строке поле НС.

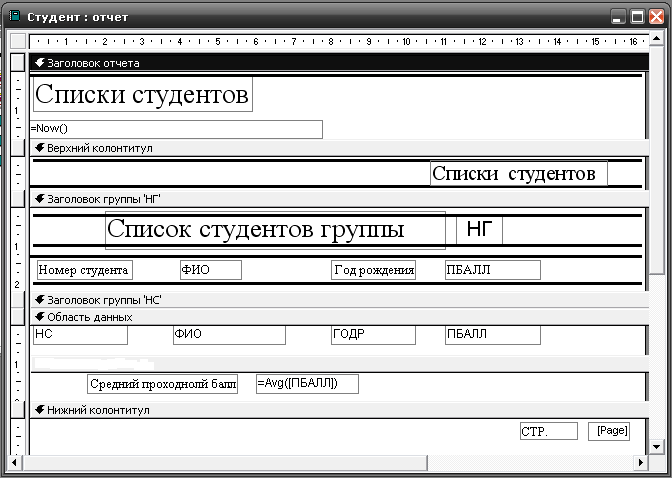


Рис. 4. Окно конструктора отчетов при разработке отчета для вывода данных из одной таблицы СТУДЕНТ

**Размещение данных в разделах отчета**

Данные в отчете должны быть размещены в соответствии с проектом макета отчета. Для рассматриваемого примера проект отчета приведен выше (см. рис. 1).

**Размещение полей из таблиц**

Процедура размещения полей из таблиц включает два этапа.

1. Размещение поля группировки.

Значение поля группировки должно быть представлено один раз в области заголовка группы. В примере отчета поле НГ должно быть размещено в разделе **Заголовок группы 'НГ'** (НГ Header). Если список полей источника не открыт, нажмем кнопку панели инструментов конструктора отчетов **Список полей** (Field List) Перетащим с помощью мыши поле НГ из списка полей в раздел **Заголовок группы**. Откорректируем подпись поля, изменив ее на **Список студентов группы**. Установим нужный шрифт для элементов. Для установки размеров рамки по размеру текста подписи выполним команду **Формат | Размер | По размеру данных** (Forrnat | Size | Size То Fit) или нажмем соответствующую кнопку панели инструментов.

2. Формирование табличной части отчета.

В соответствии с проектом макета необходимо разместить поля для табличной части так, чтобы их имена располагались в области данных, а подписи полей для формирования заголовка столбца - в области **Заголо­вок группы** (Group Header). Тогда в области данных после формирования отчета будут размещены строки табличной части, а в области заголовка - названия столбцов. При этом следует иметь в виду, что после перетаскивания полей из списка источника в область данных они поместятся туда вместе с подписями. Для

перемещения подписей в область **Заголовок группы** их можно вырезать и, сделав текущей область заголовка, , вставить. Заметим, что подписи также можно создать заново, Воспользовавшись кнопкой панели элементов **Надпись (**Label).

В примере формируемого отчета последовательно разместим поля НС, ФИО, ДАТАР, ПБАЛЛ в области данных. Поле размещается вместе с подписью, которую система берет из свойств полей таблицы СТУДЕНТ. Подписи полей следует переместить в область **Заголовок группы** путем вырезания и вставки. Если они не совпадают с названиями столбцов в проекте макета, их необходимо откорректировать.

**Включение вычисляемого поля для расчета итоговых значений**

Для создания вычисляемого поля с итоговым значением по группе записей разместим его в раздел **Примечание группы 'НГ'** (НГ Footer) (см. рис. 4). Для включения расчетного реквизита используется кнопка **Поле** (Техt Вох) на панели элементов, которая позволит сформировать элемент **Свободный** (Unbound).

Далее нужно открыть свойства создаваемого элемента и на вкладке **Данные** (Data) ввести в строку **Данные** (Control Source) необходимое выражение для выполнения групповой операции (суммирование, расчет среднего арифме­тического и т. п.). При необходимости, следует выбрать в раскрывающемся списке **Число десятичных знаков**(DecimaIPlaces) нужную точность расчета. На вкладке Макет (Format) в раскрывающемся списке **Формат поля**(Format) выбирается требуемый формат.

В примере формируемого отчета для формирования расчетного реквизита Средний проходной балл группы нажмем кнопку **Поле** (Техt Вох) и разместим элемент **Свободный** (Unbound) в раздел **Примечание группы 'НГ'** (НГ Footer) (см. рис. 4).

Откроем окно свойств элемента **Свободный** (Unbound). Запишем на вкладке **Данные** (Data) в строку **Данные** (Control Source) функцию = Avg ( [ПБАЛЛ] ) для расчета среднего арифметического значений поля ПБAЛЛ по записям студентов одной группы.для записи функции можно воспользоваться построителем выражений, вызвав его в окне свойств поля (рис. 5) кнопкой, расположенной справа от строки **Данные** (Control Source).

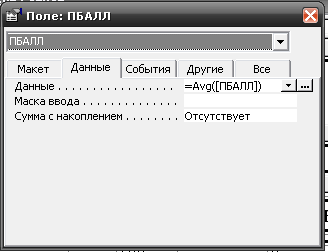


Рис. 5. Окно свойств поля для вычисления

В окне построителя (рис. 6) для формирования функции откроем папку **Функции**, выберем в ней папку **Встроенныефункции**. Затем в средней панели выберем из списка значение **Статистические**. В правой панели выберем нужную функцию Avg.

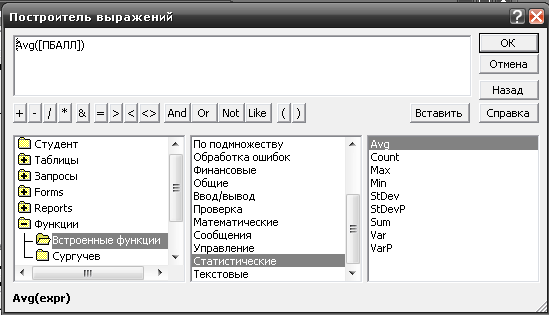


Рис. 6. Окно построителя выражений

В окне свойств поля на вкладке **Макет** (Format) в строке **Число десятичных знаков** (DecimaIPlaces) запишем значение 2. В раскрывающемся списке **Формат поля** (Format) выберем значение **Фиксированный** (Fixed).

Отредактируем подпись вычисляемого поля. Для этого выделим подпись и вызовем окно свойств. В окне свойств на вкладке **Макет** (Format) в строке **Подпись** (Caption) запишем - средний проходной балл группы. Такие дейст­вия, как изменение подписи или ввод выражения в поле можно выполнить и не обращаясь к свойствам элементов непосредственно внутри рамки элемента.

**Добавление текущей даты и номера страницы**

Для добавления в отчет *текущей даты* можно использовать встроенную функцию **Now**  из категории **Дата/время** (Date/time). Сначала в области **Заголовок отчета** (Report Header) с помощью кнопки **Поле** (Техt Вох) создается свободный элемент. В окне его свойств на вкладке **Данные** (Data) в строке **Данные** (Control Source) нужно записать значение =NOW(). Функцию **Now**  можно выбрать в построителе, отметив из встроенных функций категорию**Дата/время**. На вкладке **Макет** (Format) в раскрывающемся списке Формат поля (Format) выбирается значение **Длинный формат даты** (General Date). Подпись этого поля можно удалить.

Для добавления номера страницы в области нижнего колонтитула создается поле как свободный элемент. В окне его свойств на вкладке **Данные** (Data) в строку **Данные** (Control Source) записывается выражение =[Раgе]. Подпись этого поля можно отредактировать, записав в его свойствах на вкладке **Макет** (Format) в строке **Подпись** (Caption) - Ор.

Рассмотренный вариант добавления текущей даты и номера страниц показан на примере отчета, приведенного на рис. 4.

Существуют и другие способы формирования поля даты и номера страницы. Добавить в отчет поле текущей даты и времени можно, выполнив в режиме конструктора команду**Вставка** | **Датаивремя**(Insert | Date апd Time). Уста­новка в диалоговом окне **Дата и время** (Date апd Time) флажков **Формат даты** (Include Date) и/или **Формат времени** (Include Time) позволяет вставить текущую дату и/или текущее время и выбрать нужный вариант форматов (рис. 7). В случае отсутствия раздела заголовка поле текущих даты и времени вносится в разделданных.

В результате в отчете будет создано поле, в окне свойств которого на вкладке **Данные** (Data) в строке **Данные** (Control Source) будет записано выражение = Date ( ). Если в отчете имеется раздел заголовка, поле добавляется в этот раздел.

Добавить в отчет *поле нумерации страниц* можно в режиме конструктора с помощью команды **Вставка | Номера страниц** (Insert | Page Number). В диалоговом окне **Номера страниц** (Page Numbегs) выбираются параметры, опре­деляющие формат, расположение и выравнивание номеров страниц (рис. 8). Для печати номера страницы на первой странице устанавливается флажок**Отображать номер на первой странице** (Show Number оп First Раge.).

Вывод номеров страниц можно задать, формируя соответствующее выражение в окне свойств поля на вкладке **Данные** (Data) в строке **Данные** (Сопtrol Source). Такое выражение можно получить в построителе, открыв там папку **Общие выражения** и далее - категорию **Номер страницы** (Page N). При этом построитель предлагает выражение **="Cтраница"&(Page)**. Если выбрать категорию **Страница N из М** (Page N of М), построитель предлагает выражение **="Страница " & [Page] & " из " & [pages]**.

**Завершение оформления отчета**

Для окончательного оформления необходимо ввести в раздел **Заголовок отчета** (Report Header) название отчета. Для этого используется кнопка панели элементов **Надпись** (Label). Далее с помощью кнопок панели форматирования устанавливается нужный шрифт надписи. Для вывода названия отчета на последующих страницах оно вводится также в раздел **Верхний колонтитул** (Page Header). Для этого можно скопировать название из раздела **Заголовок отчета** (Report Header), вставить в раздел **Верхний колонтитул** (Page Header) и выбрать нужный шрифт. Затем следует в окне свойств отчета на вкладке **Макет** (Format) в раскрывающемся списке **Верхний колонтитул** (Page Header) выбрать значение**Без заголовка**. Окно свойств отчета может быть вызвано при установке курсора на квадрате пересечения линеек.

В примере формируемого отчета (см. рис. 4) заголовок списки СТУДЕНТОВ введен и отредактирован в область заголовка отчета и область Колонтитула. В соответствии с проектом отчета созданы линии, при использовании кнопки панели элементов **Линия** (Line).

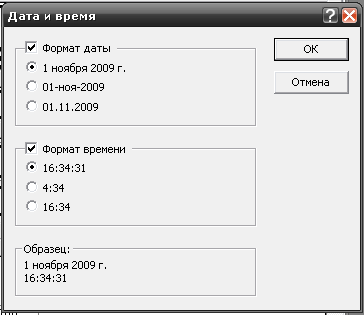


Рис. 7. Окно выбора формата даты и/или времени

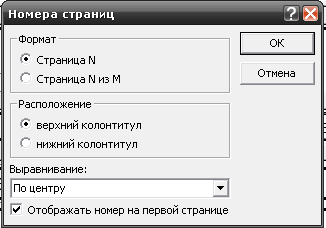


Рис. 8. Окно выбора параметров номеров страниц

**Просмотр и печать отчета**

**Просмотр отчета**

Переход из режима конструктора отчетов в режим предварительного просмотра осуществляется с помощью кнопки **Вид** (View) на панели конструктора. Для просмотра ранее созданного отчета выберите в окне базы данных на вкладке **Отчеты** (Reports) нужный отчет и нажмите кнопку **Просмотр** (Preview). Вид отчета при просмотре аналогичен его виду при печати

В режиме предварительного просмотра имеется своя панель инструментов. Список кнопок приведен на рис. 9.

Для просмотра нужных страниц отчета можно использовать стандартное *поле номера страницы* в нижнем левом углу окна отчета.

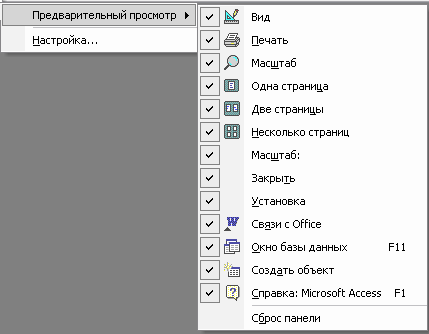


Рис. 9. Панель инструментов предварительного просмотра отчета

**Печать отчета**

Кнопка **Печать** (Print) панели инструментов режима предварительного просмотра (см. рис. 9) позволяет вывести данный отчет на печать. С помощью команды **Файл | Параметры** страницы (File | Page Setup) можно выбрать принтер, задать формат бумаги, размер полей, расстояние между строками, ориентацию (книжная, альбомная) и т. д. Команда **Файл | Печать** (File | Print) позволяет выбрать для печати отдельные страницы отчета, распечатать заданное число копий, вывести отчет в файл, который должен распечатываться в другое время.

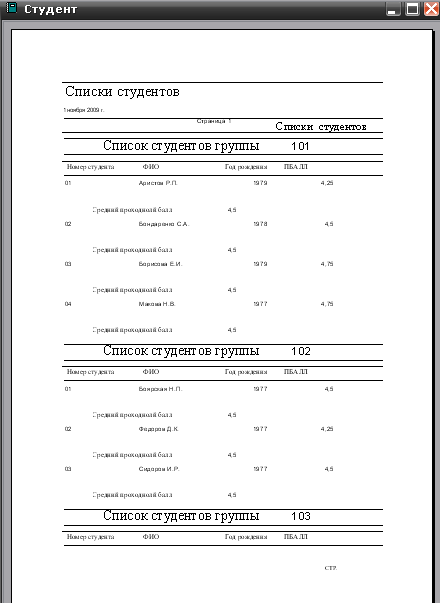


Рис. 10. Первая страница отчета Списки студентов с данными из таблицы СТУДЕНТ в режиме *предварительного просмотра*

На рис. 10 приведена *первая страница примера* отчета Списки студентов в режиме предварительного просмотра.

**Контрольные вопросы**

1. Что такое отчет?
2. Как создать отчет?
3. Для чего используется сортировка и группировка данных?
4. Что включает в себя окно конструктора отчетов?
5. Какие этапы включает процедура размещения полей из таблиц?
6. Что такое вычисляемое поле?
7. Как добавить вычисляемое поле?
8. Как добавитьтекущую дату и номер страницы?
9. Как просмотреть отчет?
10. Как распечатать отчет?

# Практическая работа №8

# Создание многотабличного отчета

**Цель:** научить студентов разрабатывать отчет на основе двух взаимосвязанных таблиц, создавать отчеты командой «автоотчет», просматривать содержимое отчета и отправлять его на печать.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

**Разработка многотабличных отчетов**

Наряду с однотабличными отчетами Accessпозволяет создавать более сложные отчеты, обеспечивающие вывод данных из нескольких взаимосвязанных таблиц базы данных. Для вывода взаимосвязанных данных многотабличный отчет автоматически используются связи, установленные в схеме данных базы данных.

Многотабличные отчеты могут содержать основную часть и включаемую часть, т.е. подчиненный отчет. Для каждой из этих частей в качестве источника данных выбирается своя таблица или несколько таблиц. Многотабличные отчеты могут также включать поля из разных таблиц и при этом не иметь подчиненных отчетов.

**Практическая часть**

**Разработка отчета на основе двух взаимосвязанных таблиц**

Рассмотрим построение отчета для двух таблиц, находящихся в одно-многозначных отношениях. В этих отношениях одна таблица является главной, а другая — подчиненной. Построение такого отчета имеет свои особенности, в зависимости от выбора главной или подчиненной таблицы в качестве основы отчета.

**Отчет с основной частью на базе главной таблицы**

Ниже рассмотрены действия пользователя, необходимые для создания отчета на основе двух таблиц, когда в качестве основы отчета выбирается главная таблица. Процесс разработки будем иллюстрировать на примере двух таблиц: КАФЕДРА И ПРЕПОДАВАТЕЛЬ базы Данных Учебный процесс.

Пусть необходимо подготовить отчет, содержащий сведения о кафедрах и включающий списки преподавателей по кафедрам.

Создадим проект макета отчета. Предположим, что макет отчета со списками преподавателей по кафедрам должен иметь вид, представленный на рис. 1. В соответствии с проектом макета в отчет предполагается выводить данные по каждой кафедре, включая название, код и телефон, а также фамилию ифотографию заведующего. Эти данные содержатся в таблице КАФЕДРА. В табличной части по каждой кафедре необходимо вывести построчно данные о преподавателях кафедры, которые содержатся в таблице ПРЕПОДАВАТЕЛЬ.

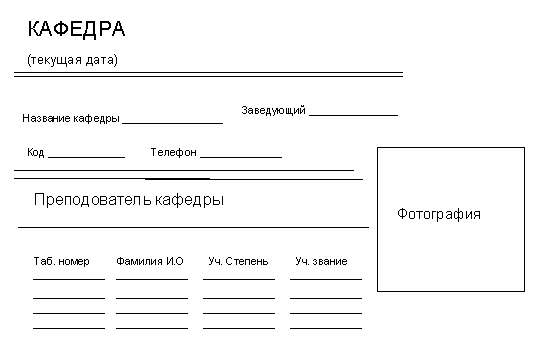


Рис. 1. Проект макета отчёта с данными о кафедрах и их преподавателях.

Проект макета даст основание выбрать в качестве основной таблицы отчета таблицу кафедра, а таблицу ПРЕПОДАВАТЕЛЬ — в качестве источника данных для *подчиненного отчета* со списком преподавателей.

**Создание основной части отчета в режиме конструктора.**

**Выбор режима создания отчета и основного источника**

Выберем в окне базы данных объект **Отчеты** (Reports) и нажмем кнопку**Создать** (New). В окне **Новый отчет** (NewReport) выберем вариант **Конструктор** (Design) для построения отчета в режиме конструктора, а также таблицу, которая будет источником данных для основной части отчета.

В рассматриваемом примере выберем в качестве таблицы-источника главную таблицу КАФЕДРА.

**Создание и размещение элементов отчета**

В окне конструктора отчетов (рис. 2) добавим раздел **Заголовок** отчета (ReportHeader) с помощью кнопки **Заголовок/примечание** отчет (ReportHeader/Footer) панели инструментов конструктора отчетов. В разделе заголовка далее можно создать элемент с текстом заголовка отчета в соответствии с проектом, воспользовавшиськнопкой панели элементов**Надпись**(Label).

В рассматриваемом примере создадим в области заголовка надпись с текстом КАФЕДРА

Для размещения полей таблицы источника можно воспользоваться списком полей источника, который открывается кнопкой **Список полей**(FieldList) на панели инструментов конструктора отчетов. Далее переместим с помощью мыши нужные поля из списка в область данных.

Разместим таким образом поля таблицы КАФЕДРА: НКАФ, ККАФ, ТЕЛ, ЗАВ и ФОТО. Затем отредактируем созданные поля и их подписи в соответствии с проектом макета (см. рис. 1).

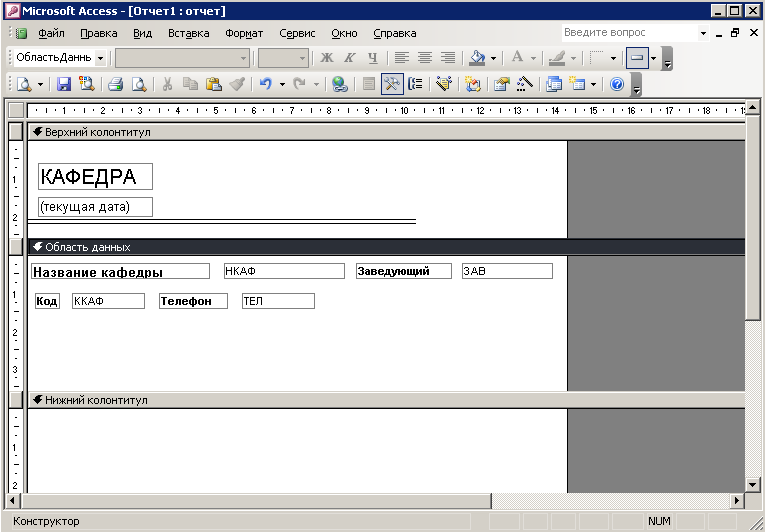


Рис. 2. Заполнение основной части отчёта о кафедрах в режиме конструктора

Включение текущей даты **в Заголовок отчета**(ReportHeader) осуществляется командой **Вставка | Дата и время** (Insert | DateandTime). Номер страницы можно поместить в **Нижний колонтитул**(PageFooter). Детально необходимые действия для этого были описаны при создании однотабличного отчета. Сохранение созданного макета отчета осуществляется с помощью команды **Файл | Сохранить**(File | Save) или нажатием соответствующей кнопки **Сохранить**(Save) на панели инструментов конструктора отчетов.

Сохраним созданный отчет под именем Кафедры.

**Создание подчиненного отчета**

Для вывода в отчет данных из подчиненной таблицы в виде табличной части необходимо создать отчет, который будет включен в основную часть отчета в качестве подчиненного.

В рассматриваемом примере для вывода в отчет списков преподавателей по кафедрам из таблицы ПРЕПОДАВАТЕЛЬ ПОДГОТОВИМ подчиненный отчет.

**Создание отчета командой Автоотчет**

На основе подчиненной таблицы можно создавать обычный однотабличный отчет. Для автоматического создания отчета в окне **Новый отчет**(NewReport) воспользуемся вариантом **Автоотчет: ленточный**(AutoReport; Tabular). В качестве источника данных этого отчета выберем подчиненную таблицу. В примере это таблица ПРЕПОДАВАТЕЛЬ.

Выбор ленточного отчета позволит получить мпогозаписевый макет для подчиненного отчета с подписями полей в названиях столбцов, которые были заданы в свойствах таблицы. Завершается создание автоотчета выводом в режиме просмотра отчета, а котором в качестве заголовка взято имя таблицы. В отчет включатся все поля таблицы источника.

Отчет для таблицы ПРЕПОДАВАТЕЛЬ, полученный по команде **Автоотчет: ленточный**после перехода в режим конструктора, показан на рис. 3.

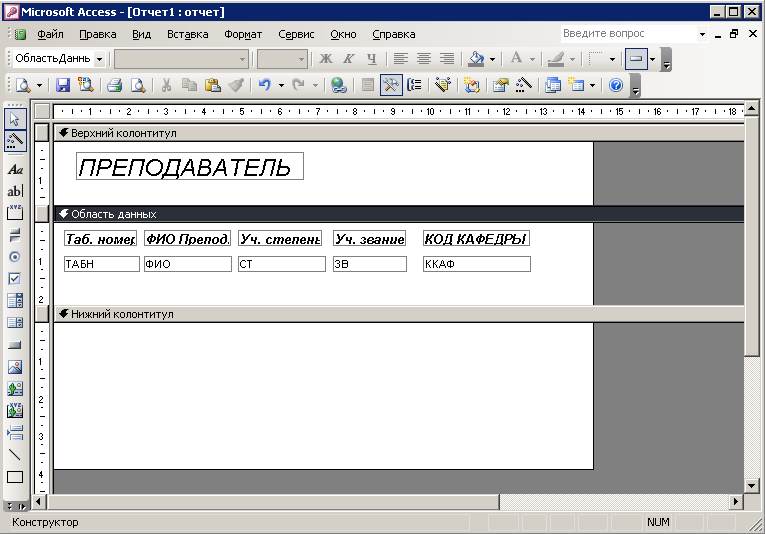


Рис. 3. Отчёт в режиме конструктора, созданный командой Автоотчет: ленточный

**Доработка подчиненного отчета в режиме конструктора**

Для того чтобы подчиненный отчет можно было включить в основную часть отчета в нужном виде в соответствии с проектом макета, его, как правило, необходимо доработать в режиме конструктора.

Верхний колонтитул подчиненного отчета, в котором после создания командой **Автоотчет: ленточный**оказались размещенными надписи полей (см. рис. 3), не отобразится в режиме просмотра полного отчета. Поэтому для отображения в полном отчете заголовков столбцов табличной части можно перенести названия столбцов из верхнего колонтитула в заголовок подчиненного отчета. Для этого выделим все подписи полей в верхнем колонтитуле и перетащим их с помощью мыши в область заголовка. После этого в подчиненном отчете целесообразно удалить оба колонтитула, нажав кнопку **Колонтитулы**(PageHeader/Footer) на панели конструктора.

В рассматриваемом примере после таких действий подчиненный отчет примет вид, показанный на рис. 4. Здесь в нем были также удалены поле ккаф и его подпись код кафедры, т. к. в подчиненном отчете значения этого поля будут повторяться во всех строках о преподавателях одной кафедры, а однократное отображение кода кафедры предусмотрено в основной части отчета. Отредактированный подчиненный отчет сохраним под новым именем преподаватели, выполнив команду **Файл | Сохранить как** (FileSaveas).

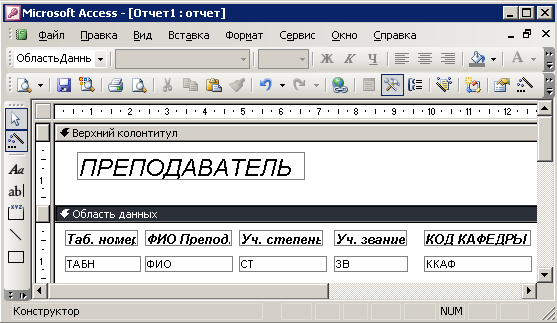


Рис. 4. Подчиненный отчёт для включения в двухтабличный отчет Кафедры

**Включение подчиненного отчета**

Для включения подчиненного отчета в основной можно использовать графический способ. Для этого достаточно в окне базы данных выделить имя подчиненного отчета и перетащить его с помощью мыши в область данных отчета, построенного на основе главной таблицы

Для рассматриваемого примера сделаем активным окно базы данных, разместим его рядом с окном отчета Кафедры в режиме конструктора (см. рис. 2). Далее перетащим с помощью мыши значок отчета преподаватели из окна базы данных в область данных отчета. После этого подчиненный отчет встраивается в общий отчет. После такого встраивания слева над рамкой подчиненного отчета будет размещен элемент с надписью подчиненного отчета (преподаватель). Его можно удалить. Отчет кафедры в режиме конструктора после внедрения подчиненного отчета преподаватели показан на рис. 5. Все подписи полей затемнены. На рис. 5 подчиненный отчет отображен внутри отчета кафедры, где доступен для редактирования, как и основной отчет.

Для того чтобы подчиненный отчет при отображении не был обрамлен рамкой, необходимо выделить его, как показано на рис. 5, открыть окно свойств и на вкладке **Макет**(Format) в раскрывающемся списке **Тип границы**(BorderStyle) выбрать значение**Отсутствует** (Transparent).

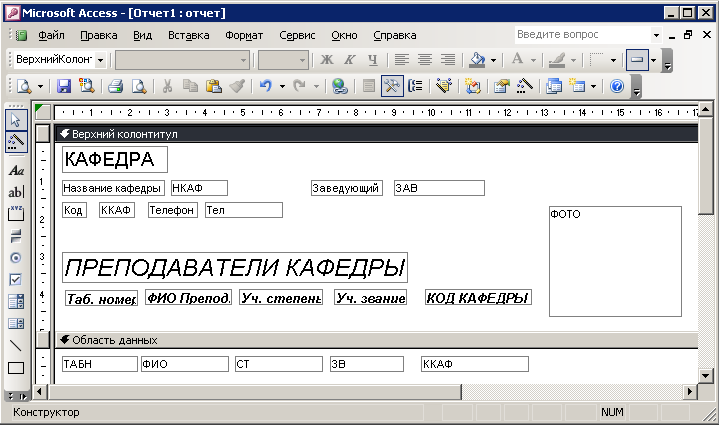


Рис. 5. Отчёт на основе таблиц КАФЕДРА и ПРЕПОДАВАТЕЛЬ в режиме конструктора со встроенным подчиненным отчётом.

Рассмотрим второй способ включения подчиненного отчета. Для включения подчиненного отчета можно также воспользоваться кнопкой панели элементов **Подчиненная форма/отчет** (Subform/Subreport). При выключенной кнопке панели элементов **Мастер элементов**(ControlWizards) в области данных основного отчета создается элемент **Свободный**(Unbound) с надписью **Внедренный**(Child №). В окне свойств этого элемента на вкладке **Данные**(Data) в раскрывающемся списке **Объект-источник**(SourceObject) необходимо выбрать имя встраиваемого отчета.

**Просмотр содержимого двухтабличного отчета и его печать**

Просмотр содержимого отчета, содержащего подчиненный отчет, осуществляется после конструирования нажатием кнопки **Предварительный просмотр**(ReportView) на панели конструктора отчетов.

Для сконструированного в примере отчета (рис. 6) убедимся в том, что он соответствует приведенному выше проекту макета отчета.

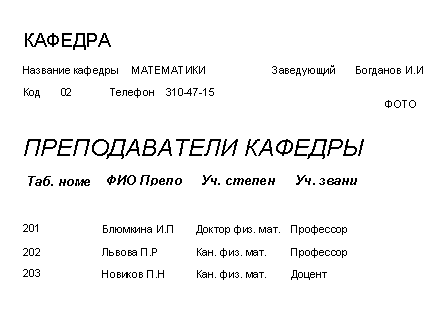


Рис. 6. Первая страница многотабличного отчёта Кафедры в режиме просмотра

**Сортировка записей**

Отображение записей отчета в нужном порядке определяется критерием сортировки. Для этого в режиме конструктора нажмем кнопку в области выделения отчета (на пересечении линеек) и вызовем контекстно-зависимое меню. В этом меню выберем команду **Сортировка и группировка**(SortingandGrouping), а в открывшемся диалоговом окне — поле сортировки. При необходимости сортировки записей подчиненного отчета нужно выполнить такие же действия для подчиненного отчета.

Вывод отчета на печать можно осуществить нажатием кнопки **Печать** (Print) на панели предварительного просмотра.

**Контрольные вопросы**

1. Что такое многотабличный отчет?
2. Для размещения полей таблицы источника можно воспользоваться…?
3. Могут ли многотабличные отчеты не содержать подчиненных таблиц?
4. Если таблицы находятся в одно-многозначных отношениях, что это означает?
5. Как добавить раздел Заголовок и Примечания в отчет?
6. Как вставить Дату и время в отчет?
7. Как создать подчиненный отчет?
8. Как подчиненный отчет включить в основную часть отчета?
9. Как убрать Колонтитулы?
10. Как сменить Тип границ?

# Практическая работа №9

# Создание макроса для решения задачи, использующей запросы.

**Цель работы:** научиться создавать макрос, формировать макрокоманды, формировать ссылки на объекты, организовать выполнение макроса.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

Пусть необходимо рассчитать количество студентов в группах и внести эти данные в поле кол и таблице ГРУППА. Подсчет количества студентов реализован запросом Число студентов в группе. Обновление ноля кол на основе полученных вэтом запросе данных выполняет запрос обновление ГРУППА\_КОЛ. Последовательное выполнение этих запросов решает поставленную: задачу.

**Практическая часть**

Создадим макрос, который последовательно выполнит эти запросы. Кроме того, включим в макрос макрокоманды, автоматизирующие действия пользователя по управлению объектами на экране при решении задачи и, в частности, действия, обеспечивающие отображение результатов решения задачи на экране в удобном виде.

1.Чтобы начать создание макроса, выберите в окне базы данных вкладку **Макросы** (Macros) и нажмите кнопку**Создать**(New). Откроется окно макроса.

2.Макрокоманды, составляющие макрос, вводятся в столбец **Макрокоманда** (Action). Для ввода макрокоманды нажмите кнопку раскрытия списка макрокоманд в этом столбце и выберите макрокоманду**Открыть Запрос**(OpenQuery), чтобы выполнить первый запрос.

3. Выберите имя запроса число студентов вгруппе из списка в области **Аргументы макрокоманды** (ActionArguments) в нижней части окна макроса.

4.Введите в столбец **Примечание** (Comment) текст, описывающий результат выполнения макрокоманды, например, Создание таблицы с результатами подсчета числа студентов в группе.

5.Длявыполнения запроса обновление. ГРУППА\_КОЛ введите следующую макрокоманду**Открыть Запрос** (OpenQuery).Порядок размещения макрокоманд в бланке определяет последовательность их выполнения.

6. Для включения этой макрокоманды в макрос воспользуйтесь перемещением объектов с помощью мыши. В окне базы данных выберите запрос обновление ГРУППА\_КОЛ и перетащите его в строку макрокоманды. Сформируется макрокоманда**ОткрытьЗапрос** (OpenQuery), в аргументах которой автоматически сформируется имя открываемого запроса, режим отображения запроса и режим работы с егоданными (рис. 1)

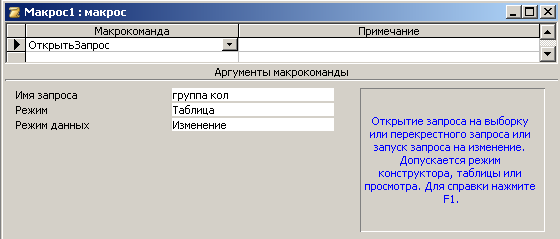


Рис. 1. Макрокоманда, сформированная перетаскиванием запроса из окна базы данных

7.Сохраните макрос под именем**Расчет числа студентов в группах**, воспользовавшись кнопкой панели инструментов макроса**Сохранить** (Save).

8. Выполните макрос, нажав кнопку **Запуск** (Run) панели инструментов

9. Во время выполнения макроса на экране отображаются предупреждающие сообщения, выдаваемые выполняющимися запросами. Для временного отключения вывода на экран окон сообщений дополните макрос макрокомандой**УстановитьСообщения** (SetWamings). В строке аргумента**Включить сообщения** (WarningsOn} выберите значение **Нет**(No) Для возобновления вывода сообщений после выполнения запросов следовало бы задать значение **Да** (Yes) Однако эту макрокоманду можно опустить, поскольку после прекращения выполнения макроса вывод предупреждений восстанавливается автоматически.

10.Организуйте вывод на экран таблиц ГРУППА и число студентов, чтобы сделать наглядным процесс работы макроса и обновления таблицы. Для этого допишите макрос, как показано на рис. 2. В представленном макросе не используются условия отбора, и макрокоманды выполняются последовательно одна за другой. Порядок их выполнения определяется

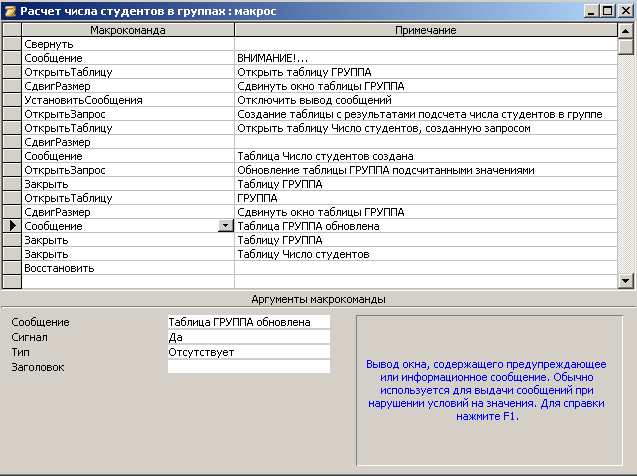


Рис. 2. Макрос задачи в режиме конструктора алгоритмов решения задачи

11.Закройте макрос и запустите его из окна базы данных, воспользовавшись кнопкой **Запуск**(Run).

12.Ход и результаты выполнения запроса по обновлению поля кол отображаются в выводимых на экран сообщениях и таблицах

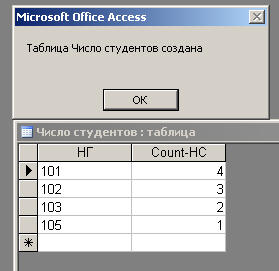
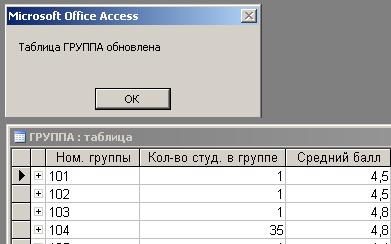


Рис. 3. Результаты выполнения макроса после вывода второго сообщения

13.Создайте в любой форме кнопку запуска макроса с помощью мыши.

Для этого в окне базы данных выберите макрос **Расчет числа студентов в группах**, который необходимо запускать при нажатия кнопки. Переместите его с помощью мыши в форму, открытую в режиме конструктора. В форму будет помещена кнопка, связанная с данным макросом. Кнопка будет иметь подпись, соответствующую имени макроса

Ниже приведено назначение используемых в программе макрокоманд

• **Свернуть** (Minimize) — позволяет свернуть окно, которое было активным в момент запуска макроса па выполнение. Если макрос запускается па выполнение из окна базы данных и, кроме этого окна, нет открытых окон, на экране будет отображено только свернутое окно базы

• **ОткрытьТаблицу**(OpenTable) — открывает таблицу ГРУППА Имя таблицы и режим ее открытия укатываются аргументами макрокоманды. Открытая таблица ГРУППА будетОтображаться на экране до выполнения макрокоманды Закрыть (Close). Это позволит просмотреть содержимое таблицы до и после выполнения запросов.

• **СдвигРазмер**(MoveSize) — позволяет задать размер и местоположение окна активного объекта. Выполнение макрокоманды после открытия таблицы изменяет параметры окна таблицы

• **Сообщение**( MsgBos) — выдает сообщение, указанное в аргументах макрокоманды.

• **Восстановить** (Resiure) — позволяет восстановить окно, свернутое перед решением задачи.

• **Установить Сообщения**{SetWarnings) — отключает вывод на экран окон сообщении с предупреждениями па время выполнения макроса.

В рассматриваемом примере отключается вывод сообщений, выдаваемых ниже выполняемыми запросами. Для временного прекращения вывода предупреждений в строке аргументаВключить сообщения(WarningsOn) задается значение Нет (No) Для возобновления вывода сообщений следует задать значение Да (Yes). После прекращения выполнения макроса вывод предупреждений восстанавливается автоматически

Замечание

Допускается использование в макросе еще не созданных объектов. Например, в аргументе **Имя Таблицы** (TableName) макрокоманды**Открыть Таблицу** (OpenTable) указано имя еще не существующей таблицы число студентов. Однако к моменту выполнения макрокоманды эта таблица должна быть создана в базе данных. В рассматриваемом примере таблица Число студентов создается запросом в ходе выполнения макроса до выполнения открывающей ее макрокоманды.

**Макрос синхронной обработки данных двух форм**

Рассмотрим, как с помощью группы макросов, которая запускается при инициировании пользователем события в одной форме, можно отфильтровать записи в другой форме.

Пусть при просмотре данных о предмете, изучаемом студентами, необходимо отображать информацию обо всех студентах, получивших двойки по этому предмету.

Для решения задачи используем формы ПРЕДМЕТ и двоечники. Однотабличная однозаписеваяформа предмет была создана ранее. Она позволяет просматривать информацию о предметах. Форму Двоечники создайте на основе запроса, который выбирает из таблиц успеваемость и студент поля КП (код предмета), ФИО (студента), ОЦЕНКА и в котором для поля ОЦЕНКА,в строке Условие отбора записано значение 2. Форма отобразит список всех записей о двойках, полученных студентами по всем предметам. Обе формы имеют поле КП, содержащее код предмета

При работе с формой ПРЕДМЕТ Одновременно на экране должна открываться форма двоечники. Причем через эту форму должны отображаться только те записи, которые соответствуют выбранному пользователем предмету в форме ПРЕДМЕТ.

Создайте группу макросов с именем двоечники, включающую макрос M1 и макрос м2. Пусть макрос M1 выполняется при открытии формы ПРЕДМЕТ, открывает форму двоечники и определяет ее местоположение на экране Макрос М2 фильтрует записи о двоечникахпо значению поля КП, взятому из текущей записи формы ПРЕДМЕТ. Макрос должен выполняться, когда пользователь, работая в форме ПРЕДМЕТ, инициирует событие Вход для поля. Создание групп макросов позволяет объединить макросы, предназначенные для решения одной задачи, и упростить сопровождение приложения

1. Для создания макроса выберите в окне базы данных вкладку **Макросы**(Масros) и нажмите кнопку**Создать** (New).

2. Для того чтобы дополнить окно макроса столбцом, позволяющим создать группу макросов, щелкните на панели инструментов **Конструктор макросов**(MacroDesign) на кнопке **Имена макросов** (MacroNames).

3. Запишите макрокоманды макросов Mlи М2, как показано на рис 4, и сохраните группу макросов под именем двоечники. Это имя будет выводиться в списке макросов в окне базы данных.

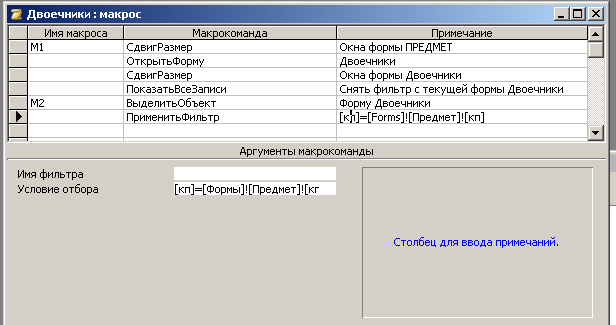


Рис. 4. Макрос, организующий синхронный просмотр данных в двух формах.

4.Для запуска макроса M1 установите связь события **Открытие** (OnOpen) формы ПРЕДМЕТ с макросом. Откройте форму в режиме конструктора, щелкните на кнопке **Свойства** (Properties), на вкладке **События** (Event) а качестве значения свойства **Открытие** (OnOpen) выберите имя макроса Двоечники.M1

5. Для запуска макроса М2 установите в качестве значения свойства события **Двойное нажатие кнопки**(OnDblClick) элемента управления поля кп формы ПРЕДМЕТ значение Двоечники М2

Событие **Двойное нажатие кнопки** (OnDblClick) для элемента управления кп возникает, когда пользователь дважды нажимает и быстро отпускает левую кнопку мыши в тот момент, когда курсор мыши установлен на поле кп или присоединенной к нему надписи.

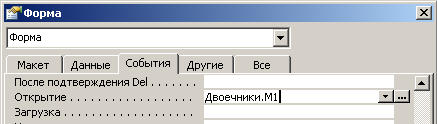


Рис. 5. Окно свойств формы ПРЕДМЕТ со ссылкой на макрос в событии **Открытие**

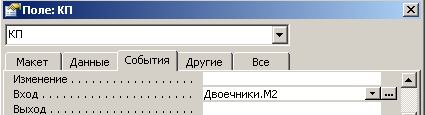


Рис. 6. Окно свойств поля КП со ссылкой на макрос в событии **Двойное нажатие кнопки**

6. Чтобы проверить работу макросов, откройте форму ПРЕДМЕТ. На экране будет отображен результат работы Ml — Открытые формы Двоечники и ПРЕДМЕТ. Причем в форме двоечники отобразятся записи о двойках, полученных по всем предметам

7. Дважды щелкните на поле с кодом предмета в форме ПРЕДМЕТ. Результат выполнения макроса М2 отобразится на экране в виде, представленном на рис. 7. К форме двоечники был применен фильтр, в котором в качестве условий отбора записано. [Kп]=Forms! [предмет]! [кп], и теперь в ней отображаются только записи со значением кода предмета из формы ПРЕДМЕТ.

8. Чтобы макрос М2 выполнялся, когда пользователь, работая в форме ПРЕДМЕТ, устанавливает курсор в поле кп:

•Удалите связь макроса нг с событием **Двойное нажатие кнопки** (OnDblClick) для элементауправления кп

• Установите его связь с событием **Вход** (Enter) этого же элемента управления.

Событие **Вход** (Enter) наступает при перемещении курсора в поле кп, при переходе к другой записи, если текущим является поле кп. Кроме того, при открытии формы, наряду с событием **Открытие** (OnOpen), для текущего элемента управления формы наступает событие Вход (Enter). При открытии формы текущим элементом управления является поле, которое стоит первым всписке, определяющем последовательность перехода по полям при нажатии клавиши табуляции. Если па первом месте стоит поле кп, то оно будет текущим и для него наступит событие **Вход** (Enter). В результате при открытии формы вы­полнятся оба макроса.

Окно действие пользователя может приводить к возникновению нескольких событий. При этом важно знать последовательность наступления этих событий, т.к. это определяет и последовательность выполнения макросов.

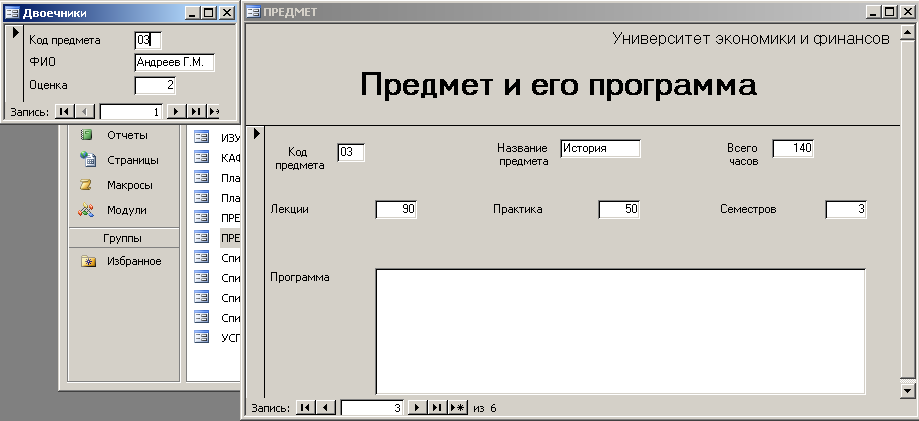


Рис. 7. Результат выполнения макроса М2.

9.Просмотрите последовательность перехода по полям в форме ПРЕДМЕТ, выбрав команду **Последовательность перехода** (TabOrder) в контекстном меню формы или элемента управления. Если поле КП стоит на первом месте, переметите его ниже.

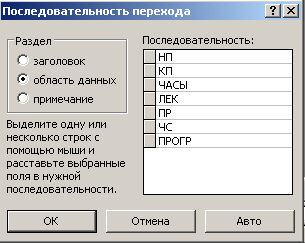


Рис. 8. В последовательности перехода поле КП сдвинуто на второе место

10.Откройте форму ПРЕДМЕТ и убедитесь, что при переходе от записи к записи синхронно отображаются соответствующие записи формы двоечники. Курсор должен оставаться в поле КП.

**Контрольные вопросы**

1. Что такое макрос?
2. Как создать макрос?
3. Порядок размещения макрокоманд в бланке определяет…?
4. Как запустить макрос?
5. Как создать в форме кнопку запуска макроса?
6. Какие используются в программе макрокоманды?
7. Как отключить вывод сообщений?
8. Что означает событие Двойное нажатие кнопки?
9. Как проверить работу макроса?
10. Что означает командаПоследовательность перехода?

# Практическая работа №10

**Создание макроса, обеспечивающего переходот диалоговой работы**

**с базой данных к выводу отчетов**

**Цель работы:** научиться создавать макросы и группы макросов, формировать

макрокоманды и ссылки на объекты, реализовать алгоритм с помощью условий и

ветвлений в макросе, организовать выполнение макроса.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

Пусть необходимо в сеансе ввода данных об успеваемости подготавливать и выводить отчеты о полученных студентами двойках и пятерках. Для накопления данных используются таблицы двоечник и отличник, которые включают поля нг (номер группы), нс(номер студента в группе), кп (код предмета) иФИО студента. Таблицы имеют составной ключ нг + нс + кп. Ключи в таблицах не позволяют повторно вводить записи с одинаковыми значениями ключа

**Практическая часть**

Ввод данных о результатах экзаменов производится через форму ведомость успеваемости (рис. 1.)

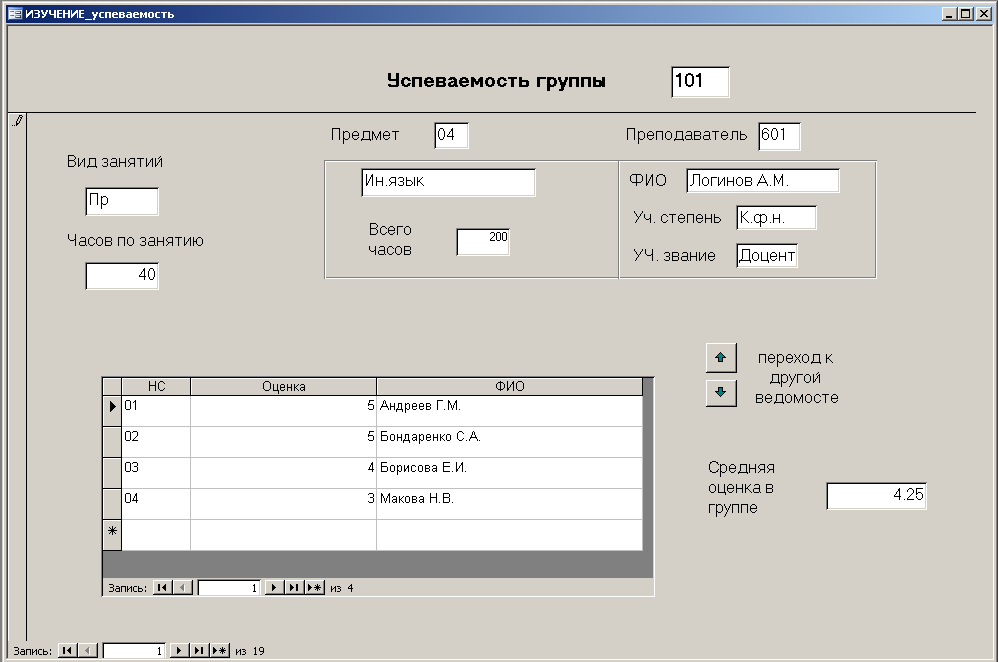


Рис. 1. Форма для ввода данных об успеваемости

Основная часть формы отображает запись из таблицы ИЗУЧЕНИЕ, дополненную связанными данными из главных по отношению к ней таблиц предмет и ПРЕПОДАВАТЕЛЬ. Подчиненная форма (ее имя — УСПЕВАЕМОСТЬ) отображает связанные с таблицей ИЗУЧЕНИЕ записи из таблицы УСПЕВАЕМОСТЬ, дополненные фамилией студента. Подсхема данных, на которой строится форма, может быть выделена из общей схемы данных базы Учебный процесс.

1.Для решения -задачи создайте группу макросов Списки успеваемости (рис. 2), включающую макросы:

* Начало, который будет открывать формы ДВОЕЧНИК и ОТЛИЧНИК, а также очищать таблицы перед формированием новых отчетов. Это позволит в очередном сеансе ввода данных в таблицу УСПЕВАЕМОСТЬ для одной или нескольких групп студентов по одному или нескольким предметам выдать отчеты для этих групп, а также использовать макрос при корректировках таблицы. Отчеты будут содержать данные только по результатам одного сеанса ввода.

• Формирование, который будет заполнять таблицы ДВОЕЧНИК иОТЛИЧНИК через соответствующие однозаписевые формы.

• Отчет, который будет распечатывать отчеты ДВОЕЧНИК и ОТЛИЧНИК по завершении ввода данных об успеваемости (отчеты созданы на основе соответствующих таблиц)

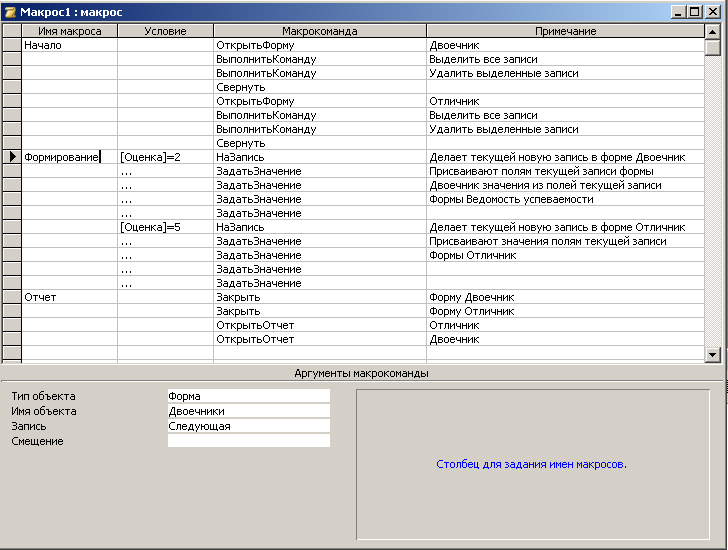


Рис. 2. Макрос формирования отчетов о двоечниках и отличниках при вводе оценок из экзаменационной ведомости.

*2.* В первом макросе с именем начало введите макрокоманды**Открыть Форму** (OpenForm), которые используются для заполнения таблиц ДВОЕЧНИК иОТЛИЧНИК во втором макросе.

3. Непосредственно за макрокомандой**Открыть Форму** (OpenForm) введите **Выполнить Команду**(RunCommand). Эта макрокоманда выполняет для текущего объекта команду стандартного меню, заданную ее аргументом. Выполнение этой макрокоманды эквивалентно выбору пользователем указанной команды в стандартном меню Access. Выберите в качестве аргумента команду **SelectAliRecords** (Выделить все записи). В качестве аргумента второй макрокоманды**Выполнить Команду** (RunCommand) выберите **Delete**(Удалить). Действия этих макрокоманд для открытой формы равносильны выполнению соответствующих команд в меню **Правка** (Edit).

Таким образом, выполняется удаление всех записей в таблицах двоечник и отличник.

4. Если формы не должны отображаться на экране, введите вслед за макрокомандами**Выполнить Команду** (RunCommand) макрокоманду **Свернуть** (Minimize), применяемую к окну текущего объекта. Если вы хотите наблюдать синхронный процесс ввода данных и заполнения таблиц для отчета, можно не сворачивать формы, а дополнить макрос макрокомандами, определяющими размещение объектов на экране, чтобы получить удобное отображение всех форм.

5. Для выполнения макроса в момент, когда пользователь открывает форму Ведомость успеваемости, в**свойства формы в строку события Открытие(OnOpen) введите имя макроса** Cписки успеваемости. Начало. В результате этого макрос будет выполняться один раз в начале сеанса ввода данных.

6.Второй макрос Формирование анализирует введенную оценку и, в зависимости от результата, формирует новую запись в одной из таблиц Двоечники или Отличники. Он должен выполняться при обновлении поля оценка в записях формы ведомость успеваемости. Свяжите этот макрос со свойством события поля ОЦЕНКА**После обновления** (AfterUpdate), введя **в строку свойства имя этого макроса** Списки успеваемости.Формирование.

7. Чтобы начать формирование записи о двоечнике, введите макрокоманду**НаЗапись** (GoToRecord), которая позволит сделать текущей новую запись в форме ДВОЕЧНИК. Укажите в аргументе макрокоманды имя этой формы, а в аргументе **Запись** (Next) укажите, что текущей должна стать новая запись.

8. Следующие макрокоманды должны заполнить поля новой записи. Это позволяет сделатьмакрокоманда**Задать Значение** (SetValue), которая присвоит полям новой записи формы двоечник значения из полей текущей записи формы Ведомость успеваемости. Аргумент макрокоманды**Элемент** (Item) определяет поле, которому должно присваиваться значение. Аргумент **Выражение**(Expression) определяет поле, из которого выбирается это значение. Выражение не должно начинаться со знака равенства.

Замечание

Для ссылки на попе или элемент управления в форме, из которой вызывается макрос, достаточно указать только имя элемента, для ссылки на Другие объекты должен использоваться полный синтаксис.

9. В аргументе **Элемент** (Нет) запишитеForms!двоечник! [имя поля]. В первой макрокоманде укажите поле нг, в следующих — нс, кп и ФИО.

Замечание

Воспользуйтесь построителем для формирования сложной ссылки на элемент. Построитель вызывается щелчком на кнопке, расположенной справа от строки аргумента.

10. Поскольку присваиваемые значения выбираются из полей текущей записи подчиненной формы УСПЕВАЕМОСТЬ подчиненная форма, вызывающей макрос, в аргументе **Выражение**(Expression) макрокоманд**Задать Значение**(SelValue), запишите только имена этих полей: нг, нс, кп и ФИО соответственно.

11. Запишите аналогичный набор макрокоманд для формирования записи об отличной оценке

12. Макрокоманды формирования записи о двоечнике должны выполняться только в том случае, если в поле ОЦЕНКА введена двойка. Макрокоманды формирования записи об отличной оценке - только если в поле ОЦЕНКА введена пятерка. Чтобы анализировать введенную в форме ведомость успеваемости оценку и, в зависимости от результата, формировать новую запись в одной из таблиц ДВОЕЧНИК ИЛИ ОТЛИЧНИК, организуйте выполнение различных макрокоманд в зависимости от результата проверки условия.

13. Прежде всего, дополните окно макроса столбцом Условие (Condition), нажав соответствующую кнопку на панели инструментов.

14. В строку первой макрокоманды**НаЗапись**(GoToRecord) в столбец Условие (Condition) введителогическое выражение [ОЦЕНКА] =2. Теперь, если выражение примет значение True (Истина), т. е. при вводе в поле оценка двойки, будет выполняться эта макрокоманда и все последующие, вплоть домакрокоманды с новым условием.

15. Чтобы при значении выражения False не выполнялась группа макрокоманд, формирующая запись о двойке, проставьте в столбце условий этих макрокоманд многоточие (...).

16. В строку второй макрокоманды**НаЗапись** (GoToRecord) в столбец **Условие** (Condition) введителогическое выражение [ОЦЕНКА] =5, а в других макрокомандах этой группы проставьте многоточие (...). Введенные в столбец условия позволят пропустить выполнение этих макрокоманд, если введенная оценка отличается от пятерки.

Заданные в макросе условия позволяют выполнить формирование записи при вводе в поле ОЦЕНКА двойки или пятерки. При вводе других оценок ниодна макрокоманда макроса не будет выполнена.

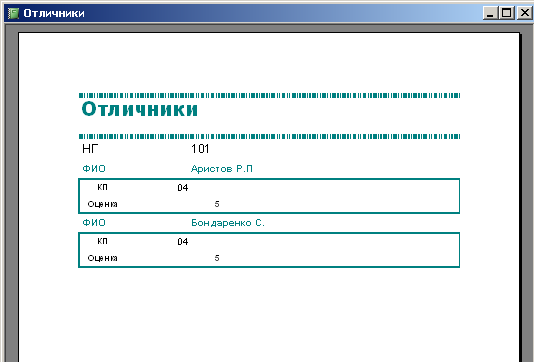


Рис. 3. Отчет об отличных оценках по введенным ведомостям успеваемости

17. Третий макрос Отчеты должен выполняться при закрытии пользователем формы ведомость успеваемости. Введите в свойствах формы в строке события **Закрытие**(OnClose) имя этого макроса: Списки успеваемости. Отчеты

18.Введите в этот макрос макрокоманды**Закрыть**(Close) для форм ДВОЕЧНИК и ОТЛИЧНИК, выбрав в строке аргументов **Сохранение** (Save) значение Да (Yes). Выполнение этих макрокоманд позволит сохранить в таблице последнюю сформированную запись и таким образом не потерять ее в отчетах.

19.Для вывода отчетов на экран в режиме предварительного просмотра используйте макрокоманду**Открыть Отчет**(OpenReport). Имя отчета и режим ее открытия укажите в аргументах макрокоманды. Если нет необходимости просматривать отчет, можно сразу получить печатный документ, установив в аргументе режим печати. Открытый для просмотра отчет будет oотображаться на экране до тех пор, пока пользователь не закроет его.

20. Для проверки макроса откройте форму ведомость успеваемости, введите новые записи, закройте форму. На экране отобразятся отчеты отличник и двоечник. На рис. 3 представлен отчет об отличных оценках, полученных студентами 101 группы по трем предметам.

**Организация диалога в макросе**

В предыдущем примере при закрытии формы ведомость успеваемости, через которую вводились данные об оценках по сданным студентами предметам, было предусмотрено выполнение макроса для безусловной выдачи на экран отчетов о двоечниках и отличниках. Дополним макрос макрокомандами, которые позволят определить необходимость вывода отчетов в диалоговом режиме.

1.Используйте в качестве диалогового окна вспомогательную форму Диалог. Форму можно создать в режиме конструктора, не указывая источника записей, с единственным элементом - полем со списком значений. На рис. 4 представлена форма диалог, в которой поле со списком имеетимя Поле Со Списком0. В свойствах поля со списком задайте в строке **Источник строк** (RowSource) имена отчетов, выполнение которых может запросить пользователь, а также предусмотрите отказ от вывода отчетов

2.Измените группу макросов списки успеваемости, как показано на рис. 6. В макросе Отчет вместо макрокоманд вывода отчетов запишите макрокоманду открытия формы диалог.

**Замечание**

Выделение макрокоманд открытия отчетов в отдельный макрос Отчет! позволяет пользователю организовать работу с формой Диалог.

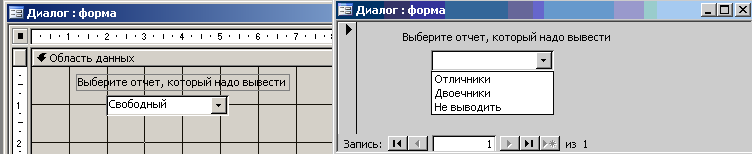


Рис. 4. Форма, включающая поле со списком значений

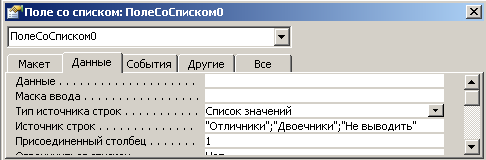


Рис. 5. Источники строк поля со списком значений в форме Диалог

1. В макросе Отчет1 введите макрокоманды вывода отчетов и условия их выполнения.

Текстовые константы, с которыми сравниваются введенные значения, заключаются в кавычки.

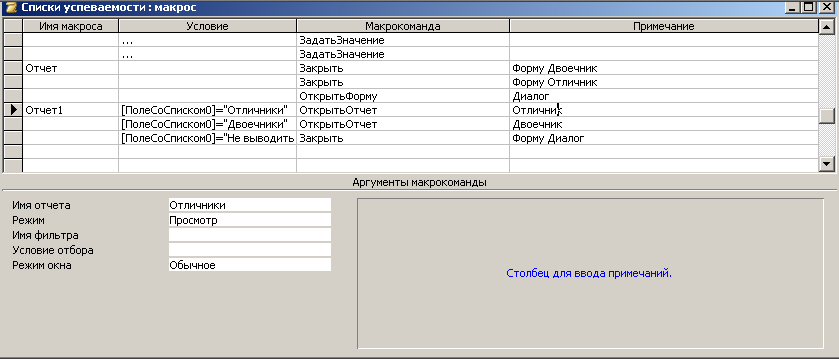


Рис. 6. Макрос вывода выбранного пользователем отчета

4. В форме диалог для элемента поле Со Списком0 откройте окно свойств (рис. 7) и выберите в раскрывающемся списке **После обновления** (AfterUpdate) значение Списки успеваемости. Отчет1.

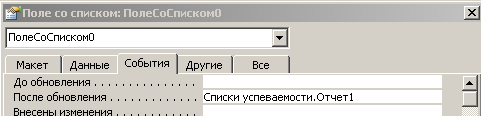


Рис. 7. Связь события**После обновления с макросом**

5. Откройте форму Ведомость успеваемости, введите несколько записей о полученных студентами оценках, в том числе двойках и пятерках.

6. Закройте форму и вы перейдете в режим отображения окна формы Диалог.

7. Для проверки работоспособности макроса отчет1 выберите в форме диалог одно из значений. Начнет выполняться макрос отчет1, в котором будет анализироваться введенное значение и, если задано такое условие, будет выведен отчет.

8. Для распечатки отчета в любое время откройте форму Диалог, выберите нужный отчет в списке, и макрос отчет1 выведет выбранный отчет на экран.

9. Для вывода вопросов о необходимости печатать отчеты можно использовать функцию MsgBox. Замените макрокоманды в макросе Отчет1, как показано на рис. 8.

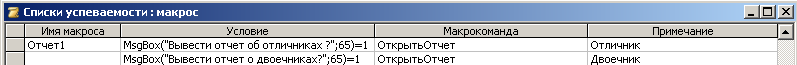


Рис. 8. Использование функции в условии макрокоманды

10. Функция MsgBox () выведет на экран окно, содержащее сообщение, подождет, пока пользователь нажмет кнопку, и возвратит значение, указывающее, на какую кнопку нажал пользователь. Текст сообщения формируется первым аргументом функции, тип окна, число и названия кнопок — вторым аргументом, текст заголовка окна — третьим аргументом. Аргументы, заданные в примере, приводят к выводу окна, представленного на рис. 9

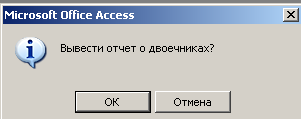


Рис.9.Окно сообщения, выводимое функцией MsgBox

ФункцияMsgBox возвращает значение 1 при нажатии в окне сообщения кнопки ОК и значение 2 при нажатии кнопки **Отмена** (Cancel).

**Управление отображением элемента в форме**

Пусть в форме Ведомость успеваемости при просмотре данных об оценках, полученных студентами, если в поле ОЦЕНКА обнаружено значение 2, то должен появляться текст Двоечник!.

1. Создайте в примечании подчиненной формы элемент управления Надпись со значением двоечник!. Предположим этот элемент имеет имя надпись 18.

Замечание

Только размещение надписи в примечании ленточной формы позволит отобразить ее однократно. Причем обеспечится связь ее с текущей записью ленточной формы

2. Свойству надписи **Вывод на экран** (Visible) присвойте значение**Нет** (No).

3. Создайте макрос, в котором свойству **Вывод на экран** (Visible) будет присваиваться значение Да (Yes) при просмотре записи с двойкой. Для этого включите в макрос столбец **Условие** (Condition). В этот столбец введите логическое выражение ОЦЕНКА=2. В столбце **Макрокоманда** (MacroCommand) выберите макрокоманду**ЗадатьЗначение** (SetVaiue). В строку аргумента **Элемент** (Item) введите ссылку на свойство **Visible** (ВыводНаЭкран) элемента управления надпись 18:

Надпись 18. Visible;

В строку аргумента **Выражение** (Expression) введите значение этого свойства — Да (Yes).

Заметим, что в ссылке Надпись18.visible нет необходимости указывать имя подчиненной формы, поскольку она является текущей при получении управления макросом.

4. Для того чтобы при переходе к записи с другой оценкой надпись двоечник! делалась невидимой, введите вторую макрокоманду**ЗадатьЗначение** (SetVaiue). В условии этой макрокоманды введите ОЦЕНКА<>2, а в аргументе **Выражение** (Expression) в качестве значения свойства **Visible** укажите**Нет** (No) (рис. 10).

5. Сохраните макрос под именем Сигнал.

6. Для связи макроса с подчиненной формой УСПЕВАЕМОСТЬ подчиненная форма в окне ее свойств на вкладке **События** (Event) в раскрывающемся списке **Текущая запись** (OnCurrent) выберите имя макроса Сигнал.

7. Проверьте работоспособность макроса. Откроите форму ведомость успеваемости и, просматривая записи, убедитесь, что если в поле ОЦЕНКА стоит 2, то появляется надпись Двоечник!.

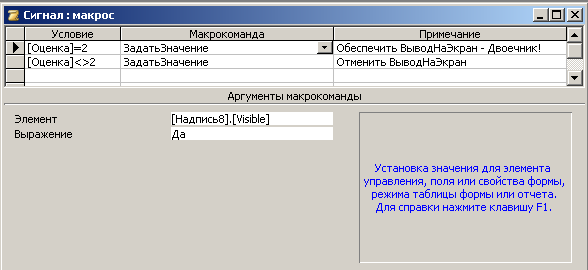


Рис. 10. Макрос, изменяющий значения свойств события

**Контрольные вопросы**

1. Что означает макрокоманда Выполнить Команду?
2. Что означает макрокоманда Свернуть?
3. Что означает макрокоманда На Запись?
4. Что означает макрокоманда Задать Значение?
5. Что означает макрокоманда Открыть Отчет?
6. Как проверить правильность работы макроса?
7. Как организовать диалог в макросе?
8. Текстовые константы, с которыми сравниваются введенные значения, заключаются в…?
9. Что делает функция MsgBox ()?
10. Как связать макрос с подчиненной формой?

# Практическая работа №11

**Создание процедуры поиска по первичному ключу**

**Цель работы: н**аучиться создавать процедуру обработки событий с помощью мастера, анализировать её*,* научиться корректировать процедуру, выполнять запросы в процедуре обработки события . создавать процедуру поиска по первичному ключу.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

Программы VВAмогут создаваться не только пользователем, но иавтоматически приработе мастеров Access. Рас­смотрим пример процедуры обработки события, созданной мастером кнопок.

Пусть в форме для просмотра информации о группе по желанию пользователя необходимо открывать список студентов текущей группы

**Практическая часть**

Создадим форму группа на основе одноименной таблицы, воспользовавшись режимом **Автоформа: в столбец** (AutoForm:Columnar) Создадим форму студент на основе одноименной таблицы, воспользовавшись режимом **Автоформа: ленточная** (AutoForm: Tabular).

Откроем форму группа врежиме конструктора и добавим в неёкнопку для открытия формы студент, воспользо­вавшись мастером. Для отображения в открываемой форме Студенттолько связанных записей необходимо при создании кнопки вобрать в окне мастера строку**Открыть форму для отобранных записей** (Opentheformandfindspecificdatafordisplay) и далее выбрать поле нг, по которому осуществляется связь форм и отбор записей подчи­ненной формы. Присвоим кнопке имя Список студентов (рис.1).

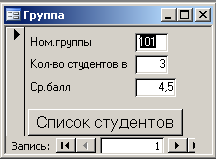


Рис. 1. Форма Группа с кнопкой для вызова формы Студент

Создавтакую кнопку, перейдем в режим формы и убедимся, что при щелчке мышью на этой кнопке открывается форма Студент, и в ней отображаются все записи, связанные с текущей записью формы Группа.При этом через форму можно произвести корректировку и добавление новых записей.

Рассмотрим процедуру, которую сформировал мастер для открытия формы при нажатии кнопки. Откроем форму группа в режиме конструктора. Установив курсор на кнопку список студентов, выберем в контекстном меню команду **Обработка событий** (BuildEvent). Открывается окно модуля формы (рис. 2), связанного с формой Группа.

В окнеотображается процедура обработки для события - щелчок мыши **(Click)** на кнопке Список студентов, подготовленная мастером.

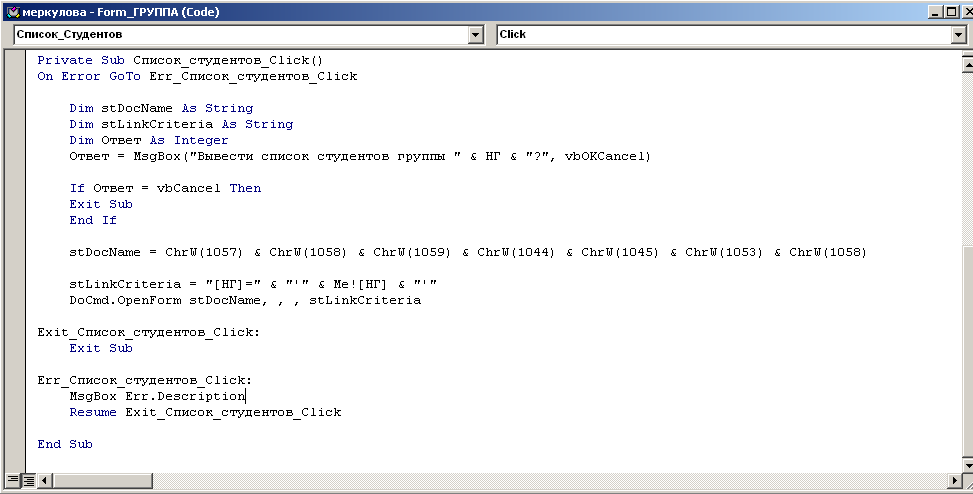


Рис. 2. Окно модуля формы Группа с процедурой обработки события Click для элемента список студентов

Процедура обработки события является личной процедурой типа Sub. Инструкция subобъявляет процедуру с со­ставным именем список\_студентов\_click(), включающим имя кнопки и имя события Пустые круглые скобки говорят об отсутствии аргументов.

Рассмотрим текст программы, записанный между инструкциями sub и EndSub.

Инструкция DoCmd.OpenFormstDocName…stLinkCriteria реализует главную функцию процедуры —открывает форму СТУДЕНТ.

DoCmd является объектом активного приложения Access, предназначенным для выполнения макрокоманд.OpenForm — метод, применяемый к данному объекту. Методу передаются два аргумента. Первый из передаваемых аргументов задает имя открываемой формы, второй передает строковое выражение, задающее условие отбора за­писей в открываемой форме. В данной инструкции для хранения значений аргументов используются перемен­ные строкового типа StDocName…stLinkCriteria.

В процедуре эти переменные ранее явно описываются инструкцией Dim.

**Dim**stDocName**As Siring**

**Dim**stLinkCriteria**As String**

**Для переменных указываются имена**stDocNameиSiLinkCriteriaи**строковый тип данных**String.

Прежде чем использовать описанные в процедуре переменные, им должны быть присвоены значения.

**Инструкции**stDocName = "Студент" и stLinkCriteria = "[НГ]=" &’’’’&Мe! [нг] &’’’ являются инструкциями присваи­вания, благодаря которым переменные получают конкретные значения. Первая переменная сохраняет имя фор­мы студент, вторая — выражение, указывающее, что связь таблиц осуществляется по полю нг и выборка данных в открываемой форме Студент должна производиться по значению этою поля в текущей записи формы группа. В процедурах VBA для ссылок на элементы управления текущей формы или отчета может быть использовано ключевое слово Me, заменяющее полный идентификатор текущего объекта — формы группа Forms! [группа]

Аргументы, записываемые в инструкции, разделяются запятыми. Два аргумента в инструкции DoCmdOpenForm ... были пропущены, запятые фиксируют их отсутствие. В конце инструкции отсутствие недостающих аргументов не помечается.

**Инструкция**OnErrorGoTo Егг\_Список\_студентов\_Сlick**позволяет при** возникновении ошибки передать управле­ние программе обработки ошибок.. Метка Егг\_список\_студентов\_click: определяет начало этой программы.

Программа обработки ошибок помещена между инструкциями ExitSub и EndSub, что позволяет отделить ее от части программы, соответствующей нормальному выполнению процедуры. В этой процедуре программа обра­ботки ошибок включает две инструкции.

MsgBox выводит стандартное сообщение об ошибке. Аргумент Err.Description определяет строковоезначение, которое возвращается при ошибке. Description является свойством объекта Error. Значением свойстваDescription является краткое предупреждающее сообщение с описаниемошибки.

Инструкция Resume указывает, с какой инструкции необходимо возобновить выполнение процедуры. В данномслучае — передает управление на метку, за которой стоит инструкция, осуществляющая выход из процедуры.

**Корректировка процедуры, созданной мастером**

Процедура, созданная мастером, может быть откорректирована и дополнена пользователем. Например, для под­тверждения необходимости вывода списка студентов текущей группы можно записать инструкции, организующие диалог, после описания строковых переменных (см. рис. 2).

Dim Ответ AsInteger

Ответ = MsgBox(" Вывести список студентов группы " &НГ& "?",vbOKCancel)

IfОтвет = vbCancelThen

ExitSub

EndIf

Первая инструкция определяет переменную Ответ для сохранения числового значения. Вторая присваивает этой переменной значение, возвращаемое функцией MsgBox. Функция MsgBox выводит в диалоговом окне (рис. 3) во­прос, включающий номер группы нг из текущей записи формы, и кнопки **ОК** и **Отмена** (Cancel). Возвращаемое функцией значение 1 или 2 соответствует нажатой кнопке.

Третья инструкция

If условие **Then**

[инструкция]

[Else

[инструкции] ]

EndIf

выполняет группу инструкций в зависимости от значения выражения (True или False), заданного в условии. Ес­ли была нажата кнопка Cancel, выполняется инструкция, завершающая процедуру по Exitsub, иначе выпол­нение процедуры продолжается с инструкции, следующей за Endif. Приставка указывает на использование зна­чений встроенных констант VBA.

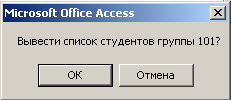


Рис. 3. Диалоговое окно функции MsgBox

Для открытия списка студентов только в режиме чтения достаточно дополнить инструкцию DoCmd. OpenForm… пятым аргументом acFormReadOnly.

Приставка ас указывает, что вкачестве аргумента используется значение встроенной константы Access.   
Заменитевпроцедуре ключевое слово Ме наForms! [Группа]иубедитесь, что процедура работает по-прежнему.  
Запишите вкачестве аргумента, задающегоусловие отбора, прямо винструкцию выражение   
“[нг]=”&’’’’&Me![нг]&’’’’.При этом можно удалить инструкцию описания переменной stLinkCriteriaиприсваи-  
ванияей значения.

**Выполнение запросов впроцедуре обработки события**

Создайте процедуру, которая подключается при закрытии формы Студент и, если список студентов групп сокра-  
щалсяили пополнялся новыми записями, обновляет число студентов вгруппах.

Для подсчета изменившегося числа студентов вгруппе иобновления поля кол втаблице ГРУППА воспользуйтесь   
созданными для этих целей запросами *Число студентоввгруппахиОбновление ГРУППА\_КОЛ*.

Кроме того, поскольку для некоторых групп могут быть уничтожены все записи остудентах, адва предыдущих   
запроса не решают задачу обновления поля кол, вэтом случае потребуется запрос обнуление ГРУППА\_КОЛбезпод-  
чиненных вСТУДЕНТ (рис 4).

Выполнение перечисленных запросов впроцедуре позволит автоматически поддерживать правильные значения в  
таблице группа и освободит пользователя от необходимости помнить ивручную выполнять разрозненные дейст-  
вия. Впроцедуре для выполнения запросов может быть использован объектDoCmdиего метод OpenQuery.

Поскольку при работе сформой может ине производиться изменения числа записей остудентах, предусмотрите   
впроцедуре диалоговый режим спользователем, врезультате которого определитесь, нужно ли выполнять обнов-  
лениетаблицы группа. Для этого используйте функцию MsgBox,которая выводит вдиалоговом окне текст, задан-  
ный ее первым аргументом, атакже определяемые вторым аргументом кнопки, например, **Да**и**Нет**(рис. 5)   
и, как любая функция, возвращает значение.

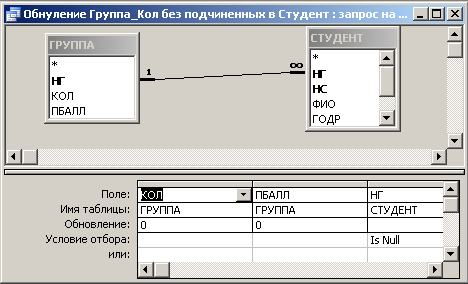


Рис.4. Запрос на обновление полей втаблице Группа для записей, не

имеющих подчиненных

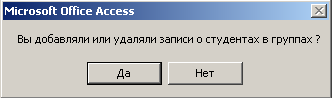


Рис. 5. Диалоговое окно функции MsgBox

Для анализа возвращенного функцией значения ипринятия решениядальнейшихдействияхиспользуйте инструкцию условного перехода.

Ifусловие Then

[инструкции]

[Else

[инструкции]]

1ndif

**о**

Эта инструкция взависимости от значения выражения, заданного вусловии (Trueили False), выполняет ту или иную группу инструкций. Для создания процедуры выполните следующие действия:

l.Откройте форму студент врежиме конструктора.

2. Откройте окно свойств формы на вкладке **События** (Events). Враскрывающемся списке **Закрытие** (Close) выберите значение **[Процедура обработки события]**ищелкните по значку построителя вконце строки. Таким образом, связывается выполнение создаваемой процедуры сзакрытием формы. Воткрывшемся окне модуля формы вшаблон процедуры обработки события PrivateSubForm\_Close() (рис 6) запишите инструкции процедуры.

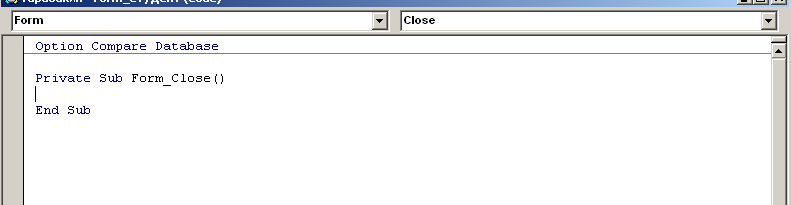


Рис. 6. Инструкции начала иконца процедуры обработки события Closeдля формы Студент

3. Для определения переменных, сохраняющих строковые значения имен запросов ичисловое значение, воз-  
вращаемое функцией MsgBox, запишите инструкциюDim.

DimstDocName,stDocNamel,stDocName2AsString, ОтветAsInteger

1. Запишите инструкцию присваивания переменной ответ значения, возвращаемого функцией MsgBox.

Ответ =MsgBox("Выдобавляли или удаляли записи остудентах вгруппах?", vbYesNo)

Приставка vbвзначении второго аргумента указывает на использование значения встроенной константы VBA, определяющего вывод вдиалоговом окне кнопок**Да**и**Нет.**

1. Для проверки, нажата ли пользователем вдиалоговом окне кнопкаНет ивыполнения вэтом случае инст-  
   рукции выхода из процедуры ExitSub, запишите инструкцию условного перехода, опустив необязательную ее часть.

IfОтвет=vbNoThen

ExitSub

EndIf

Если переменная Ответ получила значение, соответствующее нажатию кнопки**Нет,** выполняется инструкция, следующих за Thenизавершающая процедуру по ExitSub, иначе выполнение процедуры продолжается син-  
струкции, следующей за Endif.

1. Для обновления поля кол втаблице группа запишите инструкции выполнения перечисленных выше запросов. Используйте объект DoCmdиприменяемый кданному объекту метод openQuery.Аргументом этого метода яв-  
   ляетсяимя выполняемого запроса. Если для сохранения имен запросов предусмотрены переменные, им, пре-  
   ждечем они будут использованы, должны быть присвоены эти имена.

StDocName="Числoстудентов вгруппах”

stDocNamel="06нoвлениеГРУППА\_КОЛ"

stDocName2="Обнуление ГРУППА\_КОЛ без подчиненных вСТУДЕНТ"

DoCmd.SetWarningsFalse

DoCmd.OpenQuery(stDocName)

DoCmd.OpenQuery(stDocNamel)

DoCmd.OpenQuery(stDocName2)

DoCmd.SetWarningsTrue

Метод Setwarningsсаргументом Falseпозволяет отключить вывод стандартных предупреждений на экран во время выполнения запросов. После выполнения запросов этот режим восстанавливается.

В окончательном виде процедураможет иметь вид, представленный на рис. 7.

7. Закройте окно редактора VBAивозвратитесь вAccess, выбрав команду **File|CloseandReturntoMicrosoftAccess.**

8.Откройте форму Студент. Добавьте или удалите несколько записей остудентахОткройте таблицу группа иубе-  
дитесь, что ваша процедура работает.

9.Измените инструкцию DoCmd,заменив вней имена переменных на имена запросов. Удалите из инструкции описания переменные stDocName,stDocNamelиstDocName2иинструкции присваивания им значений. Убе-  
дитесь, что ивэтом случае процедура работает правильно.

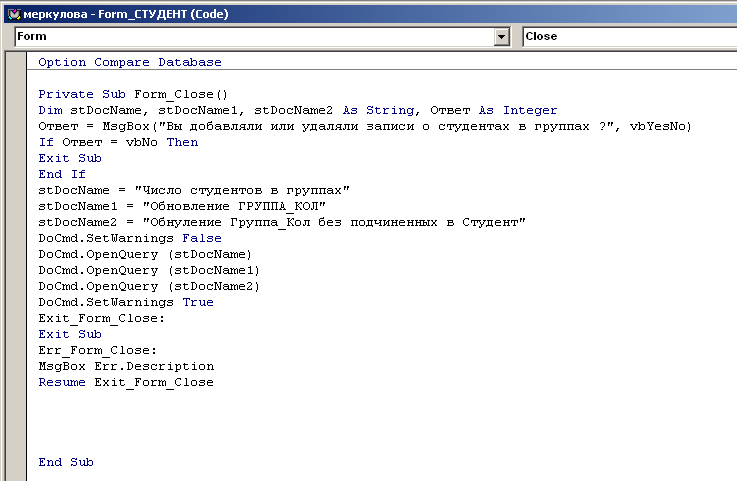


Рис. 7. Процедура обработки события (Close) для формы Студент

**Поиск данных сиспользованиеминдексов**

Рассмотрим, как найти данные остуденте втаблице базы данных Учебный процесс по первичному ключу. На-  
помним, что таблица СТУДЕНТ имеет составной ключ, включающий поля, номер группы нг иномер студента в  
группе нс. Для выполнения процедур поиска воспользуемся кнопочной формой поиск..Создадим эту форму в  
режиме конструктора форм, не выбирая для нее источника данных. Создадим кнопку для запуска процедуры,   
используя панель элементов при отключенном мастере.Врежиме формы она может иметь вид, представ-  
ленный на рис. 8.

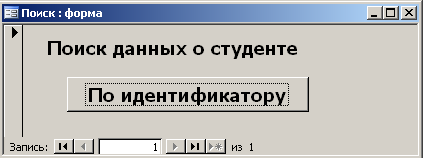
****

Рис. 8. Кнопочная форма для запуска процедуры

Для быстрого поиска записи по первичному составному ключу используйте метод Seekобъекта DAO.Recordset.  
Объект DAO.Recordset представляет набор записей, который обеспечивает доступ кего записям иобработку ка-  
ждойиз них. Длясоздания набора записей используйте метод openRecordsetобъекта Database(база данных).

Метод Seekприменяется киндексированному набору записей —объекту DAO.Recordset,источником записей   
для которого является таблица. Он находит запись, удовлетворяющую заданным условиям для текущего индек-  
са, иделает эту запись текущей. Текущий индекс указывается спомощью свойства Indexобъекта DAO.Recordsetпе-  
редвызовом метода Seek..

Для выполнения сформулированной задачи впроцедуре понадобятся: объектная переменная типа Databaseдля   
хранения ссылки на объект, представляющий базу данных, итипа DAO.Recordset—для хранения ссылки на   
объект, представляющий набор записей. Синтаксис методов SeekиOpenRecordsetтребует использования   
объектной переменной для представления существующего табличного Объекта DAO.Recordsetи объекта Database.

**Процедура поиска по первичному ключу**

Процедура поиска по первичному ключу может быть записана следующим   
образом:

PrivateSubКнопка\_Сliск()

* Определение объектных переменных

Dim dbsУчебный\_пpoцeccAs Database

Dim rstСтудентAs DAO.Recordset

* Определение строковых переменных.

DimstrNGAsString

DimstrNSAsString

* Запись вобъектную переменную ссылки на текущую БД

SetdbsУчебный\_процесс =CurrentDb()

* Открытие набора записей для таблицы СТУДЕНТ изапись ссылки на него вобъектную переменную rstСтудент

SetrstСтудент=dbsУчебный\_процесс**.**openRecordset**\_**(" СТУДЕНТ ",dbopenTable)

* + Задает индекс сименем PrimaryKey,построенный по ключу таблицы НГ+НС;этот индекс будет использован при поиске.

rstСтудент.Index="PrimaryKey"

* + Присваивает строковым переменным значения, задаваемые пользователем вокне ввода параметров

strNG=InputBox("Введиномер группы", "Ввод параметров поиска")

strNS=InputBox("Введиномер студента", "Ввод параметров поиска")

* + Метод Seek, примененный кнабору записей, отыскивает иделает текущей запись суказываемыми варгументах значениями первичного составного ключа

rstСтудент.Seek"=", strNG, strNS

* + Инструкция условного перехода проверяет наличие записи суказанным ключом.

IfrstСтудент.NoMatchThen

* + Вывод сообщения изавершение процедуры

MsgBox"Идентификатор не найден!"

ЕхitSub

EndIf

* + Вывод данных из указанных полей найденной записи

MsgBox"Студент-" &rstСтудент! [Фио]**&", "&**rstСтудент![ГОДР]&\_"года рождения", vbОKonly,\_"Данные из записи, найденные методом Seek"

EndSub

Рассмотрим более подробно объяснение используемых впроцедуре инструкций.

1. Откройте окно модуля формы, щелкнув на кнопке панели инструментов **Программа** (Code). Форма должна быть закрытаили находиться врежиме конструктора.

2. Вшаблон процедуры обработки события click(нажатие кнопки) для **кнопки поиски по первичному ключу (идентификатору)** PrivateSubКнопка\_click()запишите приведенные выше инструкции процедуры.

3. Приведенные ниже инструкции описывают объектные переменные для хранения ссылок на базу данных инабор записей.

DimdbsУчебный\_процессAsDatabase

DimrstСтудентAsDAO.Recordset

4. Для присваивания объектной переменной ссылки используется инструкция *Set.* Приведенная ниже инст-  
рукция присваивает объектной переменной типа Databaseзначение ссылки на текущую базу данных.

SetdbsУчебный\_процесс=**CurrentDb()**

5. Для поиска записи втаблице Студент сиспользованием индекса создайте на ее основе набор записей. Ис-  
пользуйте для этого метод OpenRecordsetобъекта Databaseисохраните ссылку на этот объект вобъектной переменной rstСтудент.

SetrstСтудент=dbsУчебный\_процесс.OpenRecordset("СТУДЕНТ",dbOpenTable)

6*.* Прежде чем использовать метод Seek, присвойте свойству indexнабора записей имя первичного индекса таблицы студент. Метод seekбудет производить поиск записи, используя этот индекс. Напомним: ин-  
декссименем PrimaryKeyпостроен по ключу таблицы нг +нс.

rstСтудент.Index="PrimaryKey"

7. Для ввода параметров поиска нг инс используйте функцию InputBox,которая выводит на экран диалого-  
воеокно с полем ввода иорганизует ожидание ввода текста. При нажатии пользователем кнопки ОКфункция возвращает содержимое поля ввода. Присвойте введенное пользователем значение параметра стро-  
ковым переменным strNGиstrNS:

strNG=InputBox("Введиномер группы", "Ввод параметров поиска")

strNS=InputBox("Введиномер студента", "Ввод параметров поиска")



Рис 9. Окна ввода значений для строковых переменных

Формируемыефункцией inputBoxокна ввода значений для строковых переменных приведены на рис.9.   
Первый аргумент функции inputBoxопределяет текст, отображаемый вдиалоговом окне, второй аргумент-  
текст, отображаемый взаголовке диалогового окна, а третий аргумент—текст, отображаемый вполе ввода (в   
примере не задан).

8. Для поиска записи используйте метод Seek, примененный кнабору записей rstСтудент. Этот метод на-  
ходитзапись, вкоторой значения первичного составного ключа равны указанным варгументах значениям,   
иделает ее текущей

rstСтудент.Seek"=", strNG,strNS

9. Для проверки результата поиска используйте свойство NoMatchнабора записей rstСтудент. Свойство приоб-  
ретаетзначение True, если запись сзаданным ключом не найдена, иFalse—впротивном случае. Использо-  
ваниезначения этого свойства винструкции условного перехода if.. Then... Endifпозволяет направить   
выполнение процедуры по различным ветвям.

10. Для вывода данных из полей найденной записи используйте инструкцию MsgBox,результаты выполнения   
которой приведены на рис. 10.

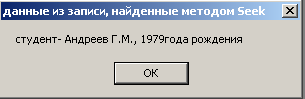


Рис. 10. Результат поиска по идентификатору

**Контрольные вопросы**

1. Что такое программа VBA?
2. Для отображения в открываемой форме только связанных записей необходимо…?
3. Программа обработки ошибок помещена…?
4. На что указывает инструкция Resume?
5. На что указывает приставка ас?
6. Для анализа возвращенного функцией значения ипринятия решениядальнейшихдействиях, какую необходимо использовать инструкцию?
7. Что позволяет метод Setwarningsсаргументом False?
8. Что такое индекс?
9. Для быстрого поиска записи по первичному составному ключу используйте…?
10. Когда используется свойство NoMatch?

# Практическая работа №12

**Создание формы средствами VBA.**

**Цель работы:** научиться создавать процедуру для создания формы средствами VBA на основе таблицы, размещения в ней элементов управления, научиться размещать процедуру в стандартном модуле, организовывать её выполнение.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

Создайте в базе данных Учебный процесс средствами VBA новую форму на основе таблицы ГРУППА, отображающую содержимое поля нг (номер группы).

Новая форма может быть создана в процедуре Access VBA методом CreateForm объекта Application, который возвращает объект Form. Метод CreateForm открывает пустую новую форму в режиме конструктора.

**Практическая часть**

Чтобы указать источник данных новой формы, используется свойство формы RecordSource В качестве значения свойства может быть использовано имя таблицы, запроса или SQL-оператор. Присваиваемое свойству значение должно иметь текстовый (string) тип данных

Для размещения в форме различных элементов управления используется методCreateControl, с помощью которого можно создать и поле (элемент **TextBox**) с присоединенной надписью (элемент **Label**). Аргументы этого метода указывают:

• первый — имя открытой формы или отчета, в котором создается элемент управления,

**•** второй — тип создаваемого элемента управления;

**•** третий — область формы, в которой следует разместить элемент; по умолчанию это область данных (**Detail section**);

**•** четвертый — имя родительского элемента, к которому присоединяется создаваемый;

**•** пятый — имя поля, которое должно отображаться элементом управления **TextBox**, или подпись для элемента **Lаbеl**;

**•** шестой — координаты размещения левого верхнего угла элемента,

**•** седьмой — ширину и высоту элемента.

Для записи аргументов используются константы Access.

Перед созданием элемента управления переменным intLabelX, intLabelY intDataX, intDataY   
присваиваются значения, которые определяют координаты его размещения.

Ниже приведены инструкции процедуры, в которой создается форма со стандартным именем и единственным полем

Sub НоваяФорма ()

* + Определение объектной переменной для формы

Dim frmAs Form

* + Создание новой формы и сохранение ссыпки на нее в объектной переменной

Set frm = CreateForm

* + Определение в качестве источника данных формы таблицы ГГРППА

frm.RecordSource = "ГРУППА"

Определение объектных переменных для поля и надписи в форме   
Dim ctlLabel As Control, ctlText As Control

* + Определение переменных для координат размещения элементов

Dim intDataX As Integer, intDataY As Integer

Dim intLabelX As Integer, intLabelY As Integer

* + Присваивание значений координатам размещения элементов управления форма

intLabelX = 100

intLabelY =100

intDataX = 100

intDataY = 500

* + Создание в форме элемента управления — поле НГ по умолчанию в разделе данных и стандартного размера

Set ctlText = CreateControl(frm.Name, acTextBox,, "", "НГ",intDataX, intDataY)

* + Создание элемента управления — надпись поля НГ

setctlLabel = CreateControl(frm.Name, acLabel,,ctlText.Name,"Номергруппы",intLabelX, intLabelY)

* + Восстановить форму из свернутого состояния

DoCmd.Restore

End Sub

Дополните процедуру следующими инструкциями:

Эта инструкция выведет в окно отладки имя вновь созданной формы   
Debug.Print frm.Name

Изменение имени поля (по умолчанию поле имеет имя поле)

сtlText.Name = "полюшко"

Эта инструкция выведет в окно Immediate Windows имя поля

Debug. Print ctlText. Name

Закрытие формы с сохранением

DoCmdCloseacForm, frm.Name, acSaveYes

Разместите процедуру в стандартном модуле. Для выполнения активной процедуры выберите в редакторе Visual Basic команду Run Sub/UserForm в меню **Run**. В противном случае выберите команду **RunMacro | Run**. При выполнении последней команды отобразится окно с полным списком процедур всех стандартных модулей проекта. Выберите процедуру **Новая Форма** и выполните ее.   
Напомним, что окно отладки **Immediate** открывается в редакторе Visual Basic командой **View | ImmediateWindow.**

Перейдите в окно Access, найдите форму в окне базы данных, откройте ее и убедитесь, что процедура работает правильно.

**Контрольные вопросы**

* 1. Как создать новую форму с помощью VBA?
  2. Чтобы указать источник данных новой формы, используется…?
  3. Какой используется метод для размещения в форме различных элементов управления?
  4. Что используется для записи аргументов?
  5. Перед созданием элемента управлениячто делается с переменными?
  6. Как записывается инструкция процедуры Создание новой формы и сохранение ссылки на нее в объектной переменной?
  7. Как записывается инструкция процедурыВосстановить форму из свернутого состояния?
  8. Как записывается инструкция процедурыЗакрытие формы с сохранением?
  9. Для выполнения активной процедуры необходимо?
  10. Как проверить правильность работы процедуры?

# **Практическая работа №13**

Создание формы средствами **VBA.**

Цель работы: научиться создавать процедуру для создания таблицы средствами VBA, определять структуру таблицы, состав полей и первичный ключ.

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

**Теоретическая часть**

Создайте в базе данных Учебный процесс средствами VBA новую таблицу Деканат с полями код и название. В качестве первичного ключа таблицы используйте поле код. Для проверки работоспособности программы оформите ее, как процедуру обработки события — нажатие кнопки в некоторой форме.

**Практическая часть**

Для работы со структурой таблицы используйте объекты Database, TableDef, Field и index. Объекты Database и TableDef представляют, соответственно, базу данных и структуру таблицы. Объекты Field и index позволяют определить, соответственно, поле таблицы и ключ таблицы.

1. В процедуре создайте объектные переменные типа Database, TableDef, Field и index, предназначенные для хранения ссылок на соответствующие объекты базы данных.

Dim dbs Учебныйпроцесс As Database

Dim tbl As TableDef

Dim fiel As Field

DimindlAsIndex

2. Для создания нового объекта TableDef — таблицы с именем Деканат, примените метод CreateTableDef к объектной переменной dbsУчебный процесс, которая указывает на объект Database. Для создания новой объекта Field — текстового поля с именем Код, сохраняющего не более двух символов, примените метод createField к объектной переменной tbl, которая указывает на объект TableDef:

Setdbs Учебный процесс = CurrentDb ()

Settbl = dbsУчебный процесс. CreateTableDef («Деканат»)

Set fiel = tbl.CreateField(«Код», dbText, 2)

3. Для добавления поля (ссылка на него сохранена в объектной переменной fiel) в семейство Fields, входящее в объект TableDef, используйте метод Append и запишите:

tbl.Fields.Appendfiel

Обратите внимание, что перед именем семейства, так же, как перед именем метода, который выполняется для указанного перед этим объекта, ставится точка.

4. Для создания в таблице второго поля название используйте ту же объектную переменную fiel и также добавьте его в семейство Fields:

Set fiel = tbl.CreateField ("Название", ' dbText, 15)

tbl.Fields.Appendfiel

5. Для добавления таблицы — объекта TableDef (ссылка на него сохранена в объектной переменной tbl) в семейство TableDefs используйте метод этого семейства Append:

dbsУчебныйпроцесс. TableDef.Appendtbl

6. Теперь создайте первичный ключ новой таблицы:

• Прежде всего, создайте индекс, который будет использоваться в качестве первичного ключа, задав ему имя, например, то, которое по умолчанию используется при определении ключа таблицы в конструк­торе — PrimaryKey. Для этого используйте метод Createlndex объекта TableDef:

Setindl = tbl.Createlndex("PrimaryKey")

• Следующей инструкцией задайте поле, на основе которого строится индекс (поле код) indl.createField("Код"), и включите его в семейство полей индекса. В модели объектов доступа к данным поля индекса включаются в семейство Fields, которое включается не только в объект TableDef, но и в объект Index.

indl.Fields. Append indl.CreateField("Код")

• Объявите индекс первичным ключом таблицы деканат. Для этого присвойте свойству индекса Primary значение True:

indl.Primary = True

• Добавьте индекс (ссылка на него сохранена в объектной переменной indl) в семейство индексов объекта TableDef:

tbl.Indexes.Appendindl

7. Окончательно процедура запишется следующем виде:

Private Sub Кнопка\_Сliск ()

On Error GoTo Err\_Кнопка\_Сliск

Dim dbs Учебныйпроцесс As Database

' Определение объектных переменных для таблицы и поля

Dim tbl As TableDef

Dim fiel As Field

Dim indl As Index

Set dbs Учебныйпроцесс = CurrentDb()

' Создание нового объекта — таблицы с именем Деканат и присваивание объектной переменной ссылки на него

Settbl = dbs Учебный процесс. СrеаtеТаb1еDеf ("Деканат")

' Создание текстового поля с именем Код и длиной в 2 символа, присваивание объектной переменной ссылки на него

Set fiel = tbl.CreateField("Koд", dbTеxt, 2)

' Добавление поля в семейство Fields объекта TableDef с помощью метода Append

tbl.Fields.Appendfiel

' Создание второго поля в таблице

Set fiel = tbl.CreateField ("Название",dbText, 15)

tbl. Fields. Append fiel

' Добавлениетаблицывсемейство TablеDefs спомощьюметода Append

dbsУчебныйпроцесс.TableDefs.Append tbl

' Создание для новой таблицы индекса с именем PrimaryKey

Set ind1 - tbl. Createlndex("PrimaryKey")

' Определение поля, составляющего индекс (поле Код), идобавление его в семейство полей индекса

ind1 .Fields. Append ind1 .CreatеFiеld("Koд")

' определение индекса как первичного ключа для таблицы Деканат

ind1 .Primary = True

' добавление индекса в семейство индексов tbl. Indexes. Append ind1

Exit\_Кнопка\_Click:

Exit Sub

Еrr\_Кнопка\_Сliск;

MsgBox Err.Description

Resume Exit\_Кнопка\_Сlick

EndSub

8. Возвратитесь в Access и, нажав в форме кнопку, к которой привязана процедура обработки этого события, выполните процедуру. В БД создается таблица деканат с полями код и название, причем поле код объявлено ключевым, а индекс имеет имя PrimaryKey. Чтобы убедиться в этом, откройте таблицу в режиме конструктора.

1. Выполните процедуру повторно, и будет выведено сообщение, представленное на рис. 1

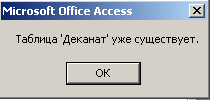


Рис. 1. Сообщение, формируемое инструкцией обработки ошибки

Это сообщение является результатом обработки ошибки, которая производится за счет инструкций в начале процедуры:

On Error GoTo Еrr\_Кнопка\_\_Сliск

ивконцепроцедуры:

Еrr\_Кнопка\_Сliск: MsgBox Err.Description

Resume Exit\_Кнопка\_Click

Аргумент Err.Description инструкции MsgBox определяетстроковоезначение, формируемоесвойством Description, объекта Err. Значением свойства является предупреждающее сообщение с описанием перехваченной ошибки.

**Контрольные вопросы**

1. Что такое первичный ключ?
2. Что представляют объекты Database и TableDef?
3. Что позволяют определить объекты Field и index?
4. Что такое индекс?
5. Для добавления поля в семейство Fields, входящее в объект TableDef, что используется?
6. Для создания нового объекта TableDef, применяется метод?
7. Что ставится перед именем семейства, так же, как перед именем метода, который выполняется для указанного перед этим объекта?
8. Как объявить индекс первичным ключом таблицы?
9. Что определяет аргумент Err.Description инструкции MsgBox?
10. Как проверить работу процедур?

# **Практическая работа №14**

Работа с базой данных в многопользовательском режиме.

Цель работы: научиться выбирать режим доступа к общей базе данных, устанавливать режим доступа по умолчанию, задавать параметры блокировки записей

***Студент должен:***

***уметь:***

* проектировать реляционную базу данных;
* использовать язык запросов для программного извлечения сведений из баз данных;

***знать:***

* основы теории баз данных; модели данных;
* особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;
* основы реляционной алгебры;
* принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;

Теоретическая часть

Выбор режима доступа к общей базе данных

Общая база данных может открываться в монопольном режиме или в режиме общего доступа в соответствии с установленными параметрами, умолчанию используется режим, заданный параметрами настройки. Первоначально параметры настройки определяют открытие базы данных в режиме общего доступа

При необходимости использовать базу данных в режиме, отличном от установленного по умолчанию, нужно открыть базу данных, используя команду **Файл | Открыть**(File|OpenDatabase) или выбрав в области задач строку **Другие файлы**. Для установки режима доступа к базе данных необходимо в окне **Открытие файла базы данных**(Open) выбрать файл базы данных в списке кнопки**Открыть**— нужный режим (рис. 1). Представленные в списке режимы позволяют:

* **Открыть** (Open) — открыть, базу данных в режиме общего доступа пользователей, которые смогут как читать, так и сохранять данные и объекты в ней
* **Открыть дли чтения**(OpenRead-Only) — открыть базу данных в режиме общего доступа только для чтения. При этом невозможно сохранение изменений в данных или описаниях объектов;
* **Монопольно** (OpenExclusive) — открыть базу данных для работы только этого пользователя и предотвратить открытие ее другими пользователями;
* **Монопольно для чтения**(OpenExclusiveRead-Only) — открыть базу данных только для чтения и только для работы одного пользователя.

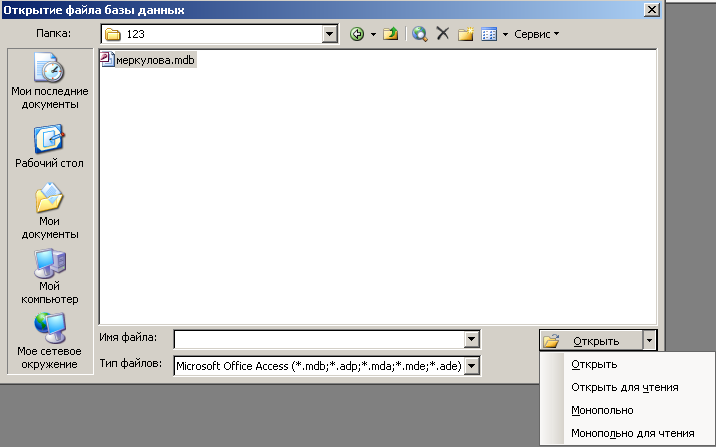


Рис. 1. Выбор режима доступа к базе данных при работе в сети

**Практическая часть**

Режим монопольного доступа

Для того чтобы открыть базу данных для монопольного доступа, выберем в меню кнопки**Открыть**(Open) команду **Монопольно**(Exclusive).

Режим монопольного доступа запрещает другим пользователям и программам доступ к этой базе данных. База данных, открытая одним пользователем в монопольном режиме, не доступна для открытия другими пользователями.

Для общей базы данных должен устанавливаться монопольный доступ в том случае, если необходимо провести такие работы, как изменение структуры БД, ввод большого объема данных, сжатие или восстановление базы данных после повреждения. Монопольный доступ, как правило, устанавливается администратором базы данных.

Чтобы многие пользователи могли одновременно работать с приложением, ни один из них не должен открывать базу данных в режиме монопольного доступа.

**Режим общего доступа**

Для того чтобы открыть базу данных в режиме общего доступа, выберите в меню кнопкиОткрыть (Open) режим Открыть (Open). Этот же режим выбирается, если не открывая списка, нажать кнопкуОткрыть.

Режим общего доступа обеспечивает коллективное использование объектов базы данных как в режиме чтения, так и в режиме обновления. Многие пользователи могут одновременно открывать одну базу данных в режиме общего доступа. Причем для некоторых из них доступ может быть ограничен только чтением.

Для того чтобы открыть базу данных в режиме только для чтения, необходимо выбрать режимОткрыть для чтения (OpenRead-Only) или Монопольно для чтения (OpenExclusiveRead-Only). При открытии базы данных в этом режиме выдается соответствующее сообщение.

Откроите базу данных Учебный процесс в режиме только для чтения и впишите в отчет текст полученного сообщения.

В этом режиме недоступны все операции по изменению объектов базы данных. Например, в открытой таблице базы данных нет строки новой записи, заблокированы изменения в полях, а в меню Правка (Edit) не видны строки удаления записей.

Если база данных открыта в режиме общего доступа даже только для чтения, другие пользователи не могут открыть ее для монопольного использования. При попытке пользователя открыть базу данных в монопольном режиме ему предоставляется только режим общего доступа. При этом выдается соответствующее сообщение.

Откройте базу данных Учебный процесс в режиме монопольного доступа с другого компьютера и впишите в отчет текст полученного сообщения.

Монопольный доступ предоставляется пользователю только в случае, если с базой данных не работает ни один из пользователей. Если некоторому пользователю предоставлен монопольный доступ к базе данных, другие пользователи не смогут открыть эту же базу данных для монопольного доступа.

Вне зависимости от режима доступа, при изменении пользователем структуры таблицы другие пользователи не смогут получить доступ к ее данным. И наоборот, если хотя бы один пользователь открыл таблицу, никто не сможет изменить ее структуру.

**Установка режима доступа по умолчанию**

Режим доступа, используемый при открытии базы данных по умолчанию, устанавливается в окне Параметры (Options) на вкладке Другие (Advanced) (рис. 2). Окно открывается командой меню Сервис | Параметры (Tools | Options).

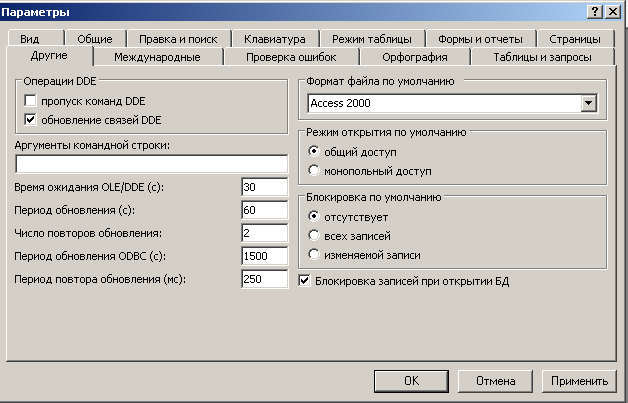


Рис. 2. Установка параметров открытия базы данных коллективного доступа, используемых по умолчанию используемых по умолчанию

Для того чтобы база данных по умолчанию открывалась для коллективного доступа, параметр **Режим открытия по умолчанию**(DefaultOpenMode) должен иметь значение общий доступ (Shared).

Заданные параметры вступят в силу до закрытия диалогового окна, если нажать кнопкуПрименить (Apply). Для применения заданных параметров и закрытия диалогового окна нажмите кнопку ОК.

**Замечание**

Параметры настройки сохраняются не в базе данных, а в файле рабочей группы пользователя. После изменения параметров настройки они используются с каждой базой данных, которая открывается или создается любым пользователем из этой рабочей группы.

Вне зависимости от установленного по умолчанию режима открытия базы данных можно в окне **Открытие файла базы данных**(Open) выбран, другой режим. Например, если по умолчанию установлен режим общего доступа, базу данных можно открывать для монопольного доступа или только для чтения.

Задание параметров блокировки записи

Параметры блокировки записей устанавливаются пользователем, макросом или программой VisualBasic в свойствах форм, отчетов и запросов. Это свойство представлено раскрывающимся списком **Блокировка записей**(RecordLocks) в окне свойств формы на вкладке **Данные**(Data), в окне свойств запроса па вкладке **Общие** (General), в окне свойств отчета на вкладке **Другие**(Other).

В формах это свойство определяет способы блокировки записей в базовой таблице при обновлении содержимого общей базы данных, а также в результирующей таблице запроса, на котором построена форма.

В отчетах это свойство определяет, блокируются ли записи в базовой таблице или запросе, на которых построен отчет, при печати или просмотре отчета.

В запросах это свойство определяет, блокируются ли записи в таблице запроса при выполнении запроса на добавление, на удаление, на создание таблицы или па обновление.

Свойство **Блокировка записей**(RecordLocks) определяет способы блокировки записей и их реализацию при попытке нескольких пользователей одновременно изменить одну и ту же запись. Когда один пользователь изменяет запись, Access может автоматически заблокировать ее, чтобы запретить другим пользователям изменять эту запись до завершения работы с ней первого пользователя.

Свойство **Блокировка записей**(RecordLocks) может принимать одно из трех значений.

* **Отсутствует**(OnLocks) (значение по умолчанию). В отчетах записи не блокируются при выводе па просмотр или на печать. В запросах записи не блокируются при выполнении запроса па изменение. В формах допускается одновременное изменение записи двумя или несколькими пользо­вателями. Пели два пользователя пытаются сохранить изменения одной записи, Access выводит предупреждающее сообщение тому из пользователей, кто сохраняет запись вторым. Этот пользователь может сохранить свой вариант записи, уничтожив изменения, внесенные другим пользователем; скопировать свою измененную запись в буфер обмена или отказаться от собственных изменений. Данное значение позволяет изменять записи в любое время, но при внесении изменений разными пользователями могут возникать конфликты.
* **Всех записей**(AllRecords). Блокируются все записи в базовой таблице или запросе при открытии формы в режиме формы или в режиме таблицы, во время просмотра или печати отчета и при выполнении запроса. Другим пользователям разрешается просматривать записи, но они не могут изменять, добавлять или удалять любые записи до закрытия первым пользователем формы, до завершения этим пользователем печати отчета или до завершения запроса, в которых установлено значение блокировки записей. Блокировка всех записей является блокировкой таблицы.
* **Изменяемой записи**(EditedRecord). Это значение свойства **Блокировка записей**(RecordLocks) может быть установлено только в формах и запросах. При этом может блокироваться страница записей. Страница — это блок данных в файле базы размером в 4 Кбайт, который может содержать одну или несколько записей. Как только любой из пользователей начинает вносить изменения в любое ноле одной из записей страницы, страница блокируется и остается заблокированной до тех пор, пока запись не будет сохранена. Дополнительно к блокировке страничного уровня Access поддерживает блокировку на уровне единственной записи. Выбор режима блокировки на уровне записи в текущей базе данных производится параметром **Блокировка записей при открытии базы данных**(Opendatabasesusingrecordlevellocking) (команда **Сервис | Параметры | Другие**(Tools | Options | Advanced)). Этот вид блокировки накладывает минимальные ограничения и позволяет многим пользователям работать в режиме обновления данных. Если этот параметр не выбран, по умолчанию будет использована блокировка на уровне страницы.

В режиме формы и режиме таблицы при попытке изменить значения в заблокированной другим пользователем записи в области выделения записи отображается символ блокировки.

Для задания режима блокировки записей для конкретной формы, запроса или отчета необходимо открыть их в режиме конструктора, вывести окно свойств и установить требуемое значение свойства **Блокировка записей**(RecordLocks).

Задание параметров блокировки записи, используемых по умолчанию

Если пользователь не определил значения свойства **Блокировка записей**(RecordLocks) в объектах базы данных, то используются параметры по умолчанию. Для того чтобы просмотреть и, если необходимо, изменить эти параметры, нужно в меню **Сервис**(Tools) выбрать команду **Параметры**(Options). В окне **Параметры**(Options) следует выбрать вкладку **Другие**(Advanced) и на ней — один из параметров в группе параметров **Блокировка по умолчанию**(DefaultRecordLocking) (см. рис. 2).

Заметим, что команда **Параметры**(Options) появляется в меню **Сервис** (Tools) только при открытой базе данных.

Автоматическое обновление отображения записей

ВAccess предусматривается автоматическое обновление через определенные промежутки времени отображения записей, просматриваемых в форме или другом объекте в режиме таблицы. Частота обновления определяется настройкой интервала обновления. Для изменения этой величины следует воспользоваться командой **Сервис | Параметры**(Tools | Options) и ввести новое значение параметра **Период обновления**(RefreshInterval) на вкладке Другие (Advanced) (см. рис. 2).

Перечислим параметры, которые регулируют процесс обновления отображения данных на экране и способствуют разрешению конфликтных ситуаций.

* **Период обновления**(RefreshInterval). Этот параметр задает число секунд, по истечении которых Access автоматически обновляет записи в объекте в режиме таблицы или в форме. Допустимыми являются значения от 1 до 32 766 секунд. При обновлении текущего объекта в режиме таблицы или формы записи не сортируются заново, не добавляются новые записи и не убираются удаленные записи. Для того чтобы увидеть эти изменения, необходимо выполнить операцию обновления данных.
* **Период повтора обновления**(UpdateRetryInterval). Этот параметр задает число миллисекунд, по истечении которых Access автоматически пытается сохранить измененную запись, заблокированную другим пользователем. Допустимыми являются значения от 0 до 1000 мс.
* **Число повторов обновления**(NumberofUpdateRetries). Этот параметр задает число попыток сохранения измененной записи, заблокированной другим пользователем. Допустимыми являются значения от 0 до 10.

Измените величину частоты автоматического обновления записей. Для этого воспользоваться командой **Сервис | Параметры** (Tools | Options) и ввести новое значение параметра **Период обновления** (RefreshInterval) на вкладке Другие (Advanced)

**Контрольные вопросы**

1. Перечислите режимы для открытия БД?
2. Чем уникален монопольный режим?
3. Какие операции доступны в режиме общего доступа?
4. Как установить режим доступа по умолчанию?
5. С помощью чего можно установить блокировку записей?
6. Охарактеризуйте каждый из способов блокировки записей?
7. Если пользователь не определил значения свойства Блокировка записей в объектах базы данных, то…?
8. Когда появляется команда Параметры в меню Сервис?
9. Что вы знаете об автоматическом обновлении?
10. Перечислите и охарактеризуйте параметры обновлений?

# Список использованных источников

1. Хансен Г., Хансен Д. Базы данных и управление. - М.: Бином, 2014.
2. Дейт К. Введение в системы баз данных. - М.: "Вильямс", 2012.
3. Конноли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. - М.- С./П.- К., 2013.
4. Грофф Д.Р., Вайнберг П.Н. SQL: Полное руководство. - Киев: BMV, "Ирина", 2011.
5. Когаловский М.Р. Энциклопедия технологий баз данных. - М.: Финансы и статистика, 2013.
6. Григорьев Ю.А., Ревунков Г.И. Банки данных. - М.: МГТУ им. Баумана, 2012.
7. Уорсли Дж., Дрейк Дж. PostgreSQL. - Санкт-Петербург, Питер, - 2013.
8. Саймон А. Когаловский М.Р. Стратегические технологии баз данных: менеджмент на 2000 год. - М.: Бином, 2012.
9. Грабер М. Введение в SQL. - М.: ЛОРИ, 2013.
10. Барсегян А.А.,Куприянов М.С.,Степаненко В.В.,Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. - Санкт-Петербург, БХВ-Петербург, 2014.
11. Хоменко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. - Санкт-Петербург, Корона, 2012.
12. Корнеев В.В., Гарев А. Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. - М. - Нолидж, 2013.