

**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА  
И ПРОДОВОЛЬСТВИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**КАФЕДРА ЭКОНОМИЧЕСКОЙ ИНФОРМАТИКИ**

**ТЕХНОЛОГИИ БАЗ ДАННЫХ И ЗНАНИЙ**

*Методические указания и задания к практическим занятиям  
для студентов экономических специальностей  
(Раздел «MS Access»)*

**МИНСК  
2008**

УДК 004.45(07)

ББК 32.97я7

Т 38

Рекомендовано научно-методическим советом факультета предпринимательства и управления БГАТУ

Протокол № 3 от 24 января 2008 г.

Составитель – ст. преподаватель *Т.В. Ероховец*

Рецензенты: канд. техн. наук, зав. каф. вычислительной техники БГАТУ  
*Н.В. Исаеня,*  
канд. техн. наук, доц. каф. моделирования и прогнозирования экономики АПК БГАТУ *Б.М. Астрахан*

УДК 004.45(07)

ББК 32.97я7

© БГАТУ, 2008

## СОДЕРЖАНИЕ

Введение .....	4
Раздел 1 Теоретические сведения. Краткая характеристика Access ...	6
1.1 Окна в Access .....	8
1.2 Поля базы данных .....	11
Раздел 2 Проектирование базы данных учебного примера .....	15
2.1 Проектирование баз данных .....	15
2.2 Проектирование баз данных РАСПИСАНИЕ .....	16
2.3 Проектирование учебной базы данных СКЛАДСКОЙ УЧЕТ .....	27
Раздел 3 Практические работы .....	31
Практическая работа № 1 .....	31
Практическая работа № 2 .....	41
Практическая работа № 3 .....	46
Практическая работа № 4 .....	61
Варианты заданий для контрольной работы .....	67
Литература .....	79

## ВВЕДЕНИЕ

В настоящее время существует множество систем управления базами данных (СУБД) и других программ, выполняющих схожие функции. Среди них особого внимания заслуживает Microsoft Access. К достоинствам Access можно отнести интеграцию с другими средствами Microsoft Office, поддержку технологии ActiveX, возможность работы в Internet и в корпоративных сетях. Этим обусловлено то, что Access подходит для решения задач если не большинства, то, по крайней мере, значительной части пользователей. Области применения Access обозначены достаточно ясно. Во-первых, пользователями этой системы могут быть люди, близкие к вычислительной технике, но не имеющие достаточно времени на ее изучение, поскольку она лежит вне области их профессиональных интересов и служит лишь подспорьем в работе. Пользователей привлекает легкость изучения системы, а также средства быстрого создания приложений. Другая сфера применения этой СУБД – предприятия, имеющие локальную сеть, так как Access зарекомендовал себя при использовании в сети.

Получить информацию о Microsoft Access можно путем изучения справочника, встроенного в систему.

Данные методические указания предназначены для первоначального освоения студентами процесса проектирования базы данных в среде СУБД MS Access. Пособие нацелено на создание студентами базы данных, включающей два внешних представления. Рассматриваются этапы проектирования базы данных: разработка концептуальной схемы базы данных, получение логической схемы, применительно к СУБД MS Access, и, наконец, проектирование объектов базы данных в среде СУБД Access.

Методика построена таким образом, чтобы на примере создания реальной базы данных студенты освоили наибольшее количество приемов и методов работы с базами данных и наиболее полно изучили возможности СУБД MS Access. Вместе с тем, целью методики является пробуждение у студентов интереса к технологиям обработки информации и проектирования баз данных.

Методические указания содержат задания трех уровней. На первом уровне для получения навыков работы в среде СУБД MS Access студентам предлагаются задания по проектированию учебной базы данных РАСПИСАНИЕ с подробными рекомендациями

по их выполнению. На втором уровне студенты выполняют задания самостоятельно, учитывая некоторые краткие указания и пояснения. На этом этапе студенты конструируют базу данных СКЛАДСКОЙ УЧЕТ, которая максимально приближена к реальной задаче бухгалтерского учета. Цель второго уровня – закрепить навыки, полученные при выполнении заданий первого уровня, а студенты, имеющие опыт работы в СУБД MS Access могут сразу приступить к заданиям второго уровня. На третьем уровне студенты выполняют контрольную работу по индивидуальному заданию в соответствии с предложенным вариантом. Варианты индивидуальных заданий предполагают создание студентами реальных баз данных для решения учетных задач на предприятии, таких как «Учет основных средств», «Учет банковских документов», «Расчеты с поставщиками и заказчиками» и др.

Навыки организации и обработки информации, а так же базы данных, созданные в процессе выполнения индивидуальных заданий, могут быть использованы студентами в дальнейшей учебной, научной и трудовой деятельности, а также могут послужить основой для выполнения дипломной работы студентами, проявившими интерес к технологиям организации и обработки информации.

# РАЗДЕЛ 1

## ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### КРАТКАЯ ХАРАКТЕРИСТИКА ACCESS

Система управления базами данных (СУБД) Microsoft Access – это одна из самых популярных СУБД общего назначения. Она позволяет создавать, редактировать и обрабатывать реляционные базы данных (БД), в которых текстовая, числовая и иная информация хранится в связанных таблицах.

В Microsoft Access будем называть объектами все то, что может иметь имя. Основными объектами в СУБД Access являются таблицы, запросы, формы, отчет, макросы, модули.

**Таблица** – это структура, предназначенная для хранения информации.

Таблица, как правило, включает информацию об объекте определенного типа, например, сведения о работниках предприятия. Строки таблицы называются *записями* и содержат характеристики одного экземпляра объекта, например, табельный номер, фамилию, должность, дату рождения, оклад и т.д.

Каждая характеристика записывается *в поле*, таким образом, запись состоит из полей. Каждое поле имеет уникальное имя и некоторое значение.

Например:

имя поля:	ФИО
значение:	Иванов

Если рассматривать значение одного и того же поля для различных записей, то получается столбец таблицы (например, список фамилий).

Поле характеризуется типом данных, единым для всего столбца. Тип данных определяет длину поля.

Записи могут идентифицироваться по некоторой уникальной характеристике, которая включает одно или несколько полей и называется *ключом*.

Ключи бывают трех типов:

**Ключевое поле счетчика** – это уникальный последовательный или случайный номер записи, появляющийся автоматически при добавлении каждой новой записи.

**Простой ключ** – уникальное поле в таблице, объявленное ключевым.

**Составной ключ** – состоящий из нескольких полей. Значение набора полей должно быть уникальным.

**Запрос** – это требование на отбор данных, хранящихся в таблицах; на выполнение определенных действий с данными. Запрос позволяет создать набор из записей, поля которых находятся в разных таблицах, и использовать его как источник данных для формы или отчета. Некоторые типы запросов позволяют вносить изменения в таблицы.

**Форма** представляет собой созданный на экране шаблон для ввода, просмотра и редактирования данных таблиц и запросов, а также для управления работой приложения.

**Отчет** предназначен для создания документа, который может быть распечатан или включен в документ другого приложения.

**Страница** – страница доступа к данным, содержащая информацию, извлеченную из базы данных и сохраненную в формате HTML.

**Макрос** – есть последовательность макрокоманд для автоматизации выполнения операций в среде Access без программирования.

**Модуль** – это программа для обработки данных, написанная на языке Visual Basic for Applications (VBA). Access позволяет создавать эффективные модули для работы с БД, содержащие меню, диалоговые окна и командные кнопки.

Все объекты БД: таблицы, запросы, формы, отчеты, макросы и модули хранятся в едином файле – в файле БД, имеющем расширение .mdb.

Для автоматизации проектирования объектов БД MS Access имеет специализированные программы-конструкторы и мастера.

**Конструктор** предоставляет пользователю ряд инструментальных средств, с помощью которых можно быстро и просто создавать и модифицировать объекты БД.

**Мастер** позволяет построить вполне законченный объект БД на основе имеющихся в коллекции Access стандартных объектов в соответствии с требованиями пользователя.

MS Access располагает разнообразными графическими средствами для оформления таблиц, форм, отчетов.

Кроме проектирования объектов БД MS Access осуществляет управление БД:

- защиту;
- репликацию (создание специальных копий БД, с которыми пользователи могут одновременно работать на разных компьютерах);
- восстановление;
- сжатие;
- просмотр сведений о БД;
- поиск файла БД по свойствам БД; а также экспорт и импорт данных.

СУБД MS Access – 32-разрядная СУБД, является частью пакета Microsoft Office, полностью совместима с текстовым редактором Word и табличным процессором Excel пакета Microsoft Office.

## 1.1 Окна в Access

Запуск Access может быть осуществлен по команде **Пуск / Программы / Access**. На экране появится стартовое окно Access (рисунок 1.1).

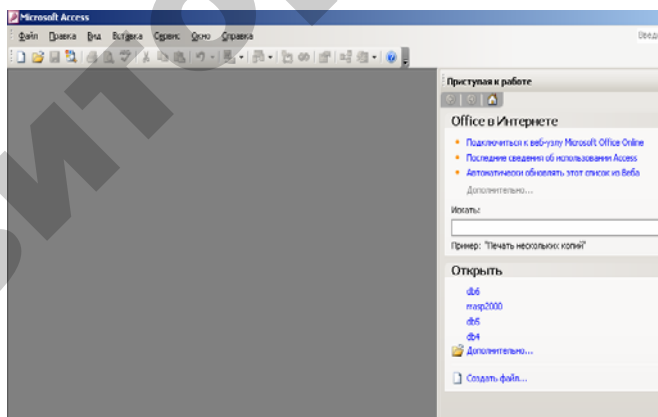


Рисунок 1.1 – Стартовое окно Access



В этом окне, выбирая соответствующие функции, можно выполнить такие действия, как создать новую базу данных или открыть существующую. Эти действия можно выполнить также по командам: **Файл / Создать**, **Файл / Открыть**. После этого появится главное окно Access (рисунок 1.2).

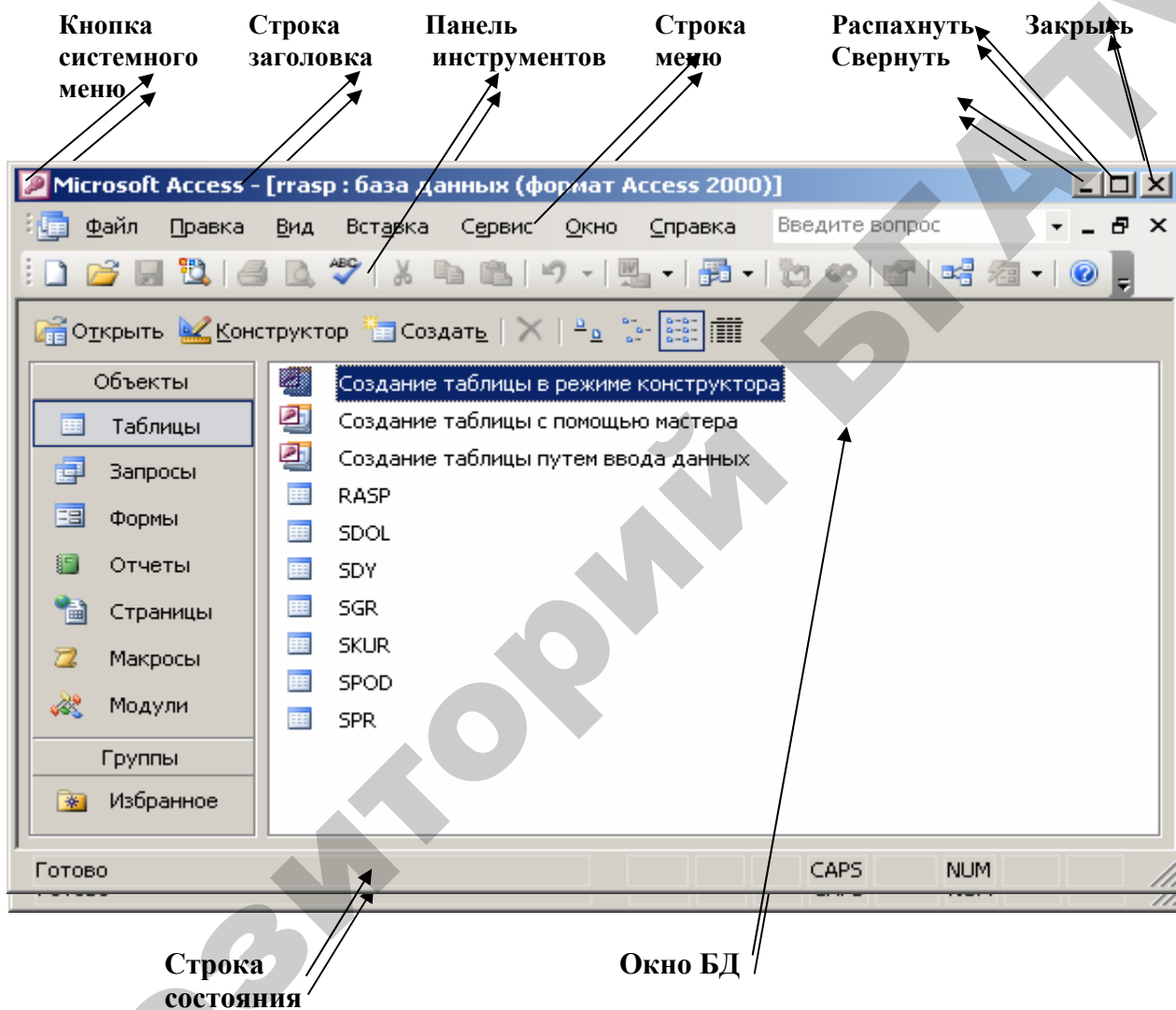


Рисунок 1.2 – Главное окно Access

Это окно можно распахнуть, свернуть, закрыть (с помощью кнопок в правом верхнем углу, подобно другим окнам Windows). *Кнопка системного меню* позволяет управлять главным окном Access. *Строка меню* содержит пункты меню, каждому из которых соответствует подменю.

Команды подменю бывают трех типов:

- выполняющие конкретные действия;
- вызывающие меню следующего уровня (рядом с такими командами изображен треугольник);

– вызывающие диалоговые окна (после таких команд стоит многоточие).

Перечень пунктов меню и их содержание изменяется в зависимости от режимов работы системы.

*Панель инструментов* представляет собой набор графических кнопок, назначение которых – это ускоренный вызов команд меню. Ее можно перемещать по экрану, изменять в размерах.

С помощью команды: **Вид / Панели инструментов** можно отобразить, спрятать, настроить любую панель, создать новую, включить отображение крупных кнопок, всплывающих подсказок о назначении кнопок, цветных пиктограмм на кнопках.

*В строке состояния* в левой части отображается информация о текущем режиме работы системы, справа – индикаторы клавиатуры. Эту строку можно отображать или нет в главном окне по команде **Сервис / Параметры / Вид**.

*В рабочей области главного окна Access* располагается окно БД. В нем сосредоточены все «рычаги управления» БД.

*В строке заголовка* окна БД отображено имя открытой БД.

В левой части окна расположены **вкладки**: Таблицы, Запросы, Формы, Отчеты, Макросы, Модули. Здесь выбирается нужный тип объекта.

Кнопки, расположенные в верхней части окна БД, служат для работы с текущим объектом БД: создания, открытия, изменения (кнопка Конструктор).

*В рабочей области* окна БД отображается **список объектов** выбранного типа. С помощью команды **Вид** можно изменять представление объектов в списке: отобразить в виде списка, крупным планом, мелкими значками, в виде таблицы с характеристиками объектов (с именами, датами создания, датами последней модификации, описаниями), упорядочивать объекты по одной из них.

Просматривать, изменять и задавать структуру объектов следует в окнах **конструкторов** таблиц, запросов, форм, отчетов, макросов и модулей.

Просматривать, изменять и задавать значения полей объектов следует в **окнах объектов** БД.

Задание, просмотр и изменение характеристик элементов объектов БД происходит в **окнах свойств**, а вывод сообщений Access осуществляется в **окнах сообщений**.

**Диалоговые окна** имеют разнообразную структуру, но в них используются одни и те же типовые элементы:


- вкладки, располагающиеся в верхней части окна и используемые для перехода на некоторый уровень меню;
- поля ввода  , в которых вводится запрашиваемая информация;
- списки, служащие для выбора нужного значения;
- раскрывающиеся списки  ;
- флажки  , позволяющие вводить или отменять совместимую команду (не исключаящую других команд группы). В диалоговом окне несколько однотипных команд объединяются в группы;
- переключатели  , позволяющие отменять или вводить альтернативную команду, назначение которой отменяет остальные команды группы.

В Access одновременно могут быть открытыми несколько окон. Разместить их удобно на экране позволяют команды подменю пункта **Окно**.

Завершение работы с Access может быть осуществлено несколькими способами:

- щелкнув по кнопке закрытия окна Access (как правило на ней изображен крестик);
- по команде **Файл / Выход**;
- нажав комбинацию **Alt + F4**;
- по команде **Закреть** из комплексного меню окна экрана Access.

СУБД Access имеет достаточно разнообразную справочную систему. Существует множество видов справок:

- можно получить контекстно-зависимую справку, нажав **F1** в любой момент работы в системе;
- выбрав путь «?» в строке меню, обратиться к стандартной справке Windows;
- щелчком мыши по кнопке  на панели инструментов, а затем на некотором элементе окна вызывается всплывающая подсказка.

## 1.2 Поля базы данных

Как отмечалось ранее, записи таблиц состоят из **полей**, значениями которых являются характеристики экземпляров объекта.

При проектировании БД поля необходимо описать в соответствии с требованиями Access.

Каждому полю следует:

- присвоить имя;
- указать тип;
- указать некоторые свойства поля.

Имя поля может состоять из не более чем 64 любых символов кроме точки (.), восклицательного знака (!), апострофа (‘) и квадратных скобок ([ ]).

Имя поля не должно начинаться с пробела и не может включать управляющие символы (с кодами ASCII от 0 до 31).

Существуют следующие типы полей в Access:

1. **Текстовый.** Позволяет вводить любые символы, в том числе и цифры, для которых не предполагается выполнение расчетов.
2. **Поле МЕМО.** Длинный текст (до 64000) символов, например, примечания или описания.
3. **Числовой.** Вводятся числовые данные, допускающие использование в математических вычислениях.
4. **Дата / время.** Значения даты или времени.
5. **Денежный.** Денежные значения. Этот тип позволяет проводить вычисления без округления значения. Максимальная точность составляет 15 знаков слева от десятичной запятой и 4 знака справа от запятой.
6. **Счетчик.** Уникальные последовательности (с шагом 1) или случайные номера, автоматически формируемые при добавлении записи в БД.
7. **Логический.** Поля, которые могут иметь одно значение из двух возможных, таких как Да / Нет, Истина / Ложь, Вкл. / Выкл.
8. **Поле объекта OLE.** Объекты, созданные в других приложениях, которые могут быть связаны или внедрены в таблицу Microsoft Access. Например, документы Microsoft Word, электронные таблицы Microsoft Excel, рисунки и др.
9. **Гиперссылка.** Содержанием поля является адрес в сетях Internet, служащий ссылкой на ресурсы Word Wide Web.
10. **Мастер подстановок.** Создает поля, позволяющие выбрать с помощью раскрывающегося списка значения из других таблиц или из списка значений.

Набор допустимых свойств поля зависит от типа поля. В таблице 1.1 приведены наборы свойств полей с часто используемыми типами данных.

Таблица 1.1

Числовой тип данных	Текстовый тип данных	Тип данных Дата / Время
Размер поля	Размер поля	-
Формат поля	Формат поля	Формат поля
Число десятичных знаков	-	
Маска ввода	Маска ввода	Маска ввода
Подпись	Подпись	Подпись
Значение по умолчанию	Значение по умолчанию	Значение по умолчанию
Условие на значение	Условие на значение	Условие на значение
Сообщение об ошибке	Сообщение об ошибке	Сообщение об ошибке
Обязательное поле	Обязательное поле	Обязательное поле
	Пустые строки	-
Индексированное поле	Индексированное поле	Индексированное поле
	Сжатие Юникод	Сжатие Юникод
	Режим IME	Режим IME
	Режим	Режим предложений
Смарт-теги	предложений IME Смарт-теги	IME Смарт-теги

Рассмотрим некоторые из указанных свойств полей.

**Размер поля.** Для числового поля допустимыми являются следующие значения:

- ◆ целые числа от 0 до 255. Данный размер поля обозначается в Access как байт;
- ◆ целые числа от -32 768 до 32 767 (обозначение размера – целое);
- ◆ целые числа от -2 147 483 648 до 2 147 483 647 (длинное целое);
- ◆ числа с плавающей точкой от - 3.402823E38 до 3.402823E38, в дробной части до 7 знаков (с плавающей точкой (4 байт));
- ◆ числа с плавающей точкой от - 1.79769313486232E308 до 1.79769313486232E308, в дробной части – до 15 знаков (с плавающей точкой (8 байт)).

Размер текстового поля – до 255 символов.

**Формат поля.** Это свойство определяет способ отображения текста, чисел, дат и значений времени на экране и на печати.

**Число десятичных знаков.** Дает возможность указывать для чисел количество дробных знаков.

**Маска ввода.** Задает маску ввода, облегчающую ввод данных в поле.

**Подпись.** Определяет текст, который выводится в подписях полей в формах и отчетах.

**Значение по умолчанию.** Позволяет указать значение, автоматически вводящееся в поле при создании новой записи.

**Условие на значение.** Определяет требование к данным, вводимым в поле.

**Сообщение об ошибке.** Позволяет указать текст сообщения, выводимого на экран, если введенные данные нарушают условие, определенное в свойстве «Условие на значение».

**Обязательное поле.** Указывает, требует ли поле обязательного ввода значения.

**Пустые строки.** Определяет, допускается ли ввод в текстовое поле пустых строк (не содержащих символов).

**Индексированное поле.** Задаёт индекс для поля, ускоряющий поиск и сортировку в таблице.

## РАЗДЕЛ 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ УЧЕБНОГО ПРИМЕРА

### 2.1 Проектирование баз данных

Процесс проектирования базы данных обычно осуществляется в 3–4 этапа.

Суть процесса проектирования состоит в том, чтобы собрать необходимые сведения из Предметной Области, изучить требования пользователей к результатам, и отобразить эти знания о Предметной Области в подходящей СУБД в виде системы баз данных.

**Этап 1: (Предварительный).** Анализ Предметной Области и требований к будущей системе баз данных.

**Результат:** наборы информационных потребностей различных групп пользователей.

**Этап 2: Концептуальное проектирование (инфологическое моделирование).** Сбор информации о Предметной Области и представление ее в формализованном виде.

На втором этапе создаются описания внешних представлений пользователей, а также концептуальное представление всей БД (в виде внешних схем и концептуальной схемы). Описание производится без ориентации на используемые в дальнейшем программные и аппаратные средства.

**Результат:** Концептуальная схема и внешние схемы пользователей (или инфологическая модель)

**Этап 3: Логическое проектирование (дatalogическое моделирование).** На третьем этапе выполняют отображение концептуальной схемы в логическую схему с использованием конкретной СУБД. При отображении в реляционную СУБД концептуальная схема преобразуется в набор связанных двумерных таблиц. Эта модель строится в терминах информационных единиц, допустимых в конкретной выбранной СУБД.

Во внешних схемах часто задается не только логическая структура части БД с точки зрения конкретного пользователя (или приложения), но и допустимые режимы обработки в рамках этой внешней схемы. Использование аппарата внешних схем облегчает работу пользователя, т.к. он должен знать структуру только части БД, которая к тому же приспособлена к его потребностям.

**Результат:** Логическая схема данных, в терминах выбранной СУБД.

**Этап 4: Физическое проектирование (физическая модель).** Привязка Логической схемы (ДЛМ) к конкретной среде хранения. Физическая модель определяет используемые аппаратные ресурсы, способы физической организации данных в среде хранения. Физическая модель строится с учетом возможностей СУБД.

**Результат:** описание физической структуры БД в виде **схемы хранения**.

## 2.2 Проектирование базы данных РАСПИСАНИЕ

Пусть в результате первого и второго этапов получены следующие внешние представления.

**Первое** внешнее представление, содержит информацию для составления расписания проведения занятий преподавателями на факультете. Расписание должно содержать варианты, удобные для его восприятия преподавателями (вывод всех данных по каждому преподавателю), и варианты, удобные для восприятия студентами группы (вывод всех данных по группе).

Расписание может выглядеть подобно таблице 2.1.

Таблица 2.1

Преподаватель	Должность	Группа	Дисциплина	День недели	Кол-во часов
Иванов А.П.	Профессор	1 эи	СБДЗ	Понедельник	2
Иванов А.П.	Профессор	1 эи	СБДЗ	Среда	4
Иванов А.П.	Профессор	2 эи	Информ.технол.	Понедельник	2
Иванов А.П.	Профессор	2 эи	Информ.технол.	Пятница	4
Иванов А.П.	Профессор	3 эи	ОС	Среда	4
Иванов А.П.	Профессор	3 эи	ОС	Четверг	6
Иванов А.П.	Профессор	3 эи	ОС	Пятница	2
Петров М.А.	Доцент	1 эи	ЭММ	Понедельник	2
Петров М.А.	Доцент	1 эи	ЭММ	Среда	4
Петров М.А.	Доцент	3 эи	Выс.математика	Среда	4
Петров М.А.	Доцент	3 эи	Выс.математика	Четверг	2
Петров М.А.	Доцент	3 эи	Выс.математика	Пятница	2

**Семантическое моделирование** (подход «Объект – Свойство – Связь») – наиболее популярный подход к формальному описанию предметной области.



Этот подход основан на признании факта существования в реальном мире **объектов**. Объекты имеют наборы характеристик (или **свойств**) и взаимодействуют между собой с помощью **связей**.

Методологии проектирования, основанные на идеях семантического моделирования, часто называют **нисходящими** методологиями, т.к. они начинают с высшего уровня абстракции – конструкции реального мира, и заканчивают на уровне создания конкретной схемы базы данных.

Рассмотрим определения основных понятий семантического моделирования:

**Сущность (entity)** – собирательное понятие, некоторая **абстракция** реально существующего **объекта, процесса** или **явления**, о котором необходимо хранить информацию в базе данных. В семантическом моделировании применяют не просто понятие «сущность», а говорят «тип сущностей»:

**Тип сущностей** – определяет набор объектов с одним и тем же набором свойств. **Экземпляр сущности** – конкретный объект в наборе.

Например, если мы хотим описать преподавателей университета, то все те общие свойства, которые присущи всем преподавателям (это может быть «Табельный номер», «Фамилия», «должность», «Педагогический стаж»), сформируют **тип сущности ПРЕПОДАВАТЕЛЬ**. Тогда каждый отдельный преподаватель является **экземпляром сущности ПРЕПОДАВАТЕЛЬ**.

Каждый тип сущности имеет набор свойств (характеристик), присущих всем экземплярам данного типа.

**Свойство (attribute, атрибут)** – поименованная характеристика сущности, которая принимает значения из некоторого множества значений (домена).

Для идентификации **экземпляра типа сущности** используются специальные свойства. Это может быть одно или несколько свойств, значения которых позволяют **однозначно** отличать один экземпляр сущности от другого. Этот набор специальных свойств называется **первичным ключом**. Например, в типе сущностей **ПРЕПОДАВАТЕЛЬ** первичным ключом следует объявить поле **ТАБЕЛЬНЫЙ НОМЕР**, так как это поле будет иметь уникальное значения для каждого преподавателя.

Различают типы сущностей: **простые**, т.е. неделимые сущности, и **сложные** сущности. Сложные сущности бывают:

- **составные** – соответствуют отображению «целое – часть» (пример, компьютер – устройства);
- **обобщенные** – соответствует отображению «род-вид» или «супертип-подтип» (пример, принтер: лазерный, струйный, матричный);
- **агрегированные** – соответствует обычно какому-либо **процессу**, в который вовлечены другие объекты.

В нашем внешнем представлении участвуют простые типы сущностей: ПРЕПОДАВАТЕЛЬ, ГРУППА, ДИСЦИПЛИНА, ДЕНЬ НЕДЕЛИ, ДОЛЖНОСТЬ и агрегированный тип сущностей – РАСПИСАНИЕ.

Наиболее известна модель графического представления концептуальной схемы базы данных, созданная *Питером Ченом* (первая статья, посвященная этому методу, появилась в 1967 г.). Она называется «модель «Сущность-Связь» (Entity-Relationship model, ER-model). Особенностью этой модели является то, что части предметной области, соответствующие объектам, свойствам и связям изображаются в виде диаграмм. **Взаимодействия** между двумя сущностями, например, ПРЕПОДАВАТЕЛЬ и РАСПИСАНИЕ ЗАНЯТИЙ представлены фразами «*проводит несколько*» и «*проводится одним*». ПРЕПОДАВАТЕЛЬ проводит несколько занятий по РАСПИСАНИЮ, но каждое конкретное занятие проводится лишь одним преподавателем.

Говорят, что в предметной области объекты **взаимодействуют** друг с другом посредством **связей (relationships)**.

В связи может участвовать два и более объектов. Связи, в которых участвуют два объекта, называются **бинарными**. Связи, в которых участвуют три объекта - **тернарные**, и т.д. Различают такие типы связей: «один – к – одному» (1:1), «один – ко – многим» (1:M), «многие – ко – многим» (N:M).

Связь «**один – к – одному**» означает, что каждому экземпляру одного типа сущностей соответствует один экземпляр другого типа сущностей, и наоборот (рисунок 2.1, *а*).

Связь «**один – ко – многим**» означает, что каждому экземпляру одного типа сущностей (А) соответствует один или более экземпляров другого типа сущностей (В), однако каждому экземпляру типа В соответствует только один экземпляр типа А (рисунок 2.1, *б*).

Пример. ГРУППА ↔ РАСПИСАНИЕ.

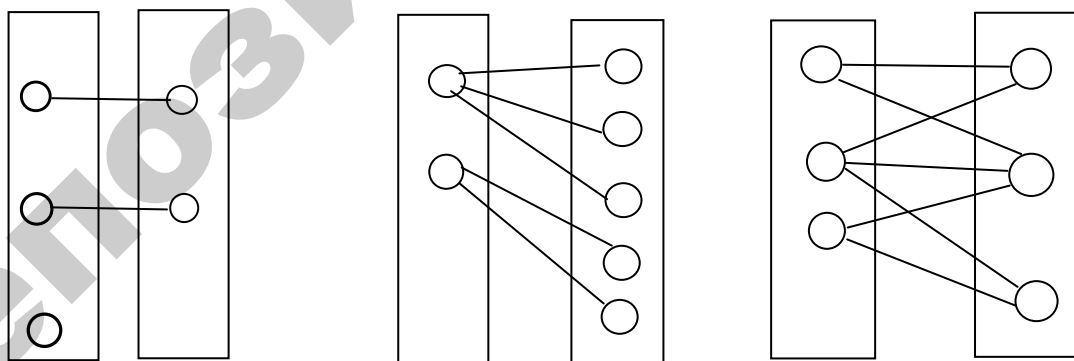
Связь «многие – ко – многим» означает, что каждому экземпляру одного типа сущностей (А) соответствует один или более экземпляров другого типа сущностей (В), и наоборот (рисунок 2.1, в). В реляционных БД этот тип связи сводится к связи «один – ко – многим».

В нашем примере каждый преподаватель проводит много занятий по расписанию, но каждое занятие проводится одним преподавателем, поэтому связь между сущностями ПРЕПОДАВАТЕЛЬ и РАСПИСАНИЕ типа 1:М. Аналогичный

ти имеют и все остальные связи.

п имеют и

Итак, получена следующая ER-диаграмма нашего внешнего представления (рисунок 2.2).



а) Один – к – одному

б) Один – ко – многим

в) Многие – ко – многим

Рисунок 2.1 – Виды связей между таблицами

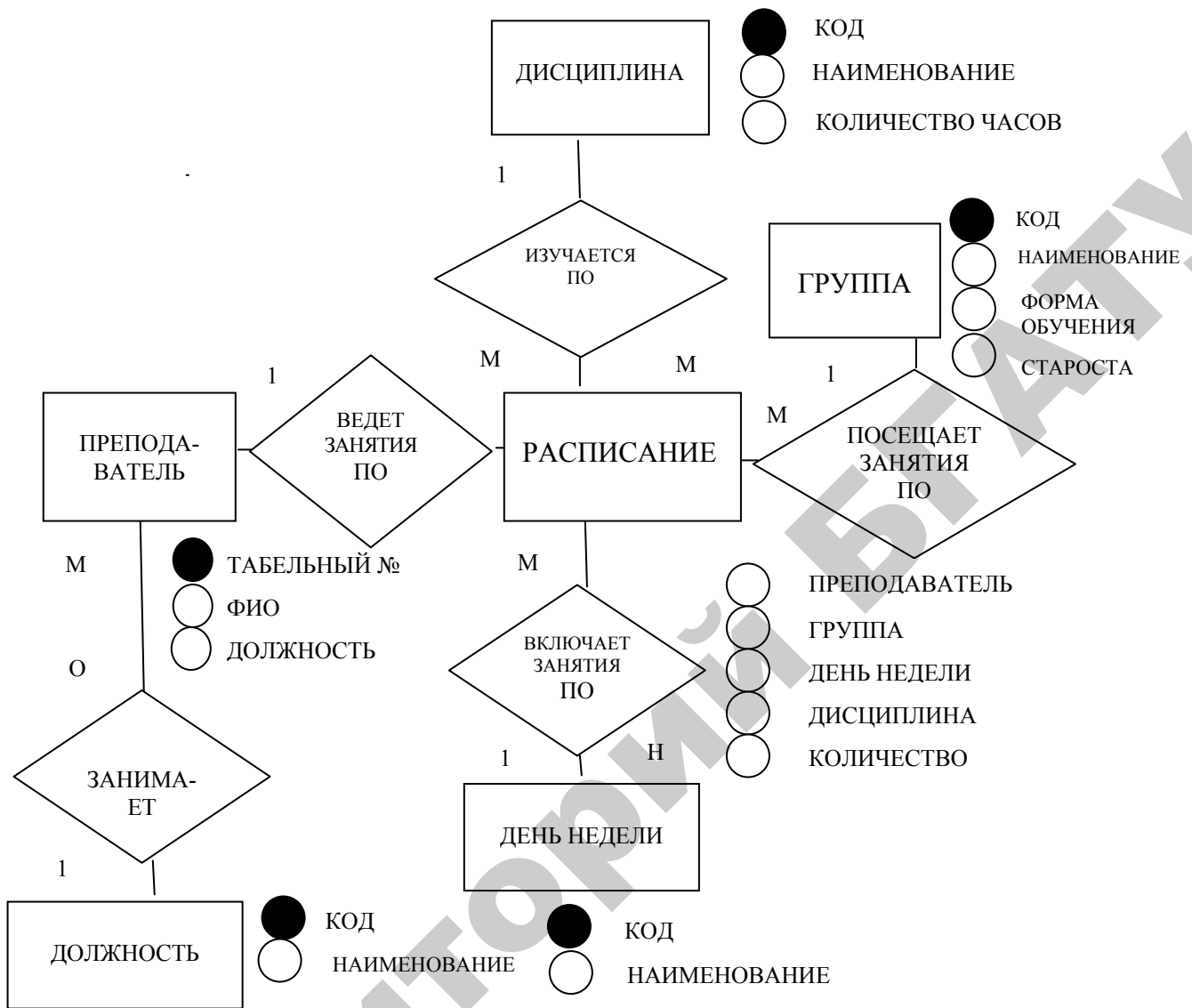


Рисунок 2.2 – ER-диаграмма для первого внешнего представления

**Замечание 2.1** Свойства, обозначенные затемненными кружками означают ключевые поля.

При определении типов сущностей ПРЕПОДАВАТЕЛЬ, ГРУППА, ДИСЦИПЛИНА, ДЕНЬ НЕДЕЛИ, ДОЛЖНОСТЬ мы неявно добавили по одному свойству к каждому типу: «Табельный номер преподавателя» и коды объектов, и сделали их первичными ключами. Как известно, в концептуальном моделировании с помощью диаграмм Чена, каждый простой тип сущностей должен обладать первичным ключом. Поэтому часто, чтобы не делать первичным ключом длинную текстовую строку (значение которой к тому же может не быть уникальным), вводят дополнительный атрибут, основная задача которого – быть уникальным идентификатором

экземпляра типа сущностей. Например, может возникнуть ситуация, когда у преподавателя совпадет и фамилия, и имя, и отчество, в этой ситуации различие будет устанавливаться по табельному номеру.

**Второе** внешнее представление описывает информацию для отдела кадров и соответствует следующей ER-диаграмме.

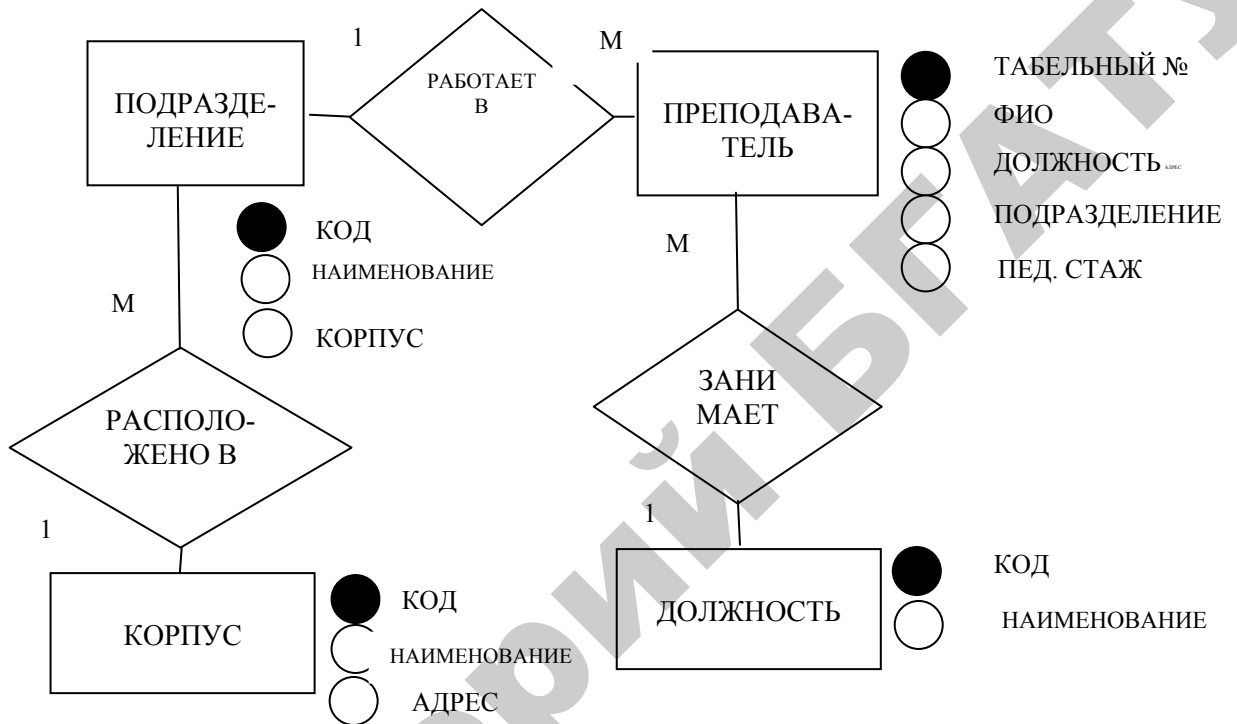


Рисунок 2.3 – ER-диаграмма для второго внешнего представления

Объединяя оба внешних представления, получаем концептуальную схему проектируемой базы данных, назовем ее РАСПИСАНИЕ (рисунок 2.4).

В процессе логического проектирования, с ориентацией на СУБД MS Access, и с учетом законов нормализации базы данных, получена следующая совокупность логически связанных таблиц (таблицы 2.2–2.9).

**Замечание 2.2** Очевидно, что сущности отобразились в таблицы, свойства – в поля записей.

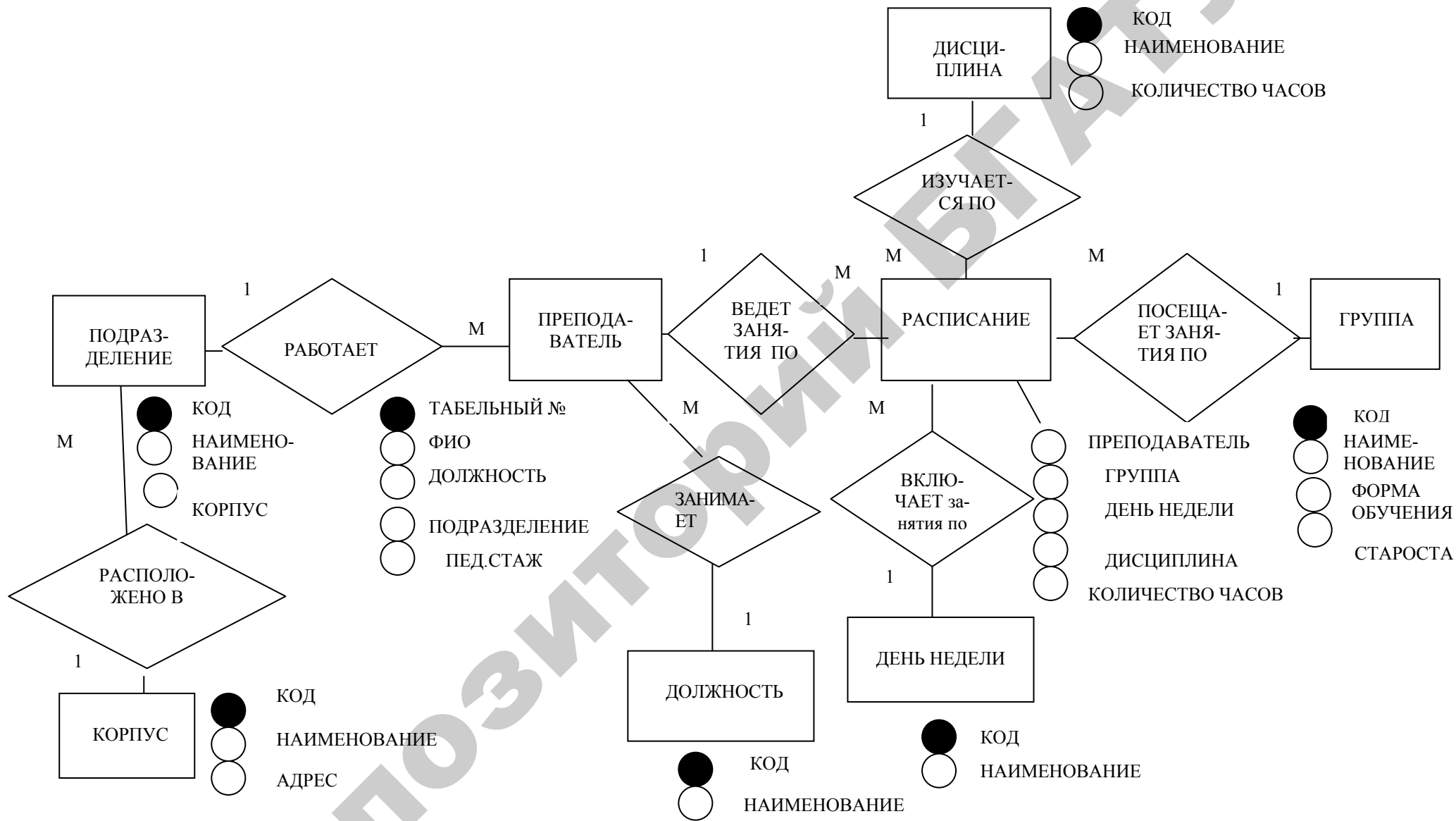


Рисунок 2.4 – ER-диаграмма базы данных РАСПИСАНИЕ

1. Таблица-расписание – назовем ее **RASP**

Таблица 2.2

TNPR	GR	KKUR	DAY	KOLCH
1	1	1	1	2
1	1	1	3	2
1	2	3	1	4
1	2	3	5	4
1	3	2	3	6
1	3	2	4	2
1	3	2	5	2
2	1	4	1	2
2	1	4	3	4
2	3	5	3	8
2	3	5	4	6
2	3	5	5	2

Со следующей структурой записи:

Имя поля	Тип данных	Описание
TNPR	– числовой	Табельный номер преподавателя
GR	– числовой	Код группы
KKUR	– числовой	Код дисциплины (курса)
DAY	– числовой	Код дня
KOLCH	– числовой	Количество часов

2. Справочник преподавателей, имеющий имя **SPR**:

Таблица 2.3

TNPR	FIO	KDOL	PSTAG	KPOD
1	Иванов	1	25	1
2	Петров	4	5	3
3	Федоров	3	15	1
4	Нестерович	2	16	4

Со следующей структурой записи:

Имя поля	Тип данных	Описание
TNPR	– числовой	Табельный номер преподавателя
FIO	– текстовый	Фамилия преподавателя
KDOL	– числовой	Должность
PSTAG	– числовой	Педагогический стаж
KPOD	– числовой	Код подразделения (места работы)

3. Справочник групп (**SGR**)

Таблица 2.4

KGR	NAIMGR	FO	KOF	STAR
1	1 эи	д	4	Васильев В.М.
2	2 эи	д	2	Лойко С.И.
3	3 эи	д	4	Денисов Р.В.
4	5 тк	з	2	Старин А.И.
5	64 м	в	2	Кулик А.В.

Со следующей структурой записи:

Имя поля	Тип данных	Описание
KGR	– числовой	Код группы
NAIMGR	– текстовый	Наименование группы
FO	– текстовый	Форма обучения
KOF	– числовой	Коэффициент
STAR	– текстовый	Фамилия старосты

#### 4. Справочник дней (SDY)

Таблица 2.5

KDN	NDN
1	Понедельник
2	Вторник
3	Среда
4	Четверг
5	Пятница
6	Суббота

Со следующей структурой записи:

Имя поля	Тип данных	Описание
KDN	– числовой	Код дня
NDN	– текстовый	Наименование дня

#### 5. Справочник курсов (дисциплин), имеющий имя SKUR

Таблица 2.6

KKUR	NAIMKUR	CHAS
1	ТБДЗ	76
2	ОС	48
3	Информ. технологии	60
4	ЭММ	64
5	Высшая математика	70

Со следующей структурой записи:

Имя поля	Тип данных	Описание
KKUR	– числовой	Код дисциплины (курса)
NAIMKUR	– текстовый	Наименование дисциплины
CHAS	– числовой	Общее количество часов

#### 6. Справочник должностей, имеющий имя SDOL

Таблица 2.7

KDOL	NAIMDOL
1	Профессор
2	Доцент
3	Старший преподаватель
4	Ассистент
5	Старший лаборант



Со следующей структурой записи:

Имя поля	Тип данных	Описание
KDOL	– числовой	Код должности
NAIMDOL	– текстовый	Наименование должности

## 7. Справочник подразделений (SPOD)

Таблица 2.8

KPOD	NPOD	KORP
1	Кафедра ЭИ	1
2	Отдел АСУ	3
3	Кафедра ВМ	4
4	НИС	1
5	Кафедра АСУ	2

Со следующей структурой записи:

Имя поля	Тип данных	Описание
KPOD	– числовой	Код подразделения
NPOD	– текстовый	Наименование подразделения
KORP	– числовой	Код корпуса

## 8. Справочник корпусов, имеющий имя SKOR

Таблица 2.9

KKOR	NAIMKOR	ADRES
1	Главный корпус	Г. Минск, пр. Независимости, 99
2	Лабораторный корпус №2	Г. Минск, пр. Независимости, 93
3	Лабораторный корпус №3	Г. Минск, пр. Независимости, 97
4	Лабораторный корпус №4	Г. Минск, пр. Независимости, 95
5	Лабораторно-испытательный корпус	П. Боровляны, ул. Гая, 18

Со следующей структурой записи:

Имя поля	Тип данных	Описание
KKOR	– числовой	Код корпуса
NAIMKOR	– текстовый	Наименование корпуса
ADRES	– текстовый	Адрес

**Замечание 2.3** Так как в таблице 2.1 экземпляры объектов (преподаватель, должность, группа, дисциплина, день недели) многократно повторяются, в базе данных таблицы 2.2 их удобнее представлять кодами (короткими обозначениями). А наименования экземпляров объектов и другая информация об объектах может быть получена из справочников объектов, которые могут содержать разные сведения об объектах.

Проверим, нормализованы ли таблицы (отношения) в разрабатываемой базе данных.

**Отношение соответствует первой нормальной форме (1NF)**, если оно удовлетворяет следующим требованиям:

- отношение не должно иметь повторяющихся записей;
- в отношении не должно быть повторяющихся групп, т.е. атрибутов, которые имеют несколько значений в каждой строке (т. е. не должно быть множественных свойств отношений).

**Отношение соответствует второй нормальной форме (2NF)**, если оно удовлетворяет следующим требованиям:

- отношение должно находиться в 1-й нормальной форме;
- любое не ключевое поле однозначно идентифицируется полным набором ключевых полей.

**Отношение находится в третьей нормальной форме**, если оно удовлетворяет следующим требованиям:

- оно должно находиться во 2-й нормальной форме;
- ни одно из не ключевых полей не идентифицируется с помощью другого не ключевого поля.

Очевидно, что все разработанные таблицы отвечают 1-й, 2-й и 3-й нормальным формам.

Таблица RASP является главной по отношению к таблицам-справочникам. В ней все объекты представлены своими числовыми кодами, по которым легко получить соответствующие наименования из справочников. Таким образом, таблица представлена компактно и занимает мало оперативной памяти.

В таблицах-справочниках каждому объекту соответствует только одна запись, содержащая сведения об объекте. Например, в справочнике групп для каждой группы содержится одна запись, указывающая: код группы, ее наименование, форму обучения, коэффициент, фамилию старосты. Этой же группе соответствует несколько записей в таблице RASP. Поэтому между указанными таблицами устанавливается связь «один – ко – многим» по полям **KGR** (из таблицы **SGR**) – **GR** (из таблицы **RASP**).

**ВНИМАНИЕ!** Эти поля должны иметь одинаковый тип данных.

Для обеспечения этой связи поле **KGR** (код группы) в таблице **SGR** объявляется **ключевым**. Его значение уникально для каждой записи. Поле же **GR** в таблице **RASP** называется **полем внешнего ключа**. Аналогично устанавливаются связи справочников **SPR**, **SKUR** и **SDY** с таблицей **RASP**. Таблица **SPOD** (справочник подразделений) содержит коды и названия отделов и кафедр, где

работают преподаватели, причем для каждого подразделения здесь содержится одна запись, а в таблице **SPR** – код подразделения повторяется, поэтому между таблицами **SPR** и **SPOD** также следует установить связь «один – ко – многим» по полям **KPOD** (из **SPR**) – **KPOD** (из **SPOD**).

**ВНИМАНИЕ! Во всех справочниках коды объектов следует объявить ключевыми полями.** Ниже приведена информационно-логическая модель проектируемой базы данных (рисунок 2.5).

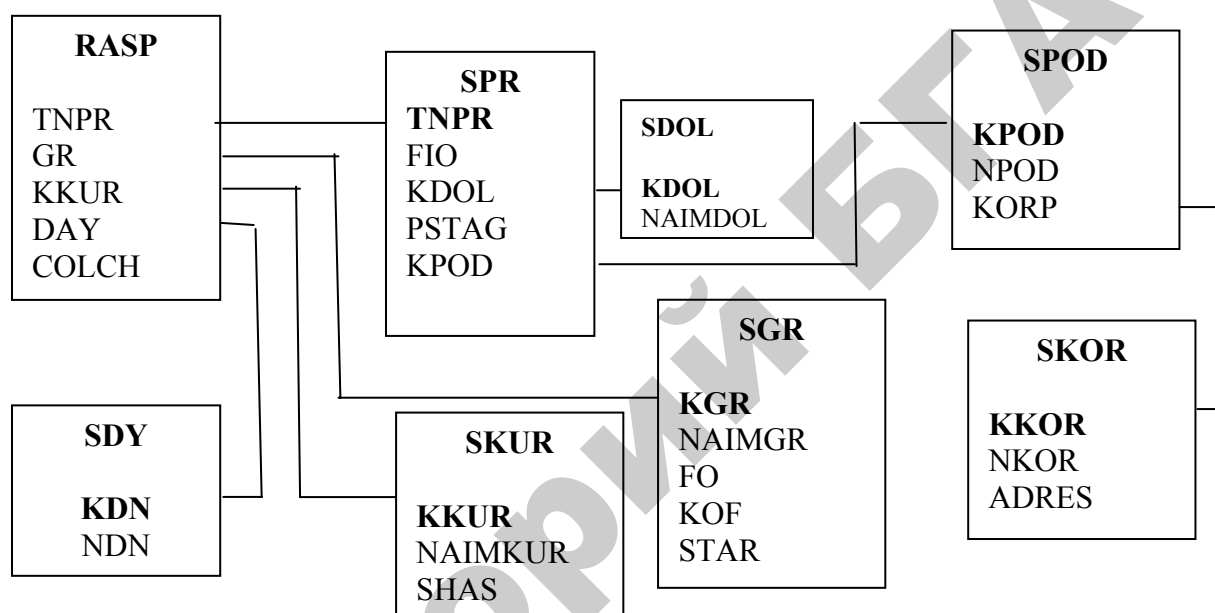


Рисунок 2.5 – Информационно-логическая модель БД РАСПИСАНИЕ

Далее следует приступить к этапу физического проектирования базы данных.

## 2.3 Проектирование учебной базы данных СКЛАДСКОЙ УЧЕТ

### Самостоятельная работа

Автоматизация складского учета канцелярских товаров.

Пусть в результате анализа информации по ведению складского учета канцелярских товаров и концептуального проектирования получена следующая концептуальная модель базы данных.

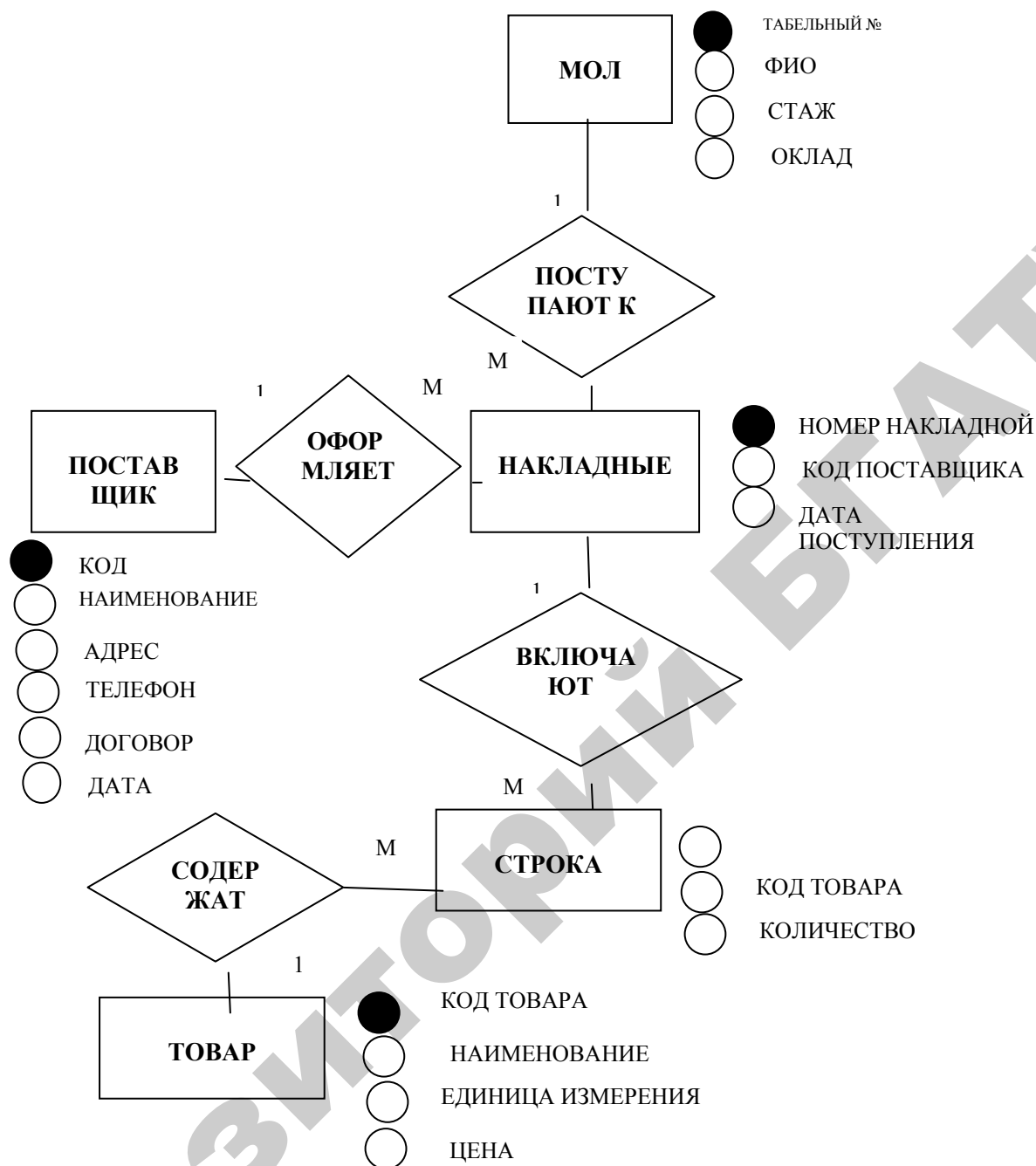


Рисунок 2.6 – Концептуальная схема базы данных КЛАДСКОЙ УЧЕТ

В результате логического проектирования получили совокупность таблиц 2.10–2.14.

1. Справочник поставщиков с именем **СпрПост**

Таблица 2.10

КодПост	НаимПост	Телефон	КонтЛицо	Договор	ДатаЗак
1	ООО Корсар	223-32-32	Жук Л. С.	345	10.01.2007
2	УП Гардарика	214-34-56	Чиж С. И.	214	15.05.2007
3	ООО Полимер	265-87-15	Иванов С. А.	658	15.05.2007
4	УП СФИНКС	234-18-45	Петров Л. С.	820	16.06.2007

Со следующей структурой записи

Имя поля	Тип данных	Описание
КодПост	Числовой	Код поставщика
НаимПост	Текстовый	Наименование поставщика
Телефон	Текстовый	Телефон
КонтЛицо	Текстовый	Контактное лицо
Договор	Числовой	Договор
ДатаЗак	Дата / время	Дата заключения

2. Справочник материально-ответственных лиц с именем **СпрМол**  
Таблица 2.11

ТабНом	ФИО	Стаж	Оклад
111	Иванов С. П.	10	400000
112	Маринич А. А.	15	500000
115	Завадский П. В.	5	350000

Со следующей структурой записи

Имя поля	Тип данных	Описание
ТабНом	Числовой	Табельный номер
ФИО	Текстовый	Фамилия имя отчество
Стаж	Числовой	Стаж
Оклад	Денежный	Оклад

3. Справочник товаров с именем **СпрТов**  
Таблица 2.12

КодТов	НаимТов	ЕдИзм	Цена
1	Бумага А4	п.	9 000
2	Бумага А3	п.	15 000
3	Файлы	шт.	50
4	Дискеты 3.5"	уп.	1 000
5	Степлер	шт.	3 000

Со следующей структурой записи

Имя поля	Тип данных	Описание
КодТов	Числовой	Код товара
НаимТов	Текстовый	Наименование товара
ЕдИзм	Текстовый	Единица измерения
Цена	Числовой	Цена

1. Таблица Накладные, с именем **Накладные**  
Таблица 2.13

НомНакл	КодПост	ДатаПост	ТабНом
1	1	11.11.2007	111
2	1	15.11.2007	111
3	2	05.12.2007	115
4	2	15.12.2007	112
5	2	20.12.2007	115
6	3	25.12.2007	111

Со следующей структурой записи

Имя поля	Тип данных	Описание
НомНакл	Числовой	Номер накладной
КодПост	Числовой	Код поставщика
ДатаПост	Дата / время	Дата поступления
ТабНом	Числовой	Табельный номер МОЛа

2. Таблица Строки накладных, с именем **СтрокиНакл**

Таблица 2.14

НомНакл	КодТов	Количество
1	1	100
1	2	100
1	5	50
2	4	60
3	2	150
3	3	1 000
4	1	120
4	5	65
5	2	200

Со следующей структурой записи

Имя поля	Тип данных	Описание
НомНакл	Числовой	Номер накладной
КодТов	Числовой	Код товара
Количество	Числовой	Количество

Информационно-логическая модель базы данных СКЛАДСКОЙ УЧЕТ изображена на рисунке 2.7.



Рисунок 2.7 – Информационно-логическая схема базы данных СКЛАДСКОЙ УЧЕТ

## РАЗДЕЛ 3 ПРАКТИЧЕСКИЕ РАБОТЫ

### Практическая работа № 1

#### Создание базы данных в Access. Построение таблиц Создание БД

Существуют три способа создания новой БД в Access.

**Первый способ.** В стартовом окне Access (рисунок 1.1) выбрать пункты *Создать файл / Новая База данных*. На экране появится окно **НОВОЙ БАЗЫ ДАННЫХ** (рисунок 3.1).

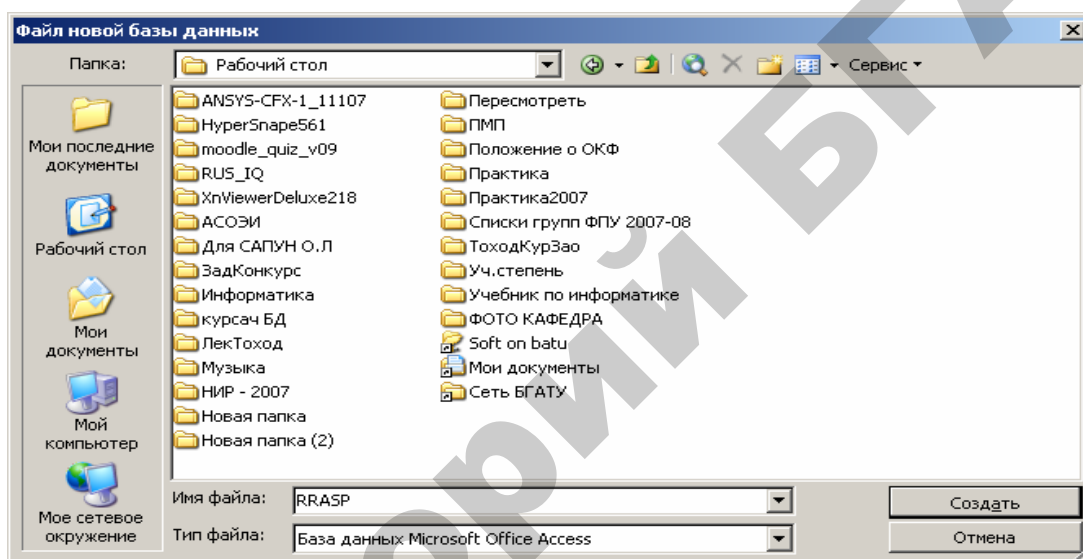



Рисунок 3.1 – Окно новой базы данных

Здесь необходимо выполнить следующие действия.

- Выбрать папку для записи файла БД: щелкнуть по  в строке ПАПКА, выбрать логический диск и на нем выбрать папку, например, D:\21эи.
- Ввести имя файла БД в соответствующую строку, например, RRASP.
- Нажать кнопку *Создать* в правом нижнем ряду кнопок. На экране появится окно базы данных (рисунок 1.2).

Далее следует создать все необходимые объекты БД: таблицы, запросы, формы, отчеты и т.д., выбирая каждый раз соответствующую вкладку в левом столбце окна.

**Замечание 3.1** Базы данных подчиняются тем же правилам именования, что и остальные файлы Microsoft Windows. Имя файла может содержать до 250 символов, в том числе пробелы, за исключением сле-

дующих знаков: \ / : \* ? ” < > | . Однако рекомендуется использовать короткие имена.

**Второй способ.** В окне Microsoft Access выбрать команды **Файл / Создать / Новая база данных**. Возникает окно файла новой БД (рисунок 3.1). Далее следует выполнить действия, описанные в первом способе.

**Третий способ.** Этот способ предполагает использование мастера БД, когда сразу описываются все объекты БД на основе стандартных объектов с учетом требований пользователя.

**Задание 1.** Создать БД с именем **RRASP** в своей личной папке любым из рассмотренных способов.

### Построение таблиц


Существует четыре способа создания таблиц в Access.

1. В процессе создания БД с помощью мастера по созданию БД.
2. С помощью МАСТЕРА по созданию таблиц, который позволяет составить таблицу из широкого списка образцов полей типовых таблиц, имеющихся в арсенале Access.
3. Пользователь самостоятельно описывает структуру таблицы в режиме КОНСТРУКТОРА, а затем вводит в описанную таблицу данные.
4. Вначале вводятся данные в пустую (неописанную) таблицу в режиме ТАБЛИЦЫ. При сохранении таблицы в соответствии с введенными значениями полей система присвоит каждому полю тип и формат, а также стандартное имя, т.е. построит МАКЕТ таблицы, который в последствии может быть изменен пользователем в режиме КОНСТРУКТОРА.


### Построение таблиц учебной базы данных


**Задание 2.** Построить таблицу **SKUR** с помощью КОНСТРУКТОРА (третьим способом).

Выполняемые действия.


- а) В окне БД (рисунок 1.2) выберем пункты **Таблицы / Создать**, в появившемся окне новой таблицы выберем **Конструктор** (или в окне БД выберем **Таблицы / Создание таблицы в режиме конструктора**). В окне КОНСТРУКТОРА (рисунок 3.2) опишем макет (структуру записи) таблицы **SKUR**. Введем имена полей, выберем для них типы. Для указания типа поля щелкнуть в колонке **Тип поля** в правой части описываемого поля, затем щелкнуть по кнопке  и выбрать из раскрывшегося списка тип (например, **Чи-**



словой). Для описания размерности числа в окне СВОЙСТВ поля (нижняя часть экрана) щелкнуть в строке **Размер поля** и с помощью кнопки  из появившегося списка выбрать необходимую характеристику. Так же установить формат поля.

б) Оставаясь в КОНСТРУКТОРЕ, объявить поле **KKUR** ключевым полем. Для этого выделить его и щелкнуть по кнопке . Ключевое поле будет помечено таким же значком. Макет таблицы изображен на рисунке 3.2.

в) Сохранить макет, выбрав **Файл / Сохранить** или с помощью кнопки . Ввести имя таблицы **SKUR**.

г) Перейти в режим ТАБЛИЦЫ с помощью кнопки  или в окне БД при выделенном имени таблицы выбрать **Режим таблицы**.

д) Ввести значения полей таблицы **SKUR** (таблица 2.6).

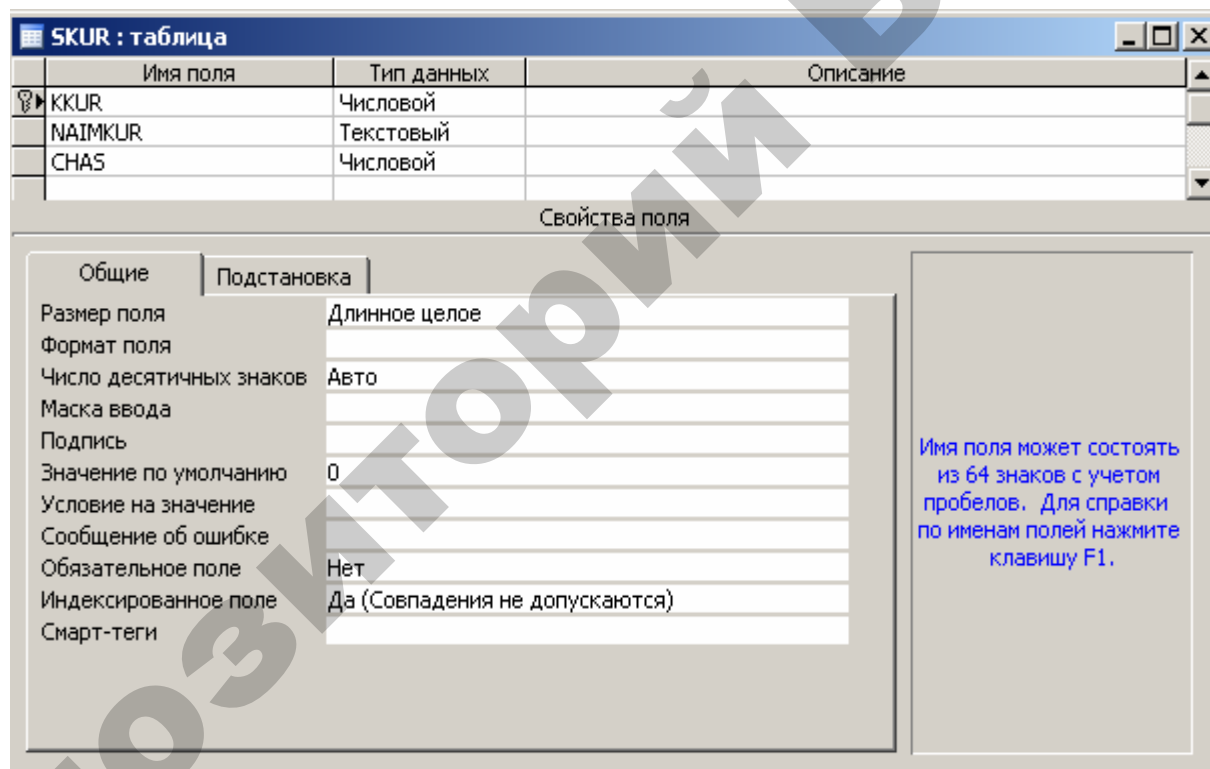


Рисунок 3.2 – Окно КОНСТРУКТОРА ТАБЛИЦ (макет таблицы SKUR)



**Замечание 3.2** Процесс ввода числовых, текстовых данных и данных типа Дата/время можно ускорить, если повторяющиеся значения в полях не набирать на клавиатуре, а вводить их из предыдущей записи нажатием комбинации клавиш CTRL – @ (амперсant).

**Замечание 3.3** В таблице активная запись обозначается треугольным маркером, а пустая запись – звездочкой. Для обозначения записи, в которую осуществляется ввод используется обозначение карандаша. Все маркеры появляются в столбце маркировки, расположенном в левой части листа данных.

**Замечание 3.4** Имена полей могут содержать до 64 символов, кроме символов: «.» (точка), «!» (восклицательный знак), «[]» (квадратные скобки), «'» (левая кавычка), но рекомендуется выбирать короткие имена полей. Пробел – разрешенный символ, но использовать его в именах объектов не рекомендуется, так как могут возникнуть проблемы с экспортом таблиц в другие приложения и при создании модулей VBA.

**Задание 3.** Построить таблицу **SGR** путем ввода данных (четвертым способом).

Выполняемые действия.

- а) В окне БД выбрать **Таблицы / Создать / Режим таблицы** или **Таблицы / Создание таблицы путем ввода данных**.
- б) Ввести данные (значения полей) в таблицу.
- в) Сохранить таблицу, введя имя **SGR**. На запрос создавать ли ключевое поле ответить НЕТ.
- г) Войти в режим **КОНСТРУКТОРА** (можно с помощью кнопки ) и откорректировать автоматически созданный макет таблицы, переименовав поля и проверив типы в соответствии с таблицей 2.4.
- д) Оставаясь в **КОНСТРУКТОРЕ**, объявить поле **KGR** ключевым полем. Для этого выделить его и выбрать команды **Правка/Ключевое поле** или щелкнуть по кнопке . Ключевое поле будет помечено таким же значком.
- е) Сохранить откорректированную таблицу.

**Задание 4.** Создать таблицы **SDY**, **SDOL**, **SKOR** одним из описанных способов.

**Замечание 3.5.** Для поля **KDOL** таблицы **SDOL** выбрать тип **Числовой**, а размер поля установить **Целое**.


**Замечание 3.6.** Во всех справочниках объектов коды объектов следует объявить ключевыми полями.

**Задание 5.** Создать структуру таблицы **RASP** с помощью **КОНСТРУКТОРА**.

Выполняемые действия:

- а) В окне БД выбрать пункты **Таблицы / Создать**, в появившемся окне новой таблицы выбрать **Конструктор** (или в окне БД выбрать **Таблицы / Создание таблицы в режиме конструктора**). В окне **КОНСТРУКТОРА** описать макет (структуру записи) таб-

лицы **RASP** в соответствии с таблицей 2.2. Ввести имена полей, выбрать для них типы.

б) Сохранить структуру таблицы, выбрав **Файл / Сохранить** или с помощью кнопки . Ввести имя таблицы **RASP**.

**Замечание 3.7.** В создаваемой таблице **RASP** не следует объявлять ключевое поле, поэтому на запрос системы о создании ключевого поля ответить *Нет*. **Внимание!** Значения полей в таблицу пока не вводить.


### Создание связей между таблицами

**Задание 6.** Установить связи между таблицами **SGR – RASP; SDY – RASP; SKUR – RASP**.

Выполняемые действия.

а) Закрывать все таблицы.

б) В окне БД нажать кнопку  или выбрать **Сервис / Схема данных**. Откроется окно СХЕМЫ ДАННЫХ.

в) Добавить в это окно связываемые таблицы (**RASP, SGR, SKUR** и **SDAY**). Для этого открыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ (если оно не открыто), выбрав команды **Связи / Добавить таблицу** или нажав кнопку  и в появившемся окне поочередно дважды щелкнуть по именам нужных таблиц (или каждую выделить и нажать кнопку **Добавить**). Макеты добавленных таблиц появятся в окне СХЕМА ДАННЫХ. Закрывать окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.

**Замечание 3.8.** Для удаления макета таблицы из окна СХЕМА ДАННЫХ следует щелкнуть по границе макета и выбрать **Правка / Удалить** или нажать клавишу DEL.

г) Для связывания таблиц выделить поле одной таблицы и при нажатой кнопке мыши переместить его на соответствующее поле другой таблицы. Например, выделить поле **KGR** в справочнике **SGR** и переместить его на поле **GR** главной таблицы **RASP**.

д) В появившемся окне ИЗМЕНЕНИЕ СВЯЗЕЙ можно изменить некоторые свойства связи, уточнить связь (ее тип **один – ко – многим**), можно установить флажки в строках **Обеспечение целостности данных, Каскадное обновление записей и Каскадное удаление записей**. В этом случае, например, при изменении или удалении записей, касающихся, группы 2эи из **SGR**, записи по группе 2эи автоматически изменяются или удаляются и из таблицы **RASP**. Установить свойство **Обеспечение целостности данных, Каскад-**

*ное обновление записей* и *Каскадное удаление записей* для создаваемой связи.

е) Создать связь нажатием кнопки *Создать*.

ж) Аналогично создать еще две связи, переместив поле **KDN** таблицы **SDY** на поле **DAY** таблицы **RASP** и поле **KKUR** из **SKUR** на поле **KKUR** в **RASP**. Установить свойства *Обеспечение целостности данных*, *Каскадное обновление записей* и *Каскадное удаление записей* для всех связей.

Общий вид окна схемы данных изображен на рисунке 3.3.

**Замечание 3.9.** Три созданные связи имеют тип **один – ко – многим**, поскольку одной записи в таблице **SDY** соответствует много записей в таблице **RASP**, одной записи в таблице **SGR** соответствует много записей в таблице **RASP**, и одной записи в таблице **SKUR** соответствует много записей в таблице **RASP**.

**Замечание 3.10.** В большинстве случаев связывают **КЛЮЧЕВОЕ ПОЛЕ** одной таблицы, например, **KDN** таблицы **SDY** с соответствующим ему полем другой таблицы (часто имеющим одно и то же имя), которое называют **полем внешнего ключа**. У нас это поле **DAY** в таблице **RASP**.

**ВНИМАНИЕ!** Связываемые поля должны содержать данные одного и того же типа.

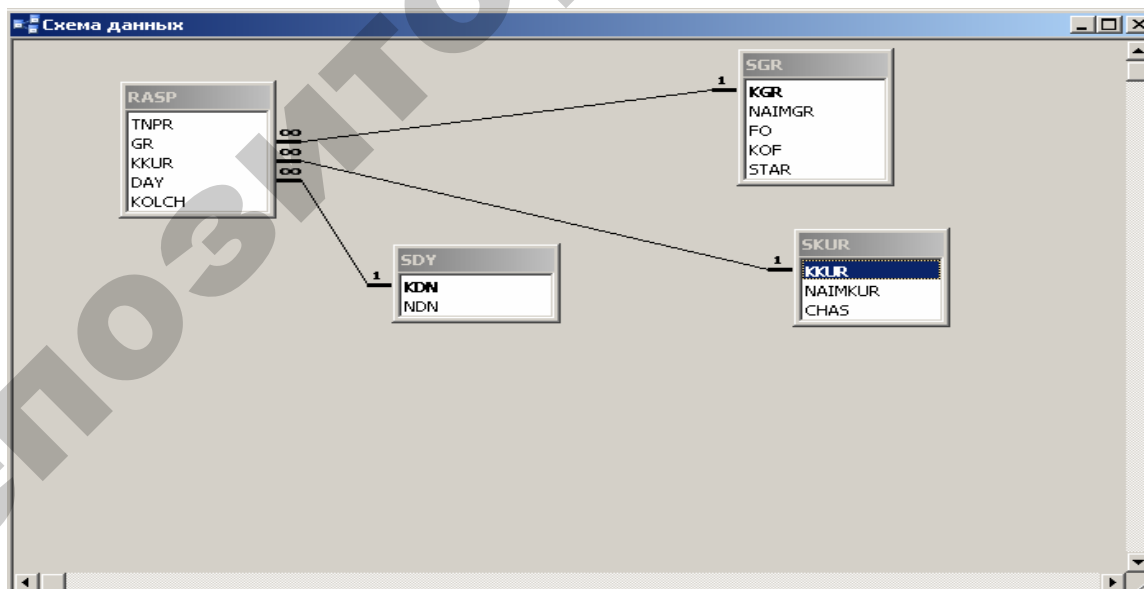


Рисунок 3.3 – Окно СХЕМА ДАННЫХ

**Замечание 3.11** Для вывода на экран всех существующих в БД связей между таблицами следует в окне СХЕМА ДАННЫХ (рисунок 3.3) выполнить следующие действия:

а) Набрать **Правка / Очистить макет**.

б) Нажать на панели инструментов кнопку  для вывода всех существующих связей.

Для просмотра связей определенной таблицы в окне СХЕМА ДАННЫХ выполнить:

а) **Правка / Очистить макет**.

б) Добавить нужную таблицу и нажать кнопку .

**Замечание 3.12** Кроме описанных выше приемов в окне СХЕМА ДАННЫХ пользователь имеет возможность выполнить следующие действия:

1. Изменить макет таблицы.

а) Установить указатель на таблицу, нажать правую кнопку мыши и выбрать **Конструктор таблиц**.

б) Внести изменения в макет.

в) Сохранить изменения и выйти из КОНСТРУКТОРА.

2. Изменить существующую связь. Для этого дважды щелкнуть по линии связи или при выделении линии связи выбрать команды **Связи / Изменить связь** и в появившемся окне ИЗМЕНЕНИЕ СВЯЗЕЙ изменить параметры.

3. Удалить связь. Для этого выделить связь и набрать **Правка / Удалить** или нажать кнопку DEL.

**Задание 6.** Заполнить таблицу **RASP** значениями данных.

Выполняемые действия:

а) Открыть таблицу **RASP** в режиме таблицы.

б) Ввести значения полей в таблицу в соответствии с таблицей 2.2. Обратите внимание на то, что при попытке ввода в поле, например, **KKUR** значения кода курса 9, которого нет в справочнике курсов **SKUR**, система выведет сообщение об ошибке и не позволит ввести значение. Так поддерживает целостность данных созданная связь таблиц **SKUR – RASP** (в соответствии с заданным свойством связи **Обеспечение целостности данных**).

**Задание 7.** Создать таблицу **SPOD** с помощью КОНСТРУКТОРА (третьим способом), присвоить полю **KORP**, тип ПОДСТАНОВКА для выбора значений этого поля из справочника корпусов (**SKOR**).

Чтобы присвоить полю **KORP** тип ПОДСТАНОВКА следует выполнить действия:

а) в списке типов полей поля **KORP** выбрать **Мастер подстановок**;

б) в открывшемся окне СОЗДАНИЕ ПОДСТАНОВКИ выбрать пункт.



**Объект «столбец подстановки» будет использовать значение из таблицы или запроса.** Нажать *Далее*.

в) В следующем окне в строке *показать* выбрать **Таблицы** и выделить таблицу **SKOR**, из которой следует выбирать значения в описываемое поле. Нажать *Далее*.

г) В следующем окне переместить из *доступных полей* в *выбранные поля*, которые будут помещены в столбец подстановки: поле **KKOR**, из которого будут выбираться значения, и поле **NAIMKOR** для наглядности. Нажать *Далее*.

д) в следующем окне указать поле, по которому следует выполнить сортировку (**KKOR**).

е) В следующем окне с помощью перемещения правой границы установить нужный размер полей столбца подстановки. Нажать *Далее*.

ж) В следующем окне задать подпись, которую содержит столбец подстановки (**KORP**). Нажать *Готово*.

На запрос *Перед созданием связи следует сохранить эту таблицу, сделать это сейчас?* Ответить *Да*, задать имя таблицы **SPOD**.

**Задание 8.** Убедиться в наличии связи между таблицами **SPOD** и **SKOR**.

а) Открыть окно СХЕМА ДАННЫХ с помощью кнопки .

б) Добавить в схему данных таблицы **SPOD** и **SKOR**. Список таблиц выводится на экран кнопкой .


в) Убедиться, что таблицы появились со связью, которая создавалась вместе с созданием поля **KORP** типа *Подстановка* таблицы **SPOD**.

## Мастер таблиц

**Задание 9.** Создать таблицу **SPR** (справочник преподавателей) с помощью **МАСТЕРА ТАБЛИЦ** (вторым способом).

Выполняемые действия:

а) Набрать в окне БД **Таблицы / Создать / Мастер таблиц** или **Таблицы / Создание таблицы с помощью мастера**.

б) Из всех предложенных образцов таблиц выбрать таблицу **Сотрудники** (отметить). Далее из колонки **Образцы полей** выбрать поле **Табельный номер** и переписать его в новую таблицу кнопкой . Затем выбрать вкладку **Переименовать** и ввести новое имя поля. Нажать **ОК**. Так же переместить и переименовать поле **Фамилия**. Для полей **KDOL** и **PSTAG** перенести и переименовать поле **Выводы**, а для поля **KPOD** – поле **КодОтдела**. Нажать *Далее*.

в) В следующем окне задать имя для новой таблицы **SPR** и выбрать пункт *Пользователь определяет ключ самостоятельно*. Нажать *Далее*.

г) На запрос *Выбери поле с уникальными для каждой записи данными* выбрать поле **TNPR**. На запрос *Какие данные должны содержаться в ключевом поле* выбрать ответ **Числа, вводимые пользователем при добавлении новых записей**. Нажать *Далее*.

д) В появившемся окне связей следует выбрать строку, где упоминается таблица, с которой необходимо установить связь. В нашем случае **RASP** (*Не связано с RASP*) и нажать *Связи*.

е) В новом окне СВЯЗИ выбрать пункт *Одной записи таблицы SPR соответствует много записей таблицы RASP* и нажать **ОК**. Произойдет возврат в предыдущее окно, где появится надпись *Связано с RASP*. Аналогично следует убедиться в установлении связей между таблицами **SPR–SDOL** и **SPR – SPOD**, эти связи установилась автоматически по полям с одинаковыми именами **KDOL** и **KPOD**. Нажать *Далее*.

ж) В следующем окне выбрать одно из трех действий:

1. *Изменение структуры таблицы.*

2. *Непосредственный ввод данных в таблицу.*

3. *Ввод данных в таблицу с помощью формы, созданной мастером.*

Выберем пункт 2 и *Готово*. Далее следует ввести данные в таблицу и сохранить ее.

з) Убедиться, что таблица **SPR** создана и связана с таблицами **RASP**, **SPOD** и **SDOL**. Для этого в окне СХЕМА ДАННЫХ добавить таблицу **SPR**.

**Задание 10.** *Установить свойство обеспечения целостности данных для всех связей.*

Для этого в окне СХЕМА ДАННЫХ открыть окно ИЗМЕНЕНИЕ СВЯЗЕЙ поочередно для каждой связи двойным щелчком по линии связи и установить флажки **Обеспечение целостности данных**, **Каскадное обновление записей** и **Каскадное удаление записей**. Общий вид окна СХЕМА ДАННЫХ приведен на рисунке 3.4.

**Замечание 3.13** В окне СХЕМА ДАННЫХ в структурах таблиц поля, объявленные ключами, выделены жирным шрифтом.

**ВНИМАНИЕ!** Проконтролировать, чтобы во всех справочных таблицах коды были объявлены ключами.

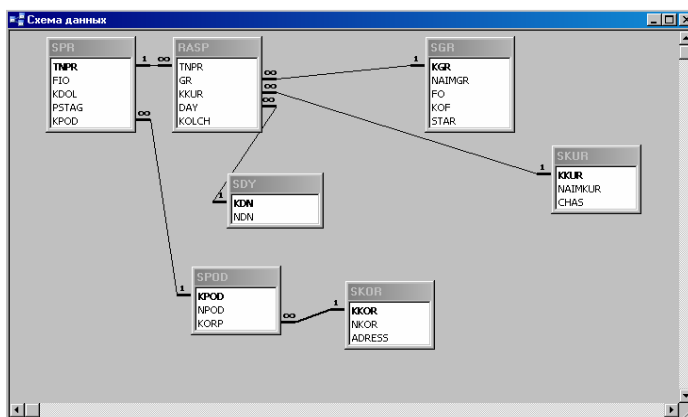


Рисунок 3.4 – Окно СХЕМА ДАННЫХ, связи между таблицами

### Задания для самостоятельного выполнения к практической работе № 1

1) Создать в личной папке базу данных **СкладскойУчет** одним из описанных ранее способов.

2) Создать таблицу **СпрПост** с помощью конструктора. Поле **КодПост** объявить ключом.

**Указание.** Для поля **Телефон** Установить маску ввода 999-99-99. Для поля **ДатаЗак** задать тип *Дата/время*, формат *Краткий формат даты*, установить маску ввода *Краткий формат даты*.

3) Создать таблицу **СпрТов** путем ввода данных. Поле **КодТов** объявить ключом.

4) Создать структуры таблиц **Накладные** (ключ – **НомНакл**) и **СтрокиНакл** (ключ не объявлять) с помощью КОНСТРУКТОРА.

**Указание.** Для поля **ДатаПост** задать тип *Дата/время*, формат *Краткий формат даты*, установить маску ввода 99.99.9999.

5) Установить связи между таблицами. Для всех связей установить свойства *Обеспечение целостности данных*, *Каскадное обновление записей* и *Каскадное удаление записей*.

6) Заполнить таблицы **Накладные** и **СтрокиНакл** данными в соответствии с таблицами 2.13, 2.14. Проверить установленные свойства связей *Обеспечение целостности данных*.

7) Создать таблицу **СпрМол** с помощью мастера таблиц. В процессе установить связь таблицы **СпрМол** с таблицей **Накладные**.

**Указание.** За основу взять стандартную таблицу *Сотрудники*, перенести и переименовать поля в соответствии со структурой. Для поля **Стаж** перенести и переименовать поле **Выводы**, для поля **Оклад** – поле **Зарплата**. Добавить таблицу **СпрМол** в СХЕМУ ДАННЫХ.

### Задания для практической работы



Создать таблицы для решения задачи согласно варианту.



## ПРАКТИЧЕСКАЯ РАБОТА № 2

### Редактирование базы данных. Обработка таблиц Копирование объектов в MS Access

**Задание 1.** *Открыть базу данных RASP. Сделать копию таблицы RASP с именем RASP1.* Выполняемые действия :

- а) В окне БД на вкладке **Таблицы** выделить **RASP**.
- б) На панели инструментов нажать последовательно две кнопки  (Копировать) и  (Вставить).
- в) Указать нужные сведения (в том числе имя **RASP1**) в окне диалога ВСТАВКА ОБЪЕКТА и нажать ОК. В БД появилась новая таблица **RASP1**.

**Замечание 4.1** Получить копию объекта в Access можно еще следующим образом. В окне БД при выделенном имени объекта выполнить команды **Файл / Сохранить как...** и ввести новое имя объекта.

**Замечание 4.2** Для копирования объекта из одной БД в другую кнопку **Копировать** следует нажать в окне первой БД (при выделенном копируемом объекте), а кнопку **Вставить** – в окне другой БД.

**Замечание 4.3** Для удаления таблицы из БД нужно выделить ее имя в окне БД и нажать клавишу DEL.

### Изменение размеров и порядка следования столбцов таблицы на экране, изменение макета таблицы

- а) Чтобы изменить ширину столбца, нужно передвинуть с помощью мыши границу столбца. Высота строки меняется аналогично перемещением граничной линии.
- б) Чтобы поменять местами столбцы, следует выделить перемещаемый столбец и с помощью мыши передвинуть его в то место, где он должен находиться.
- в) Чтобы скрыть столбец на экране, следует его правую границу переместить на левую.

## Изменение макета таблицы

Макет таблицы изменяется в режиме КОНСТРУКТОРА. Поля таблицы можно перемещать, удалять, вставлять, добавлять; кроме того, можно изменять их имена, типы, свойства.



г) Чтобы поменять местами поля в макете таблицы, следует перемещаемое поле выделить и передвинуть в нужное место.

д) Чтобы изменить формат числового поля следует выделить поле, вызвать *Свойства поля* и в пункте *Формат поля* выбрать соответствующий формат. Аналогично изменяются другие свойства поля.

**Задание 2.** Выполнить изменения размеров таблицы *RASP1* (пункты а, б, в) и изменить макет таблицы *RASP1*: поменять местами поля *TNPR* и *GR*, изменить формат поля *KOLCH*.


## Сортировка и поиск данных в таблице

**Задание 3.** Таблицу *SDY* рассортировать в алфавитном порядке наименования дней.

Для этого выделить столбец **NDN** и нажать кнопку  ,  . Записи рассортировались, и если теперь сохранить таблицу, то она останется с новой сортировкой. Для сортировки по убыванию следует нажать кнопку **Поиск в таблице**.

**Задание 4.** Выполнить следующие виды поиска в таблице *RASP*:

а) Найти запись с конкретным номером, например, № 7 в таблице *RASP*. Для этого следует в окне таблицы в поле *Номера записи* установить указатель мыши на номер записи и дважды щелкнуть мышью, после чего ввести 7 и нажать клавишу ENTER.

б) Найти запись с конкретным значением поля, например, в таблице *RASP* запись со значением 3 поля **DAY**. Для этого следует в таблице *RASP* выделить столбец **DAY**; нажать кнопку  (бинокль) на панели инструментов; ввести искомое значение 3 в поле *Образец*; определить другие необходимые параметры в окне диалога ПОИСК; нажать *Найти*. Найдется первое вхождение данного значения поля. Для поиска следующего вхождения нажать *Найти далее*.



## Использование фильтров в таблицах

С помощью фильтров можно ограничить отображение данных в таблице только теми записями, которые удовлетворяют заданному критерию.

Фильтрация осуществляется двумя способами: на основании заданного значения поля и на основании выделенной ячейки.





**Задание 5.** *Используя фильтр по выделенному, из таблицы SPOD выбрать сведения о подразделениях, расположенных в корпусе № 1.*

Выполняемые действия:

- а) Открыть таблицу **SPOD** в режиме таблицы.
- б) Поместить курсор в колонку **KORP** и выделить значение «1».
- в) Выбрать команды **Записи / Фильтр / Фильтр по выделенному**. Или нажать кнопку  (фильтр по выделенному). На экране отразится таблица в заданном виде.
- г) Удалить фильтр, командами **Фильтр / Удалить фильтр** или кнопкой .

**Задание 6.** *Из таблицы SGR выбрать сведения о группах дневной и заочной форм обучения. Воспользоваться обычным фильтром.*

Выполняемые действия:

- а) Открыть таблицу **SGR** в режиме таблицы.
- б) Изменить фильтр, выполнив команды **Записи / Фильтр / Изменить фильтр** или нажав на кнопку . Появится окно с заголовками полей таблицы.
- в) Очистить бланк фильтра по командам **Правка / Очистить бланк** или кнопкой .
- г) На вкладке фильтра **Найти** (в нижней части окна) щелкнуть в клетке столбца **FO** и из раскрывающегося списка выбрать «Д».
- д) Перейти на вкладку фильтра **Или**, в клетке столбца **FO** из раскрывающегося списка выбрать «3».
- е) Применить фильтр, выбрав команды **Фильтр / Применить фильтр** или нажав на кнопку .
- ж) Проанализировать результаты фильтрации.
- з) Удалить фильтр, командами **Фильтр / Удалить фильтр** или кнопкой .

**Задание 7.** *Из таблицы SPR выбрать сведения о преподавателях, работающих в подразделении 1, педагогический стаж которых составляет 15 и более лет. Вывести записи в алфавитном порядке фамилий.*

Эту задачу можно решить, используя расширенный фильтр. Выполняемые действия:

- а) Открыть таблицу **SPR** в режиме таблицы.
- б) Выбрать команды *Записи / Фильтр / Расширенный фильтр*. Появится окно конструктора фильтра.
- в) Очистить бланк фильтра по командам *Правка / Очистить бланк* или кнопкой
- г) Далее в бланк фильтра в строку **Поле** из списка полей таблицы переместить поля: **FIO, PSTAG, KPOD**.
- д) Задать сортировку записей в алфавитном порядке фамилий. Для этого щелкнуть в клетке на пересечении колонки **FIO** и строки **Сортировка** и из раскрывающегося списка выбрать строку *по возрастанию*.
- е) Задать условия отбора записей, для этого в строку **Условие отбора** в колонку **PSTAG** ввести условие  $\geq 15$ , а в колонку **KPOD** ввести **1**.
- ж) Применить фильтр, выбрав команды *Фильтр / Применить фильтр* или нажав на кнопку
- з) Проанализировать результаты фильтрации.
- и) Удалить фильтр, командами *Фильтр / Удалить фильтр* или кнопкой

### Задания для самостоятельного выполнения к практической работе № 2

После успешного выполнения заданий 1–7, описанных ранее, самостоятельно выполнить следующие задания в базе данных **RRASP**:

- 1) Откорректировать макет таблицы **RASP1**, поменяв местами поля **GR** и **DAY**. Вставить еще одно числовое поле **AUD** (аудитория) перед полем **KOLCH**. Заполнить его данными.
- 2) Рассортировать записи таблицы **RASP1** в порядке убывания поля **GR**.
- 3) Осуществить поиск в таблице **RASP1** по значению 2 поля **TNPR**. Просмотреть все вхождения значения.
- 4) Удалить таблицу **RASP1**.
- 5) Осуществить корректировку БД следующим образом:
  - а) Добавить в таблицу **SPR** запись

<b>TNPR</b>	<b>FIO</b>	<b>KDOL</b>	<b>PSTAG</b>	<b>KPOD</b>
5	Сидоров	3	18	1

Добавить в таблицу **RASP** три записи по Сидорову, введя любые дни, группы и курсы, содержащиеся в справочниках.

б) Изменить в таблице **SPR** табельный номер Сидорова (поле **TNPR**) на 6. Заккрыть таблицу. Убедиться, что в записях таблицы **RASP**, касающихся Сидорова, произошла аналогичная замена (поле **TNPR** приняло значение 6). Осуществилось каскадное обновление связанного поля, заданное при установке связи между таблицами.

в) Удалить из таблицы **SPR** запись о Сидорове, выделив эту запись и нажав кнопку **DEL** на клавиатуре. Открыть таблицу **RASP** и убедиться, что из нее также удалились записи, касающиеся Сидорова. Произошло каскадное удаление связанных записей, заданное при установке связи между таблицами.

б) Заккрыть базу данных **RRASP**.

### Задания в базе данных Складской Учет

Открыть базу данных **Складской Учет** и в ней выполнить следующие задания:

- 1) Используя Фильтр по выделенному, выбрать информацию о поставщиках, с которыми договор заключен 15.05.2007.
- 2) С помощью обычного фильтра выбрать накладные, поступившие от поставщика с кодом 2 и принятые сотрудником с табельным номером 115.
- 3) Применив расширенный фильтр, вывести в алфавитном порядке наименований записи, содержащие информацию о товарах, цена которых составляет 9 000 рублей и более, а единица измерения – «п.».

## ПРАКТИЧЕСКАЯ РАБОТА № 3

### Формирование запросов

#### Основные сведения о запросах

Запросы предназначены для просмотра, анализа и изменения информации в БД. Кроме того, они используются в качестве источника данных для форм и отчетов. Существует несколько видов запросов.

1. **Запросы на выборку**, позволяющие выбирать данные, соответствующие условиям отбора, из одной или нескольких таблиц; включать в результирующую таблицу лишь нужные поля в нужном порядке; выполнять необходимую сортировку записей результирующей таблицы; производить вычисления над полями записей.

2. **Запросы действия** – запросы, в результате выполнения которых изменяется сама БД. К их числу относятся:

- запрос на создание таблицы, который создает новую таблицу на основе всех или части данных из одной или нескольких таблиц;
- запрос на удаление записей, который удаляет группу записей из одной или нескольких таблиц;
- запрос на добавление записей, который добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц;
- запрос на обновление записей, который вносит общие изменения в группу записей одной или нескольких таблиц.

3. **Перекрестные запросы**. Позволяют отображать результаты статистических расчетов (такие как суммы, количество записей, среднее значение), выполняемых по данным из одного поля. Эти результаты группируются по двум наборам данных в форме перекрестной таблицы. Первый набор выводится в левом столбце и образует заголовки строк, а второй – выводится в верхней строке и образует заголовки столбцов. Например, можно посчитать общее время занятий, проводимых каждым преподавателем в каждой группе в виде:

ФИО	3 эи	2 эи	1 эи
Иванов	8	6	5
Петров	6		6

4. **Запросы с параметрами**. Позволяют при их выполнении ввести определенные сведения (параметры), например, условие отбора записей или значение для вставки в поле. Этот тип запроса удобно


использовать для выбора информации при постоянно изменяющихся условиях отбора.

Access позволяет создать запросы самостоятельно (с помощью КОНСТРУКТОРА) и с помощью программ-мастеров.

Имеется четыре мастера запросов:

- Мастер ПРОСТОЙ ЗАПРОС позволяет выводить нужные записи и нужные поля из источника запроса и при необходимости подвести итоги.
- Мастер ПЕРЕКРЕСТНЫЙ ЗАПРОС позволяет строить перекрестный запрос.
- Мастер ПОВТОРЯЮЩИЕСЯ ЗАПИСИ позволяет найти записи с повторяющимися значениями полей, в том числе и дублирующиеся записи.
- Мастер ЗАПИСИ БЕЗ ПОДЧИНЕННЫХ позволяет найти записи в одной таблице, не имеющие подчиненных записей в другой таблице.

### Создание запросов в режиме конструктора

**Действие 1.** В окне БД выбрать *Запросы / Создать*. В появившемся окне НОВЫЙ ЗАПРОС выбрать *Конструктор*. На экране появляется окно ДОБАВЛЕНИЕ ТАБЛИЦЫ. Если не появилось, то вызвать его с помощью кнопки  или по команде *Запрос / Добавить таблицу*.

**Действие 2.** В окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить все таблицы, участвующие в запросе (например, **RASP, SPR, SGR, SDY, SPOD**). Добавить таблицу можно двумя способами: либо выделить ее имя и нажать кнопку *Добавить*, либо дважды щелкнуть по ее имени. После добавления таблиц закрыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ. На экране сформируется окно КОНСТРУКТОРА ЗАПРОСОВ (рисунок 3.5).



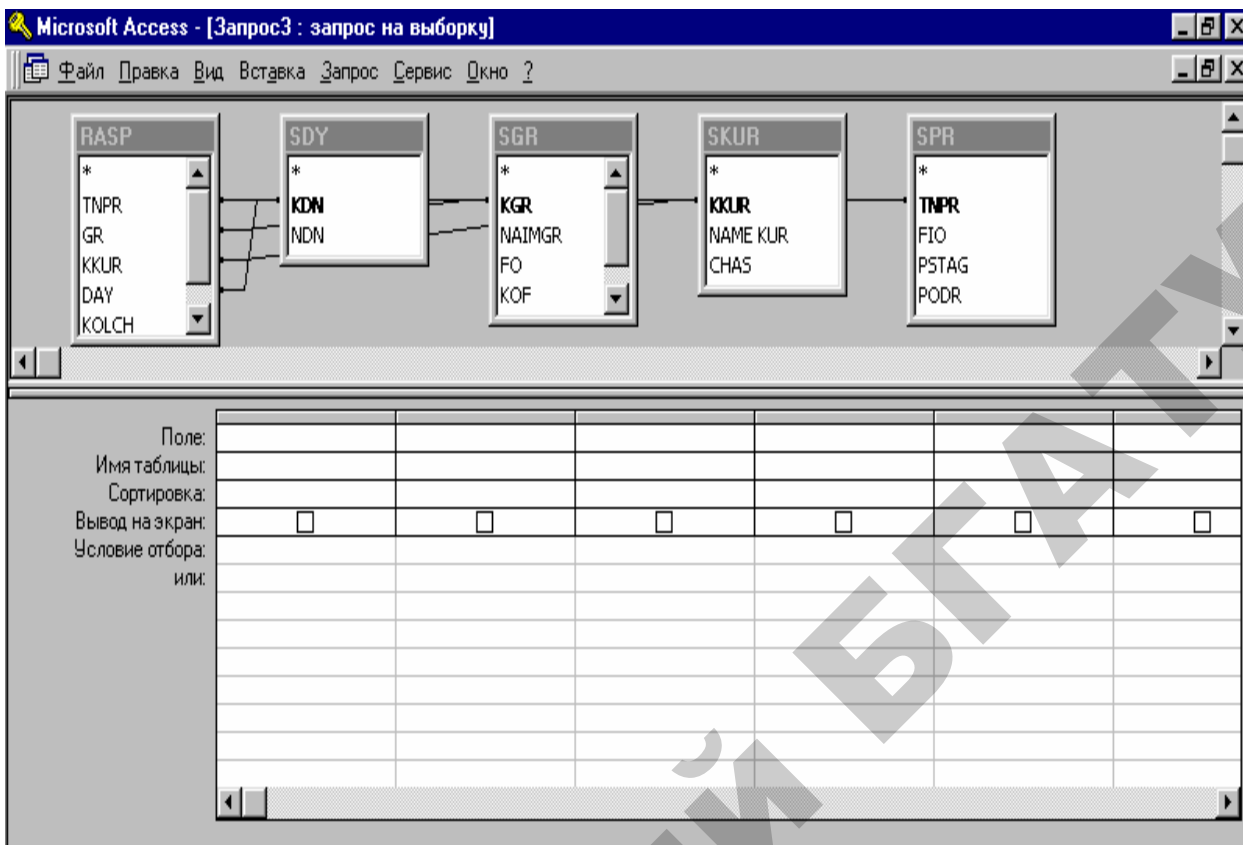


Рисунок 3.5 – Окно КОНСТРУКТОРА ЗАПРОСОВ

Это окно разделено на две части. В верхней части размещена схема данных запроса, которая содержит выбранные в качестве источников создаваемого запроса ТАБЛИЦЫ и ЗАПРОСЫ, представленные списком полей. Схема данных отображает связи между источниками, а также позволяет установить новые связи. Нижняя часть – это макет (бланк) запроса, который нужно заполнить. Каждому полю в запросе соответствует один столбец в бланке запроса. Для него предусмотрены следующие строки:

1. **Поле** указывает имя поля, участвующее в запросе.
2. **Имя таблицы** указывает, какой таблице принадлежит поле, участвующее в запросе.
3. **Сортировка** позволяет отсортировать записи результирующей таблицы.
4. **Вывод на экран** позволяет управлять отображением поля в результирующей таблице. Для отображаемых в результирующей таблице полей в этой строке должны быть включены флажки.
5. **Условие отбора** позволяет задать условие отбора записей по значению поля.



6. **Или** позволяет объединять условия отбора в логической операции «или». Эта строка может состоять из нескольких строк бланка запроса.

**Действие 3.** Для включения нужных полей в бланк запроса необходимо выполнить одно из следующих действий:

1. Перенести (с помощью мыши) нужное поле из списка полей в верхней панели в первую свободную клетку строки **ПОЛЕ**.
2. Дважды щелкнуть по имени нужного поля в списке полей.
3. Щелкнуть в клетке строки **Поле**, из раскрывшегося списка полей выбрать нужное.

Для перемещения в бланк запроса всех полей сразу можно выполнить одно из следующих действий:


1. Выделить их, дважды щелкнув по имени нужной таблицы, и переместить в строку **Поле**.
2. Перетащить в строку **Поле** звездочку, стоящую на первом месте в списке полей таблицы. В этом случае в списке полей появится только имя таблицы со звездочкой, но в результат выполнения запроса будут включены все поля.

Для удаления поля из бланка запроса следует отметить его (щелкнуть мышью в области выше имени поля) и нажать клавишу DEL.

**Действие 4.** Проверить признаки отражения полей на экране. В строке **Вывод на экран** проверить наличие флажков V в тех полях, которые нужно видеть в результирующей таблице. При создании флажки устанавливаются автоматически, и для исключения поля из результирующей таблицы флажок следует убрать.



**Замечание 3.1** При закрытии запроса все скрытые поля перемещаются в правые столбцы бланка.

**Действие 5.** Сохранение запроса можно выполнить одним из способов:

1. С помощью команд **Файл / Сохранить**.
2. Нажать кнопку  на панели инструментов.
3. Закрыть запрос и на вопрос о сохранении запроса ответить «ДА».

При сохранении запроса в поле ввода **Имя запроса** ввести имя запроса, отражающее его смысл. После сохранения запроса, его имя появляется в окне БД во вкладке **Запросы**.

**Действие 6.** Просмотреть результирующую таблицу (выполнить запрос) из окна **КОНСТРУКТОРА** можно одним из следующих способов.

1. Командой **Запрос / Запуск** или нажатием кнопки .
2. Командой **Вид / Режим таблицы**.
3. Нажав кнопку  на панели инструментов.

Из окна БД выполнить запрос можно следующими действиями:

- a. выделить имя запроса и щелкнуть по кнопке **Открыть**;
- b. дважды щелкнуть по имени запроса.

### **Создание запроса выбора по одной таблице**

**Задание 1.** *Вывести следующие сведения о группах: наименование, форма обучения, фамилия старосты. Запрос сохранить с именем ZAPR1.*

Нужные сведения содержатся в таблице **SGR**, и по ней следует создавать запрос.

Необходимые действия:

1. В окне БД выбрать **Запросы / Создать / Конструктор**.
2. В появившемся окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицу **SGR**.
3. Переместить поля **NAIMGR, FO, STAR** в бланк запроса.
4. Проверить наличие флажка **Вывод на экран (V)** для всех полей запроса.
5. Выполнить запрос одним из описанных способов.
6. Сохранить запрос с именем **ZAPR1**.

### **Создание запроса на основе нескольких таблиц с применением сортировки**

**Задание 2.** *С помощью КОНСТРУКТОРА создать запрос ZAPR2, результирующая таблица которого имела бы структуру записи, подобную структуре записи таблицы RASP, но объекты должны быть представлены своими наименованиями (взятыми из справочников). В результирующую таблицу вывести все записи таблицы RASP. Произвести сортировку по следующим полям.*

1. **ПРЕПОДАВАТЕЛЬ (FIO)** в алфавитном порядке (возрастание).
2. **ГРУППА (NAIMGR)** убывание.

*Это означает, что по одному преподавателю записи должны следовать в порядке убывания групп.*

Необходимые действия:

1. В окне БД выбрать вкладки **Запросы / Создать**. В появившемся окне НОВЫЙ ЗАПРОС выбрать **Конструктор** (или выбрать **Запросы / Создание запроса в режиме конструктора**).

2. В окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицы **RASP**, **SGR**, **SPR**, **SDY**, **SKUR**. Закрыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.
3. На экране появилось окно конструктора запросов, в верхней части которого видны макеты всех добавленных таблиц с установленными связями. В нижней части содержится пустой макет создаваемого запроса. В строку ПОЛЕ макета следует переместить поля, включаемые в результирующую таблицу: **FIO** из **SPR**, **NAIMGR** из **SGR**, **NAIMKUR** из **SKUR**, **NDN** из **SDY**, **KOLCH** из **RASP**.
4. В строке *Вывод на экран* проверить наличие флажков (V) для всех полей.
5. Сохранить и просмотреть запрос.

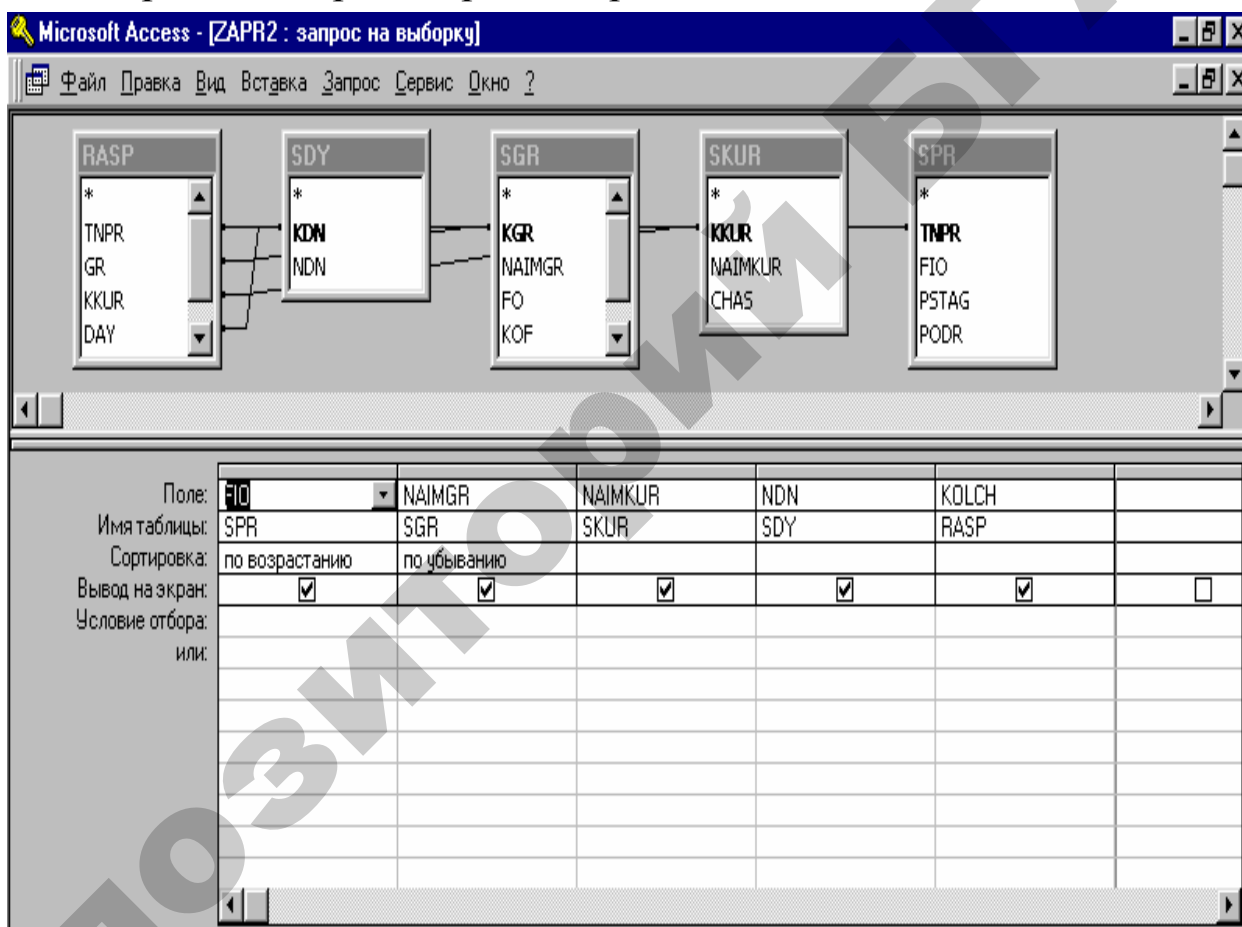


Рисунок 3.6 – Бланк запроса ZAPR2

6. Задание порядка сортировки запроса. В сортировке может участвовать до 10 полей.
  - Следует разместить поля в бланке запроса в порядке выполнения сортировки. Сортировка начинается с самого левого поля.
  - В бланке запроса в строке *Сортировка* для поля, по которому следует рассортировать, нажать кнопку РАСКРЫТИЯ СПИСКА и выбрать порядок сортировки: *По возрастанию* или *По убыванию*.

В нашем запросе для поля **ФИО** выбрать *По возрастанию*, для поля **NAIMGR** – *По убыванию*. Окончательный вид бланка запроса изображен на рисунке 3.6.

7. Сохранить и выполнить запрос.

### Создание запросов с условиями отбора

Условия отбора, позволяющие выбрать только определенные записи таблицы, задаются в строках *Условие отбора*, *или* и могут представлять из себя обычные выражения типа сравнения. В выражениях могут использоваться логические операторы NOT, AND, OR, а также конструкция BETWEEN (между).

Если выражения вводятся в несколько клеток одной строки *Условие отбора*, то они автоматически объединятся с помощью логического оператора AND. Выражения же, введенные в разные строки (*Условия отбора* и *или*) объединятся с помощью логического оператора OR.

**Задание 3.** Создать запрос **ZAPR3**, структура результирующей таблицы которого идентична **ZAPR2**, но в таблицу включить только строки, касающиеся среды.

Поле:	FIO	NAIMGR	NAIMKUR	NDN	KOLCH	
Имя таблицы:	SPR	SGR	SKUR	SDY	RASP	
Сортировка:	по возрастанию	по убыванию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:				"Среда"		
или:						

Рисунок 3.7 – Бланк запроса **ZAPR3**

Выполняемые действия:

1. Скопировать **ZAPR2** с именем **ZAPR3** (последовательно нажав кнопки *Копировать* и *Вставить*).
2. Открыть **ZAPR3** в режиме КОНСТРУКТОРА и в строку *Условие отбора* графы **NDN** ввести надпись «Среда». Макет **ZAPR3** изображен на рисунке 3.7.
3. Сохранить и выполнить запрос.

**Задание 4.** Создать запрос **ZAPR4**, выбирающий строки расписания за среду, с количеством часов от 2 до 6 включительно.

Выполняемые действия:

1. Создать запрос, аналогичный **ZAPR3**, но в строку *Условие отбора* макета в колонку **KOLCH** добавить условие  $\geq 2 \text{ AND } \leq 6$ . (Это же условие можно записать так **BETWEEN 2 AND 6**).
2. Сохранить запрос с именем **ZAPR4**.

### Создание запросов с параметрами

**Задание 5.** Создать запрос, позволяющий просмотреть расписание только одного любого дня недели.



Необходимые действия:

1. Создать запрос **ZAPR5** по аналогии с **ZAPR3**, но в строку *Условие отбора* графы **NDN** вместо надписи «Среда» ввести приглашение на ввод условия отбора в квадратных скобках, например, [Введите день недели]. Получился запрос с параметром. При выполнении запроса, перед формированием таблицы будет выводиться заданное приглашение: «Введите день недели». И, вводя нужный день, можно получить расписание занятий в этот день.
2. Выполнить и сохранить запрос.

**Замечание 3.2** Если вместо приглашения [Введите день недели] ввести выражение **Like[Введите первую букву наименования]&"\*"**, то можно выбрать записи, касающиеся дней недели, начинающихся с введенной буквы. А если ввести выражение **Like"\*"&[Введите букву, содержащуюся в наименовании]&"\*"**, то будут выводиться записи по дням, содержащим в наименовании введенную букву.




### Применение фильтров в запросах

Для ограничения объема данных, отображаемых запросом, можно так же применять фильтр. Фильтр выполняет ту же задачу, что и условия отбора, хотя в более ограниченном виде. Применение фильтра не требует модификации запроса и осуществляется в ре-

жиме выполнения запроса кнопками  (*Изменить фильтр*) и  (*Применить фильтр*) или командами *Записи/Фильтр/Изменить фильтр* и *Фильтр/Применить фильтр*.

**Задание 6.** *Применив к запросу ZAPR2 фильтр, выбрать из расписания информацию по преподавателю Петрову.*

Рекомендуемые действия:

1. Выполнить запрос **ZAPR2**.
2. Выделить колонку **ФИО**.
3. Выбрать команды *Записи/Фильтр/Изменить фильтр* или нажать кнопку .
4. В раскрывшемся окне фильтра в поле **ФИО** щелкнуть по правой части и выбрать значение для фильтрации **Петров**.
5. Выбрать команды *Фильтр / Применить фильтр* или нажать кнопку .
6. Для удаления фильтра выбрать команды *Записи/Удалить фильтр* или нажать кнопку .

### **Выполнение вычислений в запросах**

При создании запросов есть возможность выполнять вычисления над данными с просмотром результатов в результирующей таблице. Для этого в бланке запроса создается вычисляемое поле следующим образом: в пустую клетку строки **Поле** вводится имя поля с двоеточием, а затем формула расчета (операнды в формуле заключаются в квадратные скобки). Введенное имя поля будет заголовком расчетного столбца в результирующей таблице, а поля расчетного столбца будут содержать значения результата расчета по введенной формуле. Формулу расчета можно ввести с помощью Построителя выражений.


**Задание 7.** *Создать запрос ZAPR6, результирующая таблица которого должна содержать расписание в виде, удобном для восприятия студентами, и содержать сведения в следующем порядке: Наименование группы, Форма обучения, День недели, Дисциплина, Преподаватель, Должность преподавателя, Количество часов, Количество пар. Все объекты должны представляться наименованиями. Графа Количество пар рассчитывается по формуле:  $[KOLCH]/2$ . Сортировка записей должна быть следующей: 1-е поле **группа** возрастание; 2-е поле **день** возрастание.*



Выполняемые действия:

1. В окне БД набрать *Запросы / Создать / Конструктор*.
2. Добавить таблицы **RASP**, **SGR**, **SPR**, **SDY**, **SKUR**, **SDOL**.
3. Переместить в макет запроса поля **NAIMGR** и **FO** из таблицы **SGR**, **NDN** из **SDY**, **NAIMKUR** из **SKUR**, **FIO** из **SPR**, **NDOL** из **SDOL**, **KOLCH** из **RASP**.
4. В строку *Поле* первой пустой колонки ввести выражение **KOLPAR**:  $[KOLCH]/2$ , а в строке *Вывод на экран* установить флажок  [V], чтобы она стала видимой.

Чтобы ввести формулу с помощью Построителя выражений, следует выполнить действия:

- а) нажать кнопку  (*построить*) на панели инструментов;
  - б) в окне Построителя выражений в левом списке дважды щелкнуть по типу объектов *Таблицы* и выбрать таблицу **RASP**, щелкнув по ее папке;
  - в) в центральном списке дважды щелкнуть по полю **KOLCH**, чтобы его имя появилось в верхней части окна построителя;
  - г) ввести операцию / и 2. Слово «Выражение» необходимо удалить;
  - д) нажать ОК. Формула появится в бланке запроса.
5. Выполнить требуемую сортировку.
  6. Выполнить и сохранить запрос. Окончательный вид конструктора запроса изображен на рисунке 3.8.

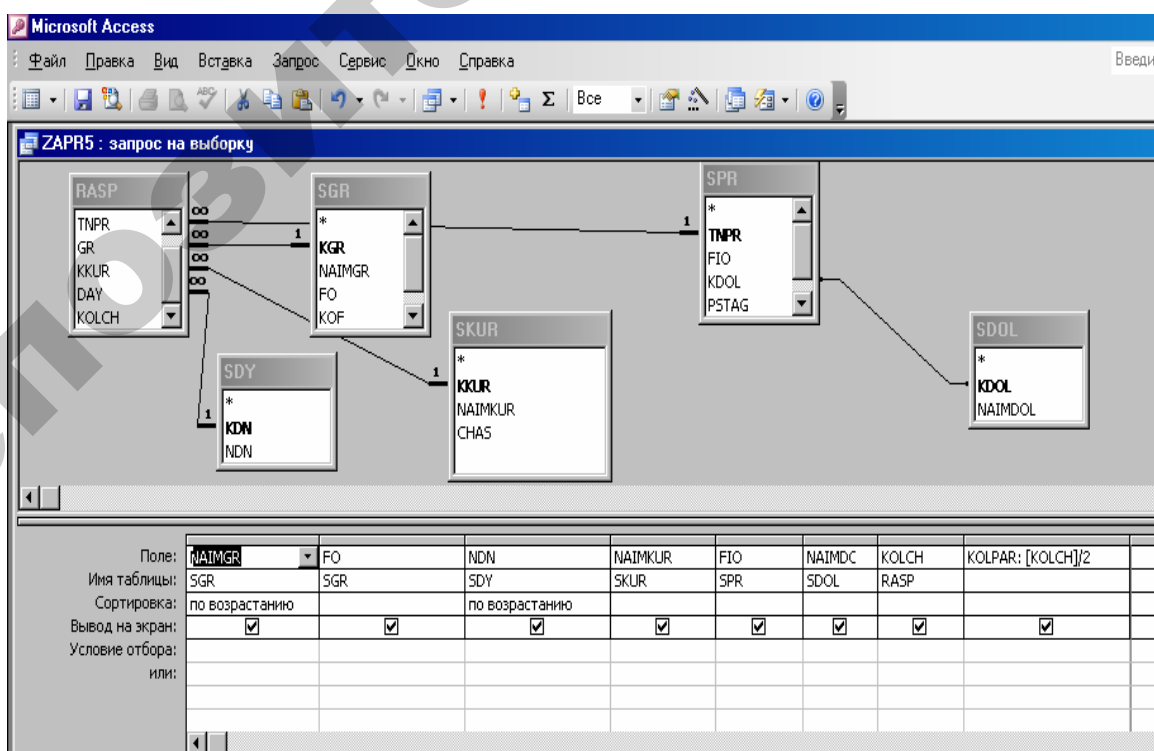


Рисунок 3.8 – Бланк запроса ZAPR6

## Создание простого запроса с помощью МАСТЕРА ЗАПРОСОВ

**Задание 8.** С помощью мастера запросов создать запрос **ZAPR7**, результирующая таблица которого должна содержать сведения в следующем порядке: День недели, Группа, Дисциплина, Количество часов, Преподаватель, Подразделение. Все объекты представляются своими наименованиями.

Выполняемые действия:

1. В окне БД набрать **Запросы / Создать / Простой запрос**.
2. Из нужных полей соответствующих таблиц скомпоновать макет запроса, перемещая поля кнопкой **>** между листами. Вид окна мастера изображен на рисунке 3.9. Нажать **Далее**.
3. В появившемся окне отметить пункт **Выбрать подробный отчет**, нажать **Далее**.
4. В следующем окне задать имя запроса **ZAPR7** и выбрать одно из предложенных действий:
  - *Открыть результат выполнения запроса;*
  - *Изменить структуру запроса.*

Нажать **Готово**.

Запрос создан. Его можно откорректировать в режиме КОНСТРУКТОРА ЗАПРОСОВ.

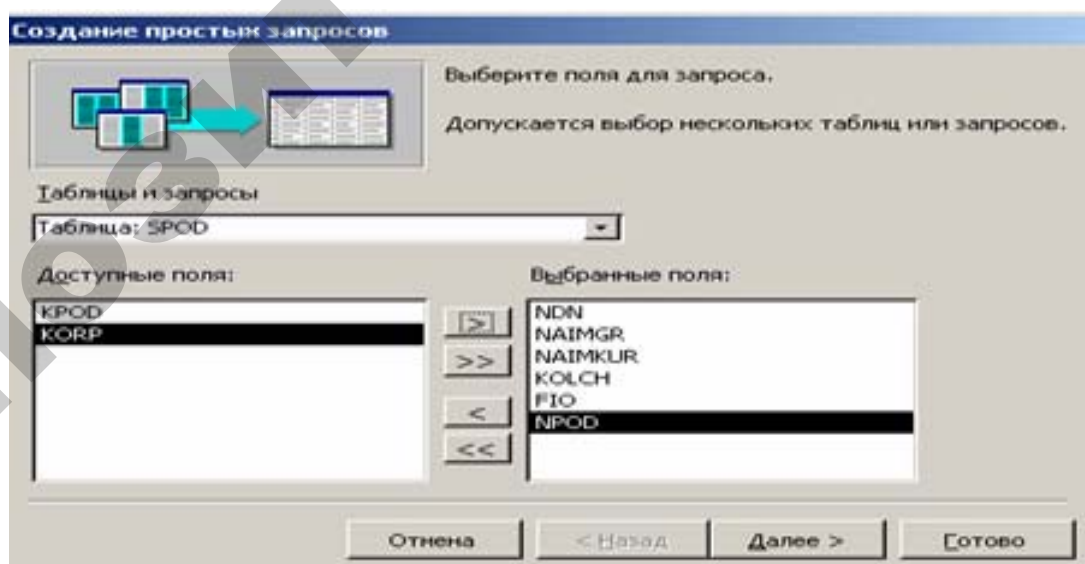


Рисунок 3.9 – Окно мастера ПРОСТОЙ ЗАПРОС



## Формирование запросов с группировкой

Часто требуется в таблице видеть не все записи, а только итоговые значения по группам записей. Расчет итогов для некоторых полей групп производится с помощью статистических функций.

**Задание 9.** Создать запрос **ZAPR8**, показывающий суммарное количество часов, проведенных каждым преподавателем за неделю.

Выполняемые действия:

1. Выбрать команды **Запросы / Создать / Конструктор** (или выбрать пункт **Создание запроса в режиме конструктора**).
2. Из окна ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить в макет запрос **ZAPR2**, предварительно выбрав вкладку **Запросы**. Закрыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.
3. В строку **Поле** макета переместить поля **FIO** и **KOLCH** из **ZAPR2**.
4. В строке **Вывод на экран** проверить наличие флажков (V) для полей.
5. Вызвать команду **Вид / Групповые операции**, в бланке запроса появится новая строка **Групповая операция**, в которой для всех полей указано **Группировка**.
6. В поле **KOLCH** вместо надписи **Группировка** задать нужную функцию (**Sum**), выбрав ее из списка функций, появившихся по щелчку в правой части поля.
7. Сохранить и просмотреть запрос.

### Создание перекрестного запроса

Перекрестные запросы позволяют отображать результаты статистических расчетов (такие как суммы, средние значения и др.) по данным из каждой строки или столбца. Далее выполняется группировка по двум наборам данных, один из которых определяет заголовки строк, а другой – заголовки столбцов.

**Задание 10.** Подсчитать количество часов, проводимых каждым преподавателем в каждой группе за неделю и вывести результаты в виде перекрестной таблицы.

Эту задачу решит перекрестный запрос. Для его создания необходимо выполнить следующие действия.

1. Создать простой запрос, включающий поля, требуемые для перекрестного запроса: **FIO** из таблицы **SPR**, **NAIMGR** из **SGR**, **KOLCH** из **RASP**. Назвать созданный запрос **ZAPRP**.

2. В окне БД выполнить команды: *Запросы / Создать / Перекрестный запрос*. В появившемся окне выполнить следующие действия:

- Выбрать *Показать запросы*, в качестве источника указать **ZAPRP**, нажать *Далее*.
- Для наименования строк выбрать поле **ГЮ**, нажать *Далее*.
- Для наименования столбцов выбрать **NAIMGR**, нажать *Далее*.
- Выбрать функцию, которую необходимо выполнить для ячеек на пересечении строк и столбцов. В нашем случае выбрать функцию **SUM** и указать поле **KOLCH**, нажать *Готово*.

3. Выполнить запрос и сохранить с именем **ZAPR9**.

### Задания для самостоятельного выполнения к практической работе № 3

Кроме описанных запросов создать следующие запросы в базе данных **RRASP**:

- 1) **ZAPR10** – перекрестный запрос, который позволит увидеть общее количество часов занятий каждой группы в каждый из дней недели.
- 2) **ZAPR11** – запрос, подобный **ZAPR5**, но добавить колонку **КОЛИЧЕСТВО ЧАСОВ ГРУППЫ**, рассчитываемую по формуле  $[KOLCH]*[KOF]$ , **KOF** – поле в таблице **SGR**.

Сортировка: 1-е поле **ГРУППА** убывание; 2-е поле **ДЕНЬ НЕДЕЛИ** возрастание по коду, т.е. Понедельник, Вторник т.д.

**Замечание 5.2** Для выполнения требуемой сортировки добавить поле **DAY** из таблицы **RASP** в запрос, выполнить по нему сортировку (2-е поле) и сделать это поле невидимым.

3) **ZAPR12** – запрос, содержащий всю информацию о преподавателях: Табельный номер, Фамилию, Наименование должности, Педагогический стаж, Наименование подразделения, Наименование корпуса расположения подразделения.

4) Закрывать базу данных **RRASP**.

### Создание запросов в базе данных **Складской Учет**

Открыть базу данных **Складской Учет** и создать в ней запросы.

1) Показать список накладных, согласно структуре таблицы **Накладные**, но Поставщиков представить наименованиями, а Материально-ответственных лиц – фамилиями. Выполнить сортировку по полю **НомНакл**.

2) Создать запрос, включив в результирующую таблицу следующие поля: Наименование поставщика (**НаимПост**), Номер на-

кладной (**НомНакл**), Код товара (**КодТов**), Наименование товара (**НаимТов**), Цену (**Цена**), **Количество**, **ДатаПост**, Фамилию сотрудника, принявшего товар (**ФИО**). Сортировка записей должна быть следующей: 1-е поле **НаимПост** возрастание; 2-е поле **НомНакл** возрастание.

3) На основе запроса пункта 2, создать запрос, выбирающий лишь информацию о товарах, поступивших после 15.11.2007, цена которых более 5 000 рублей, но менее или равна 15 000 рублей.

*Указание.* В строку УСЛОВИЕ ОТБОРА конструктора в колонку **ДатаПост** ввести выражение **>15.11.2007**, а в колонку **Цена** - **>5000 and <= 15000**.

4) Сконструировать запрос, выбирающий следующие сведения о товарах, поступивших по накладным: **НаимТов**, **КодТов**, **Цена**, **Количество**, **СтоимостьТов**, **НомНакл** (по записям таблицы **СтрокиНакл**). Сортировка: 1-е поле **НаимТов** возрастание; 2-е поле **НомНакл** возрастание.

*Указание.* **СтоимостьТов** рассчитать по формуле (**Цена\*Количество**).

5) Взяв за основу, запрос, созданный в предыдущем пункте, сконструировать запрос с группировкой, показывающий суммарное количество и стоимость каждого товара, полученного по накладным.

*Указание.* В результирующую таблицу включить поля **КодТов**, **НаимТов**, **Количество**, **Стоимость**.

6) Показать количество и стоимость товаров, поступивших к материально-ответственным лицам (сотрудникам), выбрать поля: **Таб-Номер**, **ФИО**, **НомНакл**, **НаимТов**, **Количество**, **Цена**, **Стоимость**. Выполнить сортировку по возрастанию 1-е поле – **ФИО**, второе поле – **НаимТов**.

7) Создать запрос, аналогичный запросу пункта 6, но выбирающий информацию лишь по отдельному сотруднику, а так же запрос, выбирающий информацию по сотруднику, фамилия которого содержит некоторую букву, вводимую с клавиатуры (запросы с параметрами).

8) На основе запроса пункта 6 создать запрос с группировкой, показывающий суммарную стоимость товаров, полученных каждым сотрудником.

9) С помощью перекрестного запроса показать общую стоимость каждого товара, поступившего к каждому МОЛу.

10) На основе запроса п. 2 создать запрос, показывающий общую стоимость товаров, полученных по каждой накладной (перекрестный запрос).

11) Создать перекрестный запрос, показывающий общую стоимость каждого товара, полученного от каждого поставщика (поставщик и товар представить наименованиями).

*Указание к пунктам 10, 11.* Предварительно создать запрос, подобный запросу пункта 2 с добавлением поля стоимость товара, рассчитанной по известной формуле.

### **Задание для контрольной работы**

Создать запросы согласно предложенному варианту.

## ПРАКТИЧЕСКАЯ РАБОТА № 4

### Создание запросов – действий

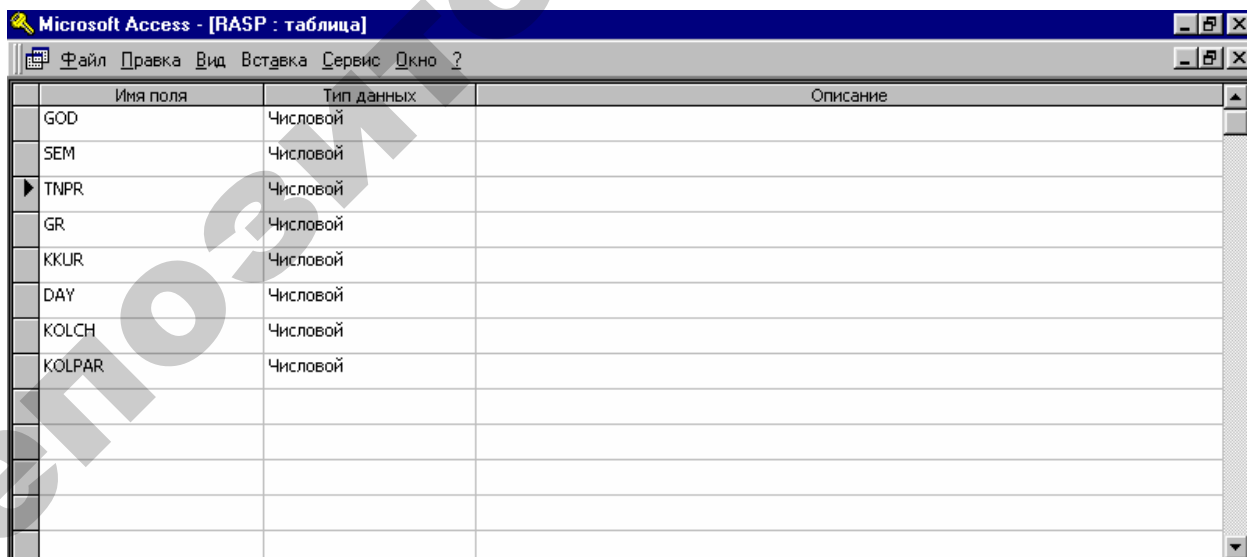
Для хранения в БД расписания за предыдущие семестры и создания расписания в следующем семестре на основе данных предыдущего семестра, выполним следующие действия.

1. В таблицу **RASP** добавим поля год (**GOD**) и семестр (**SEM**), значения которых будут идентифицировать год и семестр, к которому относится соответствующая запись таблицы. Добавим в таблицу также поле КОЛИЧЕСТВО ПАР (**KOLPAR**), вычисляемое по формуле  $[KOLCH]/2$ . Измененный макет таблицы **RASP** изображен на рисунке 3.10. Для имеющихся в таблице записей установим новым полям значения текущего года и семестра, например, 2008, 1 (с помощью запроса на обновление).

2. Для составления расписания на следующий семестр необходимо скопировать записи таблицы **RASP** за предыдущий семестр в нее же, но с новым значением полей **GOD** и **SEM**: 2009, 2 (с помощью запроса на добавление).

3. Изменить строки расписания на 2009 г., 2-й семестр в соответствии с требованиями.

4. Организовать просмотр расписания на каждый семестр отдельно.



Имя поля	Тип данных	Описание
GOD	Числовой	
SEM	Числовой	
TNPR	Числовой	
GR	Числовой	
KKUR	Числовой	
DAY	Числовой	
KOLCH	Числовой	
KOLPAR	Числовой	


Рисунок 3.10 – Измененный макет таблицы RASP

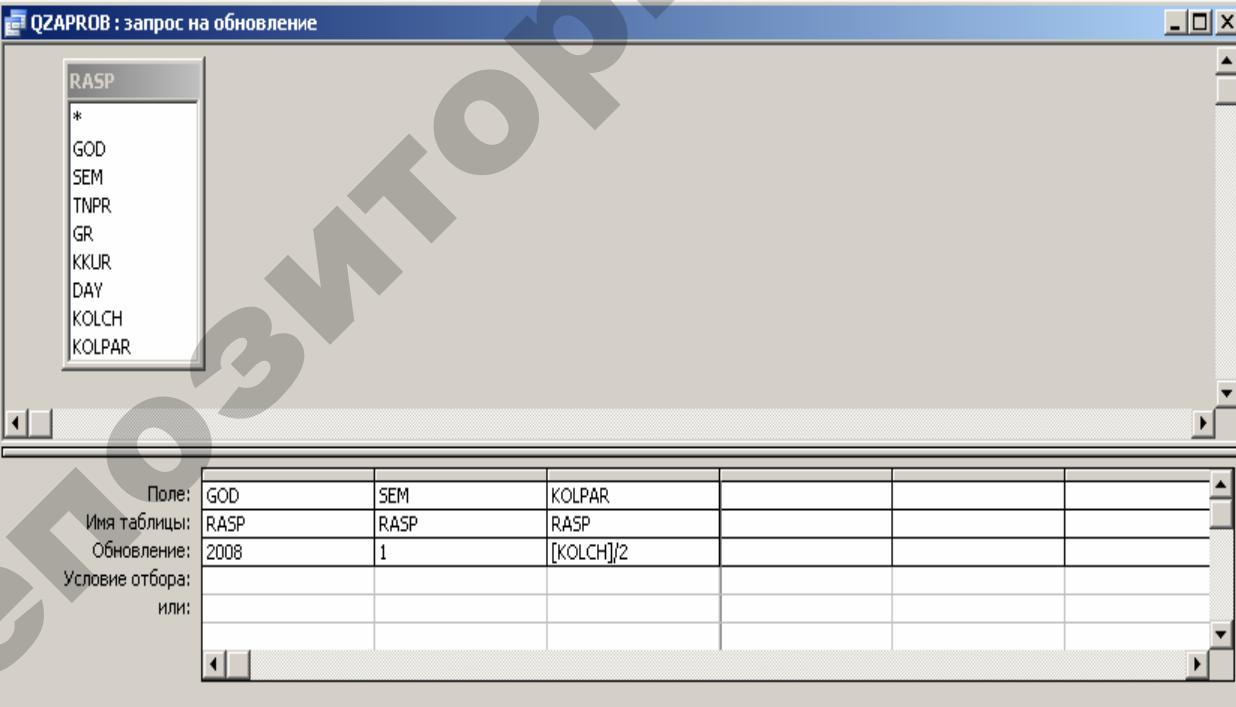
## Создание запроса на обновление

Запрос на обновление позволяет изменить значение выбранного поля во всех записях таблицы, удовлетворяющих указанным условиям. После добавления в макет таблицы **RASP** полей **GOD**, **SEM**, **KOLPAR** при просмотре таблицы видно, что эти поля пустые.

**Задание 1.** Создать запрос на обновление таблицы **RASP**, позволяющий придать новым полям нужные значения: **GOD** – 2008, **SEM** – 1, **KOLPAR** – рассчитать по формуле  $[KOLCH]/2$ .

Необходимые действия:

1. Выполнить команды **Запросы / Создать / Конструктор**.
2. В окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицу **RASP**.
3. В макет запроса переместить из таблицы поля **GOD**, **SEM**, **KOLPAR**.
4. Указать тип запроса. Для этого нажать кнопку  в режиме конструктора и выбрать тип **Обновление** или выполнить команды **Запрос / Обновление**. В появившуюся строку **Обновление** в обновляемые колонки ввести формулы обновления. Макет запроса изображен на рисунке 3.11.



Поле:	GOD	SEM	KOLPAR			
Имя таблицы:	RASP	RASP	RASP			
Обновление:	2008	1	$[KOLCH]/2$			
Условие отбора:						
или:						

Рисунок 3.11 – Бланк запроса ОБНОВЛЕНИЕ

5. Сохранить созданный запрос с именем **QZAPROB** и закрыть запрос.

6. Скопировать таблицу **RASP** с именем **RASPP** для того, чтобы в случае неудачного обновления таблицы **RASP** с запросом **QZAPROB** можно было восстановить ее первоначальное состояние.

**Замечание.** В случае использования запросов действий, изменяющих объекты базы данных, рекомендуется предварительно делать копии изменяемых объектов.

7. Обновить таблицу **RASP**. Для этого выполнить запрос **QZAPROB** (т.е. открыть его в окне БД). Следует обратить внимание на значок запросов типа **ОБНОВЛЕНИЕ**. На обе появляющиеся просьбы о подтверждении обновления ответить «ДА». Результатом работы запроса **QZAPROB** будет занесение значений в новые поля таблицы **RASP**. В этом следует убедиться, открыв таблицу **RASP**.

8. Добавить в таблицу **RASP** две любые записи за 2007 г., 1-й семестр.

### Создание запроса на добавление


Запрос на добавление позволяет дополнить существующую таблицу новыми записями.

**Задание 2.** Создать запрос **QZAPROB** на добавление в таблицу **RASP** записей этой же таблицы, касающихся 2008 г. 1-го семестра со следующими изменениями.

- **GOD** должен стать 2009;
- **SEM** – 2;
- В **KOLPAR** должно перейти значение из **KOLCH**;
- Значение поля **KOLCH** должно удвоиться.

Предполагается, что в следующем семестре количество часов увеличивается в два раза.

Необходимые действия:

1. Набрать **Запросы / Создать / Конструктор**.
2. В макет запроса перенести все поля из таблицы **RASP**.
3. В строку **Условие отбора** ввести: в графу **GOD** значение 2008, в графу **SEM** – 1.
4. Указать тип запроса, нажав кнопку и выбрав  . **Добавление** или **Запрос / Добавление**.
5. В макете запроса появилась строка **Добавление**. В ней по умолчанию содержатся имена полей, совпадающих с именами полей запроса. Имена полей в этой строке указывают, в какое поле таблицы будет добавляться поле запроса. Далее необходимо внести в макет запроса следующие изменения.

- Добавить колонку, вычисляемую по формуле  $[SEM]+1$  и в строке **Добавление** указать **SEM**; а в графе **SEM** (старой) запроса (где стоит условие отбора) в строке **Добавление** надпись **SEM** убрать.
  - Аналогично добавить колонку, вычисляемую по формуле  $[GOD]+1$ . В строке **Добавление** указать **GOD**, а в старой графе **GOD** в строке **Добавление** надпись **GOD** убрать.
  - Добавить колонку, вычисляемую по формуле  $[KOLCH]*2$  и в строке **Добавление** написать **KOLCH**.
  - В старой графе **KOLCH** в строке **Добавление** ввести **KOLPAR**, а в графе **KOLPAR** (старой) в строке **Добавление** надпись **KOLPAR** убрать. Окончательный вид макета запроса изображен на рисунке 3.12.
6. Сохранить запрос с именем **QZAPRDOB** и закрыть запрос.
7. В окне БД открыть запрос **QZAPRDOB**. Обратить внимание на значок запроса типа **ДОБАВЛЕНИЕ**. На два сообщения-предупреждения ответить «ДА». Выполнение запроса приведет к добавлению в таблицу **RASP** части записей из нее же с соответствующими изменениями.
8. Открыть таблицу **RASP**, убедиться, что добавление выполнилось правильно.

**Задание 3.** Во все созданные в практической работе № 3 запросы добавить параметры, позволяющие выбрать расписание за интересующий нас год и семестр.

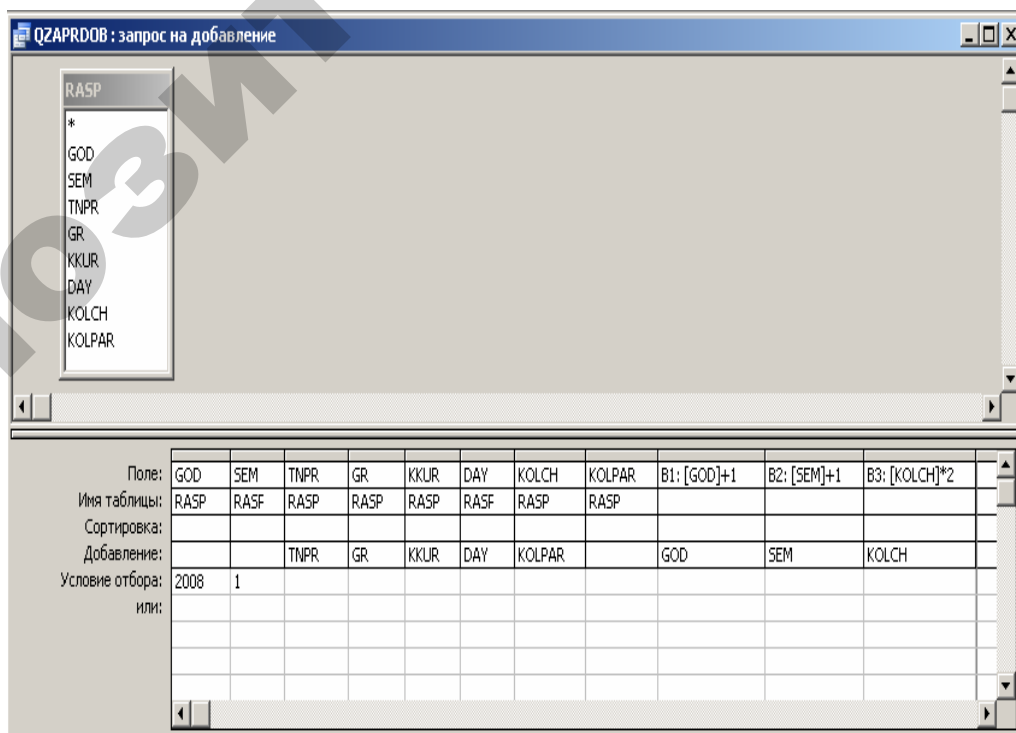


Рисунок 3.12 – Бланк запроса ДОБАВЛЕНИЕ




## Построение запроса на создание таблицы

Запрос на создание таблицы создает новую таблицу на основе информации, имеющейся в других таблицах БД.

**Задание 4.** С помощью запроса создать таблицу **RASP2** аналогичную результирующей таблице запроса **ZAPR2**, созданного в практической работе № 3.

Необходимые действия:


1. Набрать **Запросы / Создать / Конструктор**.
2. Сформировать макет запроса, аналогичный **ZAPR2**.
3. Указать тип запроса, выбрав **Запрос / Создание таблицы** или, нажав кнопку  и выбрав **Создание таблицы**.
4. В появившемся окне СОЗДАНИЕ ТАБЛИЦЫ указать имя создаваемой таблицы **RASP2**. В дальнейшем Access будет запрашивать подтверждение на удаление существующей таблицы **RASP2** перед каждым выполнением запроса на создание.
5. Сохранить запрос с именем **QZAPRSOZ**, обратить внимание на значок для запросов этого типа.
6. Открыть в окне БД запрос **QZAPRSOZ**. Результатом его выполнения будет создание таблицы **RASP2**.
7. В окне БД открыть таблицу **RASP2**, убедиться в правильности результата выполнения запроса **QZAPRSOZ**.

## Построение запроса на удаление

Запрос на удаление позволяет удалить из таблицы БД записи, удовлетворяющие заданному условию.

**Задание 5.** Удалить из таблицы **RASP2** записи, касающиеся преподавателя Петрова.

Необходимые действия:

1. В окне базы данных набрать **Запросы / Создать / Конструктор**.
2. Добавить таблицу **RASP2**.
3. В бланк запроса переместить поле **ФИО**.
4. В строке УСЛОВИЕ для поля **ФИО** ввести «Петров».
5. Указать тип запроса, выполнив **Запрос / Удаление** или нажав кнопку , выбрать тип УДАЛЕНИЕ.
6. Сохранить запрос с именем **QUDAL**, обратив внимание на значок запросов этого типа. Закрывать запрос.
7. В окне БД выполнить запрос **QUDAL**, подтвердить удаление ответом «ДА».

8. Для просмотра результата выполнения запроса **QUDAL** открыть таблицу **RASP2** и убедиться, что заданные записи удалены из запроса.

### **Задания для самостоятельного выполнения к практической работе № 4**

В базе данных **RRASP** выполнить следующие действия.

- 1) Добавить в таблицу **RASP2** поле Учебное заведение (**UCHZ**). Занести туда значение БГАТУ с помощью запроса на обновление.
- 2) Удалить из таблицы **RASP2** записи, касающиеся Дня недели «Пятница».
- 3) Закрывать базу данных **RRASP**.

### **Создание запросов-действий в базе данных Складской Учет**

Открыть базу данных **Складской Учет**. Создать в ней запросы-действия:

- 1) Добавить в таблицу **СпрМол** поле **Организация** – наименование организации, для которой создается база данных **Складской Учет**.
- 2) Занести в поле **Организация** наименование организации с помощью запроса на обновление.
- 3) Пусть известно, что поступила накладная с номером 6, идентичная по структуре накладной номер 1. Скопировать в таблице **СтрокиНакл** строки накладной номер 1 в эту же таблицу с новым значением, равным 6, поля **НомНакл**.

**Указание.** Создать запрос на добавление.

- 4) С помощью запроса на создание таблицы создать таблицу **Информация**, содержащую следующие поля: **НаимПост**, **НомНакл**, **ДатаПост**, **ФИО**, **Должность**, **Оклад**.
- 5) С помощью запроса на удаление удалить из таблицы **Информация** строки, с наименованием поставщика, содержащим букву «л».

### **Задание для контрольной работы**

Создать запросы-действия согласно предложенному варианту.

## ВАРИАНТЫ ЗАДАНИЙ ДЛЯ КОНТРОЛЬНОЙ РАБОТЫ

### Контроль успеваемости в группах

Исходные таблицы:

1) Сведения об успеваемости:

ГОД	СЕМЕСТР	ТАБ. № СТУДЕНТА	НОМЕР ДИСЦИПЛИНЫ	ОЦЕНКА
1	2	3	4	5

2) Справочник студентов:

ТАБ.№ СТУДЕНТА	ФИО	АДРЕС	ГОД ПОСТУПЛЕ- НИЯ В ВУЗ	ГОД ОКОНЧАНИЯ ВУЗА	НОМЕР ГРУППЫ	ПОЛ
1	2	3	4	5	6	7

3) Справочник дисциплин:

КОД ДИСЦИПЛИНЫ	НАИМЕНОВАНИЕ	КОЛИЧЕСТВО ЧАСОВ
1	2	3

4) Справочник групп:

КОД ГРУППЫ	НАИМЕНОВАНИЕ	КОД ФОРМЫ ОБУЧЕНИЯ
1	2	

5) Справочник форм обучения:

КОД ФОРМЫ ОБУЧЕНИЯ	НАИМЕНОВАНИЕ
1	2

Приложение должно содержать следующие функции, оформленные как пункты меню:

1) Просмотр и печать результатов сдачи сессии студентами в ВУЗе:

ГРУППА (НАИМЕНОВАНИЕ)	ФИО СТУДЕНТА	НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ	ОЦЕНКА
1	2	3	4

2) При печати получить итоги по каждому студенту, по всей группе и общие итоги по ведомости в графе оценка (общий балл).

Просмотр сведений о студентах отдельной заданной формы обучения в виде:

ФИО СТУДЕНТА	№ ГРУППЫ	НАИМЕНОВАНИЕ ФОРМЫ ОБУЧЕНИЯ	НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ	ОЦЕНКА
1	2	3	4	5

3) Для каждого студента получить на экране список всех его оценок по дисциплинам за последнюю сессию с подсчетом среднего балла. Представить в виде формы с подчиненной формой.

4) С помощью программного модуля VBA, сформировать таблицу:

ГОД	СЕМЕСТР	ТАБ. № СТУДЕНТА	КОЛИЧЕСТВО ДИСЦИПЛИН	ОБЩИЙ БАЛЛ	СРЕДНИЙ БАЛЛ
1	2	3	4	5	6

5) Функции 1, 3, 4, выполнить для текущей сессии и следующей. Скопировать данные таблицы №1 в нее же, придать новые значения полям, «год» и «семестр», в графу «оценка» при копировании занести значение 0.

## 2) Учет основных средств на предприятии

Исходные таблицы:

### 1) Картотека основных средств (ОС – основное средство):

ГОД	МЕС	ИНВ.№ ОС	НАИМЕ- НОВ ОС	ТАБ.№ МОЛА	№ ГР- УПШЫ	ДАТА ВВОДА В ЭКСПЛ.	БАЛАНС. СТОИМ	КОД НОРМЫ АМОРТ-	МЕС ИЗНОС	СУММА ИЗНОСА	ДАТА ВЫБЫ- ТИЯ	ОСТА- ТОЧ. СТОИ- МОСТЬ.
1	2	3	4	5	6	7	8	9	10	11	12	13

### 2) Справочник материально ответственных лиц (МОЛов):

ТАБЕЛЬНЫЙ № МОЛА	ФАМИЛИЯ
------------------	---------

### 3) Справочник групп основных средств

№ ГРУППЫ	НАИМЕНОВАНИЕ
1	2

### 4) Справочник норм амортизации

КОД НОРМЫ АМОРТИЗАЦИИ	НОРМА АМОРТИЗАЦИИ	НАИМЕНОВАНИЕ
1	2	3

Приложение должно выполнять следующие функции, оформленные в виде пунктов меню. Действия выполнять за текущий месяц и за следующий.

#### 1) Расчет амортизации основных средств (обновление таблицы № 1):

**Гр.10 = Гр. 8/(Гр.2(табл.№4)\*12); Гр.11 = Гр.11+Гр.10; Гр.13 =Гр.8-Гр. 11.**

Рассчитанные значения граф должны перейти в информацию по следующему месяцу в таблице №1.

#### 2) Просмотр и печать сведений об основных средствах в след. виде:

ГРУППА ОС (НАИМЕНОВАНИЕ)	ИНВ. № ОСН.СР.	НАИМЕ- НОВАНИЕ ОСН.СР.	ФИО МО- ЛА	БАЛАНСО- ВАЯ СТО- ИМОСТЬ	МЕСЯЧ- НЫЙ ИЗНОС	СУММА ИЗНОСА	ОСТАТОЧНАЯ СТОИМОСТЬ
1	2	3	4	5	6	7	8

При печатании получить итоги по группам ОС и общие итоги по ведомости в графах 5, 7, 8.

#### 3) Просмотр сведений об основных средствах в порядке возрастания табельного № МОЛа с возможностью поиска по табельному №.

#### 4) Просмотр полностью изношенных основных средств, остаточная стоимость которых равна 0.

#### 5) Для каждого МОЛа получить на экране список основных средств, числящихся за ним в виде:

ТАБ. № МОЛА	ИНВ.№ ОСН.СРЕД.	НАИМЕНОВАНИЕ	БАЛАНСОВАЯ СТОИМОСТЬ	СУММА ИЗНОСА	ОСТАТОЧНАЯ СТОИМОСТЬ
1	2	3	4	5	6

Представить в виде формы с подчиненной формой. В подчиненной форме получить итоги по графам 4, 5, 6.

б) Создав программный модуль VBA, сформировать таблицу, содержащую сведения о группах основных средств

ГОД	МЕ- СЯЦ	КОД ГРУП- ПЫ	КОЛИЧЕСТВО ВХОДЯЩИХ ОСНОВНЫХ СРЕДСТВ	ОБЩАЯ БАЛАНСОВАЯ СТОИМОСТЬ	ОБЩАЯ СУММА ИЗНОСА	ОБЩАЯ ОСТАТОЧНАЯ СТОИМОСТЬ
-----	------------	--------------------	--	----------------------------------	--------------------------	----------------------------------

**Замечание.** Списанные основные средства в расчет не включать.

7) Все указанные функции выполнить в текущем месяце и в следующем, скопировав данные таблицы 1 в нее же с новым значением поля «месяц» и, если нужно, поля «год».

### 3) Учет спецодежды на предприятии

Исходные таблицы:

1) Сведения о наличии спецодежды у сотрудников:

ГОД	МЕСЯЦ	ТАБЕЛЬНЫЙ № СОТРУДНИКА	ИНВЕНТАРНЫЙ № СПЕЦОДЕЖДЫ	ДАТА ПОСТУПЛЕНИЯ	КОЛИ- ЧЕСТВО
1	2	3	4	5	6

2) Справочник сотрудников:

ТАБЕЛЬНЫЙ № СОТРУДНИКА	ФИО
1	2

3) Справочник спецодежды:

ИНВЕНТАРНЫЙ № СПЕЦОДЕЖДЫ	НАИМЕНОВАНИЕ	ЦЕНА
1	2	3

Приложение должно выполнять функции, представленные в виде меню.

1) Просмотр и печать сведений о наличии спецодежды в виде:

ФИО СОТРУДНИКА	ИНВЕНТАРНЫЙ № СПЕЦОДЕЖДЫ	НАИМЕНОВАНИЕ СПЕЦОДЕЖДЫ	ДАТА ПОС- ТУПЛЕНИЯ	КОЛИЧЕ СТВО	ЦЕ НА	СУМ -МА
1	2	3	4	5	6	7

Гр. 6 выбирается из справочника спецодежды. Гр. 7 = Гр. 5 \* гр. 6.

При печати получить итоги по каждому сотруднику и по всей ведомости в графах 5, 7.

2) Просмотр сведений о наличии спецодежды в той же форме, но в порядке возрастания инвентарного № спецодежды.

3) Выбрать и просмотреть только ту спецодежду, стоимость которой >130 000 рублей в порядке возрастания поля «цена».

4) Для каждого наименования спецодежды просмотреть на экране список сотрудников, за которыми она числится с итогом по графе «сумма» в виде:

ИНВЕНТАРНЫЙ № СПЕЦОДЕЖДЫ	ФИО СОТРУДНИКА	КОЛИЧЕСТВО	СУММА
1	2	3	4

Представить в виде формы с подчиненной формой.

5) Создав программный модуль VBA, сформировать таблицу, содержащую сведения об общем количестве и стоимости спецодежды по каждому МОЛу:

ГОД	МЕСЯЦ	ТАБЕЛЬНЫЙ № МОЛА	КОЛИЧЕСТВО НАИМЕНОВАНИЙ СПЕЦОДЕЖДЫ	ОБЩЕЕ КОЛИЧЕСТВО ЕДИНИЦ	ОБЩАЯ СУММА
1	2	3	4	5	6

б) Все указанные действия должны выполняться в текущем месяце и в следующем. При копировании записей таблицы №1 в нее же с новым значением поля «месяц» и, если нужно, поля «год» содержимое всех граф оставить прежним.

#### 4) Поступление материалов на предприятие

Исходные таблицы:

1) Поступление материалов по накладным к материально-ответственным лицам (МОЛам):

ГОД	МЕ- СЯЦ	ТАБЕЛЬНЫЙ № МОЛА	№ ДОКУ- МЕНТА	ДАТА ПОЛУ- ЧЕНИЯ	КОД ПОСТАВЩИКА	НОМЕНКЛА- ТУРНЫЙ № МАТЕРИАЛА	КОЛИ- ЧЕСТВО
1	2	3	4	5	6	7	8

2) Справочник материалов:

НОМЕНКЛАТУРНЫЙ № МАТЕРИАЛА	НАИМЕНОВАНИЕ МАТЕРИАЛА	ЕДИНИЦА ИЗМЕРЕНИЯ	ЦЕНА
-------------------------------	---------------------------	----------------------	------

3) Справочник МОЛов:

ТАБЕЛЬНЫЙ № МОЛА	ФИО
------------------	-----

4) Справочник организаций (поставщиков):

№ ОРГАНИЗАЦИИ	НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ
---------------	--------------------------

Приложение должно выполнять следующие функции, оформленные как пункты меню:

1) Просмотр и печать сведений о поступлении материалов в виде:

ФАМИ- ЛИЯ МОЛА	№ ДОКУ- МЕНТА	ДАТА ПОЛУ- ЧЕНИЯ	НАИМЕНОВАНИЕ ПОСТАВЩИКА	НАИМЕНО- ВАНИЕ МАТЕРИАЛА	ЦЕ НА	КОЛИЧЕ СТВО	СУММА
1	2	3	4	5	6	7	8

**Гр.8=Гр.6\*Гр.7**

При печати получить итоги по МОЛам и общий итог по ведомости в графе 8.

2) Просмотр на экране сведений о поступлении материалов в виде:

ТАБЕЛЬНЫЙ № МОЛА	НАИМЕНОВАНИЕ ПОСТАВЩИКА	НАИМЕНОВАНИЕ МАТЕРИАЛА	ЦЕ- НА	КОЛИЧЕСТВО	СУМ- МА
---------------------	----------------------------	---------------------------	-----------	------------	------------

В алфавитном порядке МОЛов.

3) Просмотр той же таблицы, но в алфавитном порядке наименований материалов с возможностью поиска по наименованию материала.

4) Выбрать только материалы, цена которых <250 000 рублей и вывести по возрастанию цены.

5) Для каждого материала просмотреть на экране список поставщиков, от которых данный материал поступил с итогом по графе «сумма» в виде:

ИНВЕНТАРНЫЙ № МАТЕРИАЛА	НАИМЕНОВАНИЕ ПОСТАВЩИКА	ЦЕНА	КОЛИЧЕСТВО	СУММА
----------------------------	----------------------------	------	------------	-------

Использовать форму с подчиненной формой.

6) Создав программный модуль VBA, сформировать таблицу, содержащую сведения по каждому поступившему материалу:

Г О Д	МЕСЯЦ	ИНВЕНТАРНЫЙ № МАТЕРИАЛА	КОЛИЧЕСТВО МОЛОВ, К КОТОРЫМ МАТЕРИАЛ ПОСТУПИЛ	ЦЕНА	ОБЩАЯ СУММА ПОСТУПЛЕНИЯ
1	2	3	4	5	6

7) Все указанные действия должны выполняться в текущем месяце и в следующем. Скопировать данные таблицы №1 за текущей месяц в нее же с новым значением месяца и, если нужно, года. В графу 8 занести 0.

## 5) Приход материалов за месяц и за период с начала года

Исходные таблицы:

1) Таблица прихода материалов:

ГОД	МЕ- СЯЦ	ТАБЕЛЬНЫЙ № МОЛА	НОМЕНКЛАТУРНЫЙ № МАТЕРИАЛА	ПРИХОД (КОЛИЧЕСТВО)	
				ЗА МЕСЯЦ	ЗА ПЕРИОД С НАЧАЛА ГОДА
1	2	3	4	5	6

2) Справочник материалов:

НОМЕНКЛАТУРНЫЙ № МАТЕРИАЛА	НАИМЕНОВАНИЕ	ЕДИНИЦА ИЗМЕРЕНИЯ	ЦЕНА
1	2	3	4

3) Справочник материально-ответственных лиц (МОЛов):

ТАБЕЛЬНЫЙ № МОЛА	ФАМИЛИЯ
1	2

Приложение должно выполнять следующие функции, как пункты меню.

1) Расчет графы 6 таблицы № 1 по формуле: Гр. 6=Гр. 6 + Гр.5. (С помощью запроса на обновление таблицы).

2) Просмотр и печать сведений о приходе материалов за месяц и за период с начала года в виде:

ФАМИ- ЛИЯ МОЛА	НАИМЕНО- ВАНИЕ МАТЕРИАЛА	ЕДИНИЦА ИЗМЕРЕ- НИЯ	ЦЕНА	ПРИХОД ЗА МЕСЯЦ		ПРИХОД ЗА ПЕРИОД С НАЧАЛА ГОДА	
				КОЛ-ВО	СУММА	КОЛ-ВО	СУММА
1	2	3	4	5	6	7	8

$$\text{Гр. 6} = \text{Гр. 4} * \text{Гр. 5}; \quad \text{Гр. 8} = \text{Гр. 4} * \text{Гр. 7}.$$

Получить итоги по МОЛам и общие итоги по ведомости в графах 6 и 8.

3) Просмотр поступлений материалов к МОЛам в виде:

ФАМИЛИЯ МОЛА	НАИМЕНОВАНИЕ МАТЕРИАЛА	ЦЕНА	ПРИХОД (СУММА)	
			ЗА МЕСЯЦ	С НАЧ. ГОДА
1	2	3	4	5

В алфавитном порядке фамилий МОЛов.

4) Просмотр этой же таблицы, но в алфавитном порядке наименований материалов с возможностью поиска по наименованию материала.

5) Просмотр прихода только материалов, цена которых менее 60 000 рублей в порядке возрастания цены.

6) Для каждого материала просмотреть приход за период с начала года по МОЛам с итогом по графе «сумма» в виде:

НОМЕНКЛАТУРНЫЙ № МАТЕРИАЛА	ФИО МОЛА	ЦЕНА	КОЛИЧЕСТВО	СУММА
1	2	3	4	5

Использовать форму с подчиненной формой.

7) Создав программный модуль VBA, сформировать таблицу, содержащую сведения об общем приходе материалов к молам

Г О Д	МЕ- СЯЦ	ТАБЕЛЬ НЫЙ № МОЛА	ОБЩИЙ ПРИХОД ЗА МЕС		ОБЩИЙ ПРИХОД ЗА ПЕРИОД С НАЧ.ГОДА	
			КОЛ-ВО НАИМЕНОВАНИЙ	СУМ МА	КОЛ-ВО НАИМЕНОВАНИЙ	СУММА
1	2	3	4	5	6	7

8) Все указанные действия должны выполняться в текущем месяце и в следующем. Для этого скопировать данные таблицы 1 в нее же с новым значением поля «месяц» и, если нужно, поля «год», оставив содержимое графы 6 прежним, а в графу 5 занести 0. Дать возможность выполнять все перечисленные действия в каждом месяце отдельно.

## б) Расчет заработной платы сотрудникам предприятия

Исходные таблицы:

1) Табель сотрудников:

ГОД	МЕ- СЯЦ	ТАБЕЛ. № СОТРУДНИКА	КОД ПОДРАЗДЕЛЕНИЯ	КОЛИЧЕСТВО ОТРАБОТАННЫХ ДНЕЙ	ДНЕВНОЙ ТАРИФ
1	2	3	4	5	6

2) Справочник сотрудников:

ТАБЕЛЬНЫЙ № СОТРУДНИКА	ФИО	КОД ПРОФЕССИИ	РАЗРЯД
1	2	3	4

3) Справочник профессий:

КОД ПРОФЕССИИ	НАИМЕНОВАНИЕ
1	2

4) Справочник подразделений:

КОД ПОДРАЗДЕЛЕНИЯ	НАИМЕНОВАНИЕ
1	2

Приложение должно выполнять следующие функции, оформленные, как пункты меню:

1) Просмотр и печать сведений о начисленной зарплате в виде:

НАИМЕНОВАНИЕ ПОДРАЗДЕЛЕНИЯ	ТАБЕЛ. № СОТРУД.	ФИО	НАИМЕНОВАНИЕ ПРОФЕССИИ	КОЛ-ВО ОТРАБОТ. ДНЕЙ	ДНЕВНОЙ ТАРИФ	СУММА
1	2	3	4	5	6	7

В порядке возрастания кода подразделения, внутри одного подразделения по возрастанию табельного № сотрудника. **Гр. 7 = Гр. 5 \* Гр. 6.**

При печати получить итоги по каждому подразделению и общие итоги по всей ведомости в графах « кол-во отработанных дней » и « сумма ».

2) Просмотр этих же сведений в порядке возрастания табельного № сотрудника с возможностью поиска по табельному № сотрудника .

3) Просмотр данных только по тем сотрудникам, сумма заработка у которых превысила 1 200 000 рублей .

4) По каждой профессии из справочника профессий просмотр списка сотрудников данной профессии в виде:

КОД ПРОФЕССИИ	ФИО СОТРУДНИКА	НАИМЕНОВАНИЕ ПОДРАЗДЕЛЕНИЯ	КОЛ-ВО ОТРАБОТАННЫХ ДНЕЙ	СУММА
1	2	3	4	5

С получением итога по графе 5. (Использовать форму с подчиненной формой).

4) Создав программный модуль VBA, сформировать таблицу, содержащую сведения по профессиям

ГОД	МЕСЯЦ	КОД ПРОФЕССИИ	КОЛИЧЕСТВО СОТРУДНИКОВ	ОБЩЕЕ КОЛИЧЕСТВО ОТРАБОТАННЫХ ДНЕЙ	ОБЩАЯ НАЧИСЛЕННАЯ СУММА
1	2	3	4	5	6

5) Все указанные действия должны выполняться в текущем месяце и в следующем. При копировании записей таблицы №1 в нее же с новым значением поля « месяц » (и, если нужно, поля « год ») занести значение 0 в графу « кол-во отработанных дней ».



## 7) Учет банковских документов на предприятии по счетам

Исходные таблицы:

1) Платежные документы:

№ ДОКУМЕНТА	ПРИЗНАК ДОКУМЕНТА 1-ПОРУЧЕНИЕ 2-ТРЕБОВАНИЕ	КОД ОРГАНИЗАЦИИ	КОРРЕСПОНДИРУЮЩИЙ СЧЕТ	СУММА
1	2	3	4	5

2) Справочник организаций:

КОД ОРГАНИЗАЦИИ	НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ
1	2

3) Справочник счетов:

№ СЧЕТА	НАИМЕНОВАНИЕ СЧЕТА
1	2

Приложение должно выполнять следующие функции, оформленные в виде пунктов меню:

- 1) Просмотр банковских документов (сначала поручения, затем требования) в порядке возрастания № документа.
- 2) Просмотр банковских документов по счетам. По каждому счету вначале платежные поручения, затем платежные требования.
- 3) Просмотр только платежных поручений или только платежных требований.
- 4) Печать списка банковских документов в виде:

КОРРЕСПОНДИРУЮЩИЙ СЧЕТ	ПРИЗНАК ДОКУМЕНТА 1-ПЛ.ПОРУЧ. 2-ПЛ.ТРЕБОВ.	№ ДОКУМЕНТА	НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ.	СУММА
1	2	3	4	5

Сортировка по графе «корреспондирующий счет». По каждому счету вначале должны располагаться платежные поручения затем платежные требования. Получить итоги в графе «сумма» по каждому счету, отдельно по платежным поручениям и по платежным требованиям.

5) По каждому счету просмотреть на экране все имеющиеся платежные поручения с получением итога по графе "сумма» в виде

№ СЧЕТА	НАИМЕНОВАНИЕ ОРГАНИЗАЦИИ	№ ДОКУМЕНТА	СУММА
1	2	3	4

Использовать форму с подчиненной формой.

6) Создав программный модуль VBA, сформировать таблицу, содержащую сведения об общем количестве и сумме платежных поручений и платежных требований по каждой организации.

ГОД	МЕСЯЦ	КОД ОРГАНИЗАЦИИ	ПЛАТЕЖНЫЕ ПОРУЧЕНИЯ		ПЛАТЕЖНЫЕ ТРЕБОВАНИЯ	
			КОЛИЧЕСТВО	СУММА	КОЛИЧЕСТВО	СУММА
1	2	3	4	5	6	7

7) Все указанные действия выполнить в текущем месяце и в следующем, скопировав данные таблицы 1 в нее же, придав новое значение полю «месяц» (и, если нужно, полю «год»). В колонку «сумма» занести 0.

## 8) Переоценка основных средств на предприятии

Исходные таблицы:

### 1) Сведения о переоценке

ГОД	МЕСЯЦ	ТАБЕЛЬНЫЙ № МОЛА	№ ГРУППЫ	ИНВЕНТАРНЫЙ № ОСНОВН. СРЕДСТВ	НАИМЕНОВАНИЕ ОСНОВНЫХ СРЕДСТВ	СУММА ИЗНОСА СТАРАЯ	БАЛАНСОВАЯ СТОИМОСТЬ СТАРАЯ	КОЭФФИЦИЕНТ ПЕРЕОЦЕНКИ
1	2	3	4	5	6	7	8	9

### 2) Справочник МОЛов

ТАБЕЛЬНЫЙ № МОЛА	ФИО
1	2

### 3) Справочник групп

КОД ГРУППЫ	НАИМЕНОВАНИЕ
1	2

Приложение должно выполнять следующие функции, оформленные в виде пунктов меню.

### 1) Просмотр и печать сведений о переоценке в виде:

НАИМЕНОВАНИЕ ГРУППЫ	ИНВЕНТАРНЫЙ № ОСНОВН. СРЕДСТВА	НАИМЕНОВАНИЕ ОСНОВН. СРЕДСТВА	ФИО МОЛ	БАЛ. СТОИМОСТЬ СТАР.	СУММА ИЗНОСА СТАРАЯ	КОЭФ. ПЕРЕОЦЕНКИ	БАЛ. СТОИМОСТЬ НОВАЯ	СУММА ИЗНОСА НОВАЯ
1	2	3	4	5	6	7	8	9

**Гр. 8 = Гр.5\*Гр.7 ; Гр. 9 = Гр.6\*Гр.7.**

При печати получить итоги по группам основных средств и общие итоги по ведомости в колонках 5, 6, 8, 9.

2) Просмотр сведений о переоценке в алфавитном порядке МОЛов.

3) Просмотр сведений о переоценке в порядке возрастания инвентарного № основных средств, выбрать только те строки, коэффициент переоценки которых >1.2.

4) Для каждой группы основных средств получить на экране список основных средств, ее составляющих с итогом по графам 3 и 4 в виде

КОД ГРУППЫ	НАИМЕНОВАНИЕ ОСНОВНЫХ СРЕДСТВ.	БАЛАНСОВАЯ СТОИМОСТЬ НОВАЯ	СУММА ИЗНОСА НОВАЯ
1	2	3	4

Представить в виде формы с подчиненной формой.

5) Создав программный модуль VBA, сформировать таблицу, содержащую сведения об общей стоимости и количестве основных средств, закрепленных за каждым сотрудником.

ТАБЕЛЬНЫЙ № МОЛА	ОБЩЕЕ КОЛИЧЕСТВО ОСНОВНЫХ СРЕДСТВ.	СУММАРНАЯ БАЛАНС. СТОИМОСТЬ НОВАЯ	ОБЩАЯ СУММА ИЗНОСА НОВАЯ
1	2	3	4

6) Все указанные действия должны выполняться в текущем году и в следующем. При копировании записей таблицы №1 с новым значением поля «год» в нее же в графу 8 занести новое значения балансовой стоимости, рассчитанное по формуле: **старое значение балансовой стоимости \* коэффициент переоценки**, в графу 7 – новое значение суммы износа, рассчитанное аналогично; в графу 9 занести 1.

**Замечание.** Переоценка основных средств выполняется в январе каждого года.

## 9) Учет поступления инструментов в цех

Исходные таблицы:

1) Сведения о количестве инструментов, поступивших к материально-ответственным лицам(МОЛам):

ГОД	МЕСЯЦ	ТАБ. № МОЛА	ИНВ. № ИНСТРУМЕНТА	ДАТА ПОСТУПЛЕНИЯ	КОЛИЧЕСТВО
1	2	3	4	5	6

2) Справочник МОЛов:

ТАБЕЛЬНЫЙ № МОЛА	ФАМИЛИЯ
1	2

3) Справочник инструментов:

ИНВЕНТАРНЫЙ № ИНСТРУМЕНТА	НАИМЕНОВАНИЕ	ЦЕНА
---------------------------	--------------	------

Приложение должно выполнять следующие функции, оформленные в виде пунктов меню:

1) Просмотр и печать сведений о поступлении инструментов к МОЛам в следующем виде:

ТАБ. № МОЛА	ФАМИЛИЯ МОЛА	НАИМЕНОВАНИЕ ИНСТРУМЕНТА	ДАТА ПОСТУП- ЛЕНИЯ	КОЛИ- ЧЕСТВО	ЦЕНА	СТОИМОСТЬ
1	2	3	4	5	6	7

Гр. 6 выбирается из справочника инструментов; **Гр. 7 = Гр. 5 \* Гр. 6.**

При печати получить итоги по каждому МОЛу и общий итог по ведомости в графах «количество» и «сумма».

2) Просмотр сведений о поступлении инструментов в той же форме, что и в пункте 1, но сортировка в алфавитном порядке наименований инструментов.

3) Выбрать сведения об инструментах, цена которых < 25000 рублей. Вывести в той же форме в порядке возрастания цены.

4) Для каждого инструмента получить на экране список МОЛов, к которым они поступили, с итогом в графе «стоимость». Список должен иметь вид

ИНВЕНТАРНЫЙ № ИНСТРУМЕНТА	ФАМИЛИЯ МОЛА	КОЛИЧЕСТВО	СТОИМОСТЬ
1	2	3	4

Задание выполнить в виде формы с подчиненной формой.

5) Создав программный модуль VBA, сформировать таблицу, содержащую следующие сведения о наличии инструментов у каждого МОЛа: количество наименований инструментов, общее количество единиц и общая стоимость.

ГОД	МЕСЯЦ	ТАБ. № МОЛА	КОЛИЧЕСТВО НАИМЕНОВАНИЙ	ОБЩЕЕ КОЛИЧЕСТВО ЕДИНИЦ	ОБЩАЯ СТОИМОСТЬ
1	2	3	4		5

6) Все указанные действия должны выполняться в текущем месяце и в следующем. При копировании записей таблицы № 1 в нее же с новым значением поля «месяц» и, если нужно, поля «год» в графу «количество» занести 0.

## 10) Учет наличия товаров на складах базы

Исходные таблицы:

1) Сведения о движении товаров на складах базы:

ГОД	МЕ- СЯЦ	НОМЕР СКЛАДА	КОД ТОВАРА	ОСТАТОК НА НАЧАЛО МЕСЯЦА, КОЛ-ВО	ПОСТУПЛЕ- НИЕ, КОЛ-ВО	ОТГРУЗКА, КОЛ-ВО
1	2	3	4	5	6	7

Замечание: колонки 6 и 7 – расчетные.

2) Справочник складов:

№ СКЛАДА	НАЗВАНИЕ СКЛАДА	ФИО ЗАВЕДУЮЩЕГО
1	2	3

3) Справочник товаров:

ИНВЕНТАРНЫЙ № ТОВАРА	НАИМЕНОВАНИЕ ТОВАРА	ЦЕНА	ЕДИНИЦА ИЗМЕРЕНИЯ
1	2	3	4

4) Таблица поступления-отгрузки товаров:

ГОД	МЕСЯЦ	НОМЕР СКЛАДА	КОД ТОВАРА	ОПЕРАЦИЯ ПОСТУПЛЕНИЯ-ОТГРУЗКИ	КОЛИЧЕСТВО
1	2	3	4	5	6

Замечание: в графе 5 отображается код операции: 1-поступление, 2-отгрузка.

Приложение должно выполнять функции, представленные пунктами меню.

1) Расчет общего количества товаров каждого кода, поступившего на каждый склад и отгруженного с каждого склада на основе сведений таблицы 4 и занесение рассчитанных данных в колонки 6 и 7 таблицы 1. Выполнить с использованием модуля VBA.

2) Просмотр и печать сведений о движении товаров на складах базы :

НО- МЕР СКЛ АДА	НАИМЕ- НОВА- НИЕ ТО- ВАРА	Ц Е Н А	ОСТАТКИ НА НА- ЧАЛО МЕСЯЦА		ПОСТУПЛЕНИЕ		ОТГРУЗКА		ОСТАТКИ НА КО- НЕЦ МЕСЯЦА	
			КОЛИ- ЧЕСТВО	СУМ- МА	КОЛ- ВО	СУМ- МА	КОЛ- ВО	СУМ- МА	КОЛИ- ЧЕСТВО	СУМ- МА
1	2	3	4	5	6	7	8	9	10	11

Данные в Гр.3 берутся из справочника инструментов. Гр.5=Гр.3\*Гр.4;

Гр.7=Гр.3\*Гр.6; Гр.9=Гр.3\*Гр.8; Гр.10=Гр.4+Гр.6-Гр.8; Гр.11=Гр.3\*Гр.10.

При печати получить итоги по каждому складу и общие итоги по ведомости в графах: 5, 7, 9, 11.

3) Просмотр сведений о наличии товаров на складах базы в виде:

НАИМЕНОВАНИЕ ТОВАРА	НОМЕР СКЛАДА	ЦЕНА	ОСТАТОК НА КОНЕЦ МЕСЯЦА	
			КОЛ-ВО	СУММА
1	2	3	4	5

Рассортировать по наименованию товара

4) Просмотр сведений о товарах, ценой >50 000 руб. по возрастанию цены

5) Используя форму с подчиненной формой, выполнить:

Для каждого склада получить на экране список товаров, имеющихся в наличии с итогом по графе “сумма” в виде :

НОМЕР АПТЕКИ	НАИМЕНОВАНИЕ ЛЕКАРСТВА	ЦЕНА	ОСТАТКИ НА КОНЕЦ МЕСЯЦА	
			КОЛ-ВО	СУММА
1	2	3	4	5

6) Все указанные действия должны выполняться в текущем и в следующем месяце . При копировании данных табл. №1 в новый месяц значение Гр. 5 рассчитать по формуле: старое значение Гр. 5 + Гр.6 – Гр.7. В графы 6 и 7 занести 0.

## 11) Отгрузка товаров со складов базы

Исходные таблицы:

1) Сведения об отгрузке товаров со складов базы в текущем месяце:

ГОД	МЕСЯЦ	НОМЕР СКЛАДА	КОД ГРУЗОПОЛУЧАТЕЛЯ	ИНВЕНТАРНЫЙ № ТОВАРА	КОЛИЧЕСТВО	ДАТА ОТГРУЗКИ
1	2	3	4	5	6	7

2) Справочник складов:

№ СКЛАДА	НАЗВАНИЕ СКЛАДА	ФИО ЗАВЕДУЮЩЕГО
1	2	3

3) Справочник товаров:

ИНВЕНТАРНЫЙ № ТОВАРА	НАИМЕНОВАНИЕ ТОВАРА	ЦЕНА	ЕДИНИЦА ИЗМЕРЕНИЯ
1	2	3	4

4) Справочник грузополучателей:

КОД ГРУЗОПОЛУЧАТЕЛЯ	НАИМЕНОВАНИЕ	АДРЕС
1	2	3

Приложение должно выполнять следующие функции, оформленные как пункты меню.

1) Просмотр и печать сведений об отгрузке товаров со складов базы:

№ СКЛАДА	НАЗВАНИЕ ГРУЗОПОЛУЧАТЕЛЯ	НАИМЕНОВАНИЕ ТОВАРА	ДАТА ОТГРУЗКИ	КОЛИЧЕСТВО	ЦЕНА	СТОИМОСТЬ
1	2	3	4	5	6	7

Графа 6 выбирается из справочника товаров; Гр.7=Гр.6\*Гр.5.

При печати получить итоги по каждому магазину и общий итог по ведомости в графах «количество» и «стоимость».

2) Просмотр сведений об отгрузке товаров со складов базы в той же форме что в п.1, но сортировка в алфавитном порядке наименований товаров.

3) Выбрать только сведения о товарах, цена которых > 200 000 рублей. Вывести в той же форме, но в порядке возрастания цены.

4) Используя форму с подчиненной формой выполнить:

Для каждого склада получить на экране список отгруженных в текущем месяце товаров с итогом в графе «стоимость» в виде:

№ СКЛАДА	НАИМЕНОВАНИЕ ТОВАРА	КОЛИЧЕСТВО	СТОИМОСТЬ
1	2	3	4

5) Рассчитать общую стоимость товаров, отгруженных с каждого из складов (использовать модуль VBA). Результат вывести в таблицу

ГОД	МЕСЯЦ	НОМЕР СКЛАДА	СТОИМОСТЬ ОТГРУЖЕННЫХ ТОВАРОВ
1	2	3	4

6) Все указанные действия должны выполняться в текущем месяце и в следующем. При копировании записей таблицы №1 в нее же с новым значением поля «месяц» и, если нужно, поля «год» в графу «количество» и «дата отгрузки» занести 0.

## 12) Операции с поставщиками

Исходные таблицы:

1) Сведения о поставках сырья на предприятие в текущем месяце:

НОМЕР НАКЛАДНОЙ	КОД ПОСТАВЩИКА	КОД СЫРЬЯ	КОЛИЧЕСТВО	ДАТА ПОСТАВКИ
2	4	5	6	7

2) Справочник видов сырья:

КОД СЫРЬЯ	НАИМЕНОВАНИЕ СЫРЬЯ	ЕДИНИЦА ИЗМЕРЕНИЯ	ЦЕНА
1	2	3	4

3) Справочник поставщиков:

КОД ПОСТАВЩИКА	НАИМЕНОВАНИЕ ПОСТАВЩИКА	АДРЕС	ТЕЛЕФОН
1	2	3	4

Приложение должно выполнять функции, оформленные как пункты меню.

1) Просмотр и печать сведений о поставках сырья на предприятие:

НОМЕР НАКЛАДНОЙ	НАИМЕНОВАНИЕ ПОСТАВЩИКА	НАИМЕНОВАНИЕ СЫРЬЯ	ДАТА ПОСТАВКИ	КОЛИЧЕСТВО	ЦЕНА	СТОИМОСТЬ
1	2	3	4	5	6	7

Графа 6 выбирается из справочника товаров;  $Гр.7 = Гр.6 * Гр.5$ .

При печати получить итоги по каждому поставщику и общий итог по ведомости в графах «количество» и «стоимость».

2) Просмотр сведений о поставках сырья на предприятие в той же форме, что в п. 1, но сортировка в алфавитном порядке наименований сырья.

3) Выбрать только сведения о видах сырья, цена которых > 200 000 рублей. Вывести в той же форме, но в порядке возрастания цены.

4) Используя форму с подчиненной формой выполнить: для каждого поставщика получить на экране список поставленных им видов сырья в текущем месяце с итогом в графе «стоимость»:

КОД ПОСТАВЩИКА	НАИМЕНОВАНИЕ СЫРЬЯ	ЦЕНА	КОЛИЧЕСТВО	СТОИМОСТЬ
1	2	3	4	5

5) Рассчитать общую стоимость сырья, поставленного каждым из поставщиков (создать модуль). Результат записать в таблицу со следующей структурой.

КОД ПОСТАВЩИКА	МЕСЯЦ	СТОИМОСТЬ ПОСТАВЛЕННОГО СЫРЬЯ
1	2	3

В следующем месяце поставки сырья будут осуществляться в подобном режиме. Скопировать записи текущего месяца таблицы 1 в нее же, при копировании в поля НОМЕР НАКЛАДНОЙ и ДАТА ПОСТАВКИ занести нули.

## ЛИТЕРАТУРА

1. Кренке, Д. Теория и практика построения баз данных / Д. Кренке. – Питер, 2005.
2. Диго, С.М. Базы данных / С.М. Диго. – Москва : Финансы и статистика, 2005.
3. Гетц, К. Access : сборник рецептов для профессионалов / К. Гетц, П. Литвин, Э. Бэрон. – Питер, 2005.
4. Официальный учебный курс Microsoft: Microsoft Office Access 2003.– Москва : ЭКОМ ; БИНОМ. Лаборатория знаний, 2006.
5. Системы баз данных: проектирование, реализация и управление / Роб Питер, Коронел Карлос. – СПб., 2004.
6. Глушаков, С.В. Базы данных : учебный курс / С.В. Глушаков, Д.Б. Ломотько. – Харьков : Фолио; Ростов-н/Д : Феникс ; Киев : Абрис, 2000.
7. Базы данных : учебник для высших учебных заведений / А.Д. Хомоненко [и др.] ; под ред. Ф.Д. Хомоненко. – СПб., 2000.
8. Карпова, Е.С. Базы данных: модели, разработка, реализация : учебник / Е.С. Карпова. – СПб., Москва, Харьков, Минск, Питер, 2001.
9. Герцев, В.А. Создание БД в MS ACCESS / В.А. Герцев. – Минск : НИУП, 2001.
10. Коуров, Л.В. Общие вопросы проектирования БД. Основы языка SQL /Л.В. Коуров. – Минск : НИУП, 2001.
11. Базы данных. Проектирование, реализация и сопровождение. Теория и практика : учеб. пособие / Т. Конноли [и др.] ; пер. с англ. ; под ред. А.В. Слепцова. – Москва : Вильямс, 2000.
12. Федоров, А.Г. Базы данных для всех / А.Г. Федоров, Н.Э. Елманова. – Москва : Компьютер Пресс, 2001.
13. Новалис, С. Access : руководство по макроязыку и VBA / С. Новалис. – Москва : Лори, 2000.

Учебное издание

## **ТЕХНОЛОГИИ БАЗ ДАННЫХ И ЗНАНИЙ**

*Методические указания и задания к практическим занятиям  
для студентов экономических специальностей  
(Раздел «MS Access»)*

Составитель

**Ероховец** Тамара Викторовна

Ответственный за выпуск *О.Л. Сапун*

Редактор *А.П. Бондич*

Верстка *А.П. Бондич*

Подписано в печать 16.03.2008 г. Формат 60×84<sup>1</sup>/<sub>16</sub>  
Бумага офсетная. Гарнитура Times New Roman. Усл. печ. л. 4,7.  
Уч.-изд. л. 3,8. Тираж 220 экз. Заказ 257.

Издатель и полиграфическое исполнение  
Белорусский государственный аграрный технический университет  
ЛИ № 02330/0131734 от 10.02.2006. ЛП № 02330/0131656 от 02.02.2006.  
220023, г. Минск, пр. Независимости, 99, к. 2.