

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА
И ПРОДОВОЛЬСТВИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра экономической информатики

КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (СУБД Ms Access)

*Учебно-методический комплекс
по дисциплине «Компьютерные информационные технологии»
для студентов специальности
1-25 01 07 Экономика и управление на предприятии,
по дисциплине «Информационные технологии»
для студентов специальности
1-26 02 02 Менеджмент*

Минск
БГАТУ
2011

УДК 004.65(07)
ББК 32.97я7
К63

*Рекомендовано научно-методическим советом факультета
предпринимательства и управления БГАТУ.
Протокол № 2 от 23 ноября 2010 г.*

Составители:

старший преподаватель Т. В. Ероховец,
старший преподаватель Е. Г. Гриневич

Рецензенты:

доцент кафедры Менеджмента технологий Института бизнеса и менеджмента технологий
БГУ, кандидат технических наук, доцент *Ю. Н. Силкович*;
доцент кафедры вычислительной техники БГАТУ,
кандидат технических наук, доцент *А. И. Шакирин*

Компьютерные информационные технологии (СУБД MS Access) : учебно-методический комплекс / сост. : Т. В. Ероховец, Е. Г. Гриневич. – Минск : БГАТУ, 2011. – 204 с.
ISBN 978-985-519-348-8.

Учебно-методический комплекс содержит три модуля, в которых рассматриваются основы проектирования и создания баз данных в среде наиболее популярной и доступной настольной СУБД Access.

Учебно-методический комплекс ориентирован на теоретическое изучение и приобретение навыков разработки различных приложений баз данных, реализующих элементы задач экономики и управления.

УМК предназначен для студентов высших учебных заведений.

УДК 004.65(07)
ББК 32.97я7

ВВЕДЕНИЕ

В настоящее время существует множество систем управления базами данных (СУБД), среди которых особого внимания заслуживает Microsoft Access. СУБД MS Access поддерживает современные подходы к созданию и управлению базами данных, интегрирована с другими приложениями MS Office, имеет средства для работы в глобальной сети Internet и в корпоративных сетях. Эти достоинства позволяют решать в СУБД MS Access широкий круг современных задач.

Цель данного раздела дисциплины «Компьютерные информационные технологии» – приобретение теоретических сведений о современных технологиях организации, хранения и обработки данных в базах и практических навыков по созданию баз данных и работе с ними.

Задачи дисциплины

Дисциплина «Компьютерные информационные технологии» должна развивать у студентов компетенции:

- **академические:** владеть и применять базовые знания для решения теоретических и практических задач; уметь работать самостоятельно; владеть междисциплинарным подходом при решении проблем;
- **социально-личностные:** обладать способностью к межличностным коммуникациям; уметь работать в команде;
- **профессиональные:** уметь создавать и поддерживать в актуальном состоянии информационно-аналитическую базу организации (предприятия) с использованием современных средств и методов обработки данных; осуществлять поиск, систематизацию и анализ информации по перспективам развития отрасли, инновационным технологиям, проектам и решениям.

В результате освоения учебного материала дисциплины **студент должен знать:**

- принципы организации баз данных и их проектирования;
- назначение, архитектуру СУБД и направления их развития;
- возможности языка баз данных SQL;

- технологии обработки многопользовательских баз данных;
уметь проектировать, создавать и использовать базы данных.

Для изучения дисциплины необходимы знания, полученные при изучении «Высшей математики» (разделы: векторы и матрицы; функции и графики функций), а также изучении дисциплины «Компьютерные информационные технологии» в 1-ом семестре.

Примерный тематический план по дисциплине «Компьютерные информационные технологии» (СУБД MS Access) представлен в таблице 1.

Таблица 1

Примерный тематический план по дисциплине
«Компьютерные информационные технологии» (СУБД MS Access)

Номер и название модуля	Общее количество часов на модуль	В том числе				Всего по модулю УСПС (час)
		Лекции и (час)	УСПС (час)	Лабораторные занятия (час)	УСПС (час)	
2 семестр						
М-4. Теоретические основы баз данных и знаний. Проектирование базы данных в среде СУБД «MS Access». Формирование запросов к базе данных	30	16	4	14	2	6
М-5. Язык SQL. Проектирование приложений базы данных – запросов-действий, форм, отчетов	26	12	3	14	3	6
М-6. Автоматизация обработки данных в системах управления базами данных Системы обработки многопользовательских баз данных	12	6		6	2	2
Всего	68	34	7	34	7	14

МОДУЛЬ 4

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ БАЗ ДАННЫХ И БАЗ ЗНАНИЙ. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ В СРЕДЕ СУБД «MS ACCESS». ФОРМИРОВАНИЕ ЗАПРОСОВ К БАЗЕ ДАННЫХ

В результате изучения модуля студент должен

- **знать:** понятия и определения теории баз данных (БД), сущность основных моделей представления данных; особенности современных систем управления базами данных (СУБД); принципы и этапы проектирования БД, пользовательский интерфейс СУБД MS Access; принципы и способы конструирования таблиц, способы поддержания целостности данных в БД; сущность и назначения запросов различных типов, способы создания запросов;
- **уметь:** применять полученные теоретические знания в процессе организации данных, проектировать простейшие БД; разрабатывать структуры таблиц, создавать и связывать таблицы в СУБД MS Access; создавать простые формы для заполнения данными таблиц, конструировать запросы различных типов.

КРАТКАЯ ХАРАКТЕРИСТИКА СУБД MS 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 – 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.

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 World Wide Web.

10. **Вложение**. Вложения можно использовать для хранения нескольких файлов в одном поле, причем в этом поле можно хранить файлы разных типов (документы Microsoft Office Word 2007, презентации Microsoft Office PowerPoint 2007, изображения и т. п.).

11. **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 4.1 приведены наборы свойств полей часто используемых типов данных.

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)).

Таблица 4.1

Типы данных в СУБД Access

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
Смарт-теги	Предложений IME Смарт-теги	IME Смарт-теги

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. Задаёт 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.

Десять преимуществ СУБД Access 2007

1. Быстрое получение нужных результатов с помощью пользовательского интерфейса Office Fluent.

2. Ускоренное освоение приложения благодаря использованию встроенных решений.

3. Создание нескольких отчетов с разными представлениями одних и тех же данных.

4. Быстрое создание таблиц без применения сложных операций управления базой данных, благодаря функции автоматического обнаружения типов данных.

5. Новые типы полей, расширяющие возможности пользователей, такие как вложения и многозначные поля.

6. Сбор и обновление сведений непосредственно из источника.

7. Общий доступ к данным через Microsoft Windows SharePoint Services.

8. Отслеживание списков служб Windows SharePoint Service с использованием разнообразных клиентских возможностей Office Access 2007.

9. Перемещение данных в службах Windows SharePoint Services для оптимизации управления.

10. Доступ к данным и использование информации из нескольких источников.

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, 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 преобразование 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.**

Проектирование базы данных ВУЗ

Требуется создать базу данных для обработки информации об учебном процессе.

В результате первого этапа проектирования базы данных получены следующие внешние представления.

Первое внешнее представление соответствует требованиям учебного отдела и содержит информацию об экзаменационной сессии. Необходимо составить расписание проведения преподавателями экзаменов сессии в группах факультета. 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 представлено в таблице 4.2.

Таблица 4.2

Исходные данные для создания БД ВУЗ

<i>Преподаватель</i>	<i>Должность</i>	<i>Группа</i>	<i>Дисциплина</i>	<i>Дата экзамена</i>	<i>Время начала</i>	<i>Время окончания</i>
Иванов	Профессор	73эи	КИТ	05.06.10	8:15	12:30
Астрахан	Доцент	73эи	Выс. математика	10.06.10	10:00	14:00
Астрахан	Доцент	73эи	ЭММ	14.06.10	8:15	12:30
Квачук	Профессор	73эи	Макроэкономика	26.06.10	8:15	14:00
Петров	Ст.препод.	73эи	История Беларуси	30.06.10	10:00	14:30
Иванов	Профессор	68им	ИТ	04.06.10	8:30	15:00
Астрахан	Доцент	68им	Выс. математика	09.06.10	10:00	14:30
Астрахан	Доцент	68им	ЭММ	18.06.10	8:30	12:20
Квачук	Профессор	68им	Макроэкономика	24.07.10	10:00	15:30
Петров	Ст.препод.	68им	История Беларуси	29.06.10	12:00	16:30
Астрахан	Доцент	5эво	Всемирная история	11.06.10	8:15	13:30

Второе внешнее представление соответствует требованиям деканата и должно содержать информацию для составления экзаменационных ведомостей по каждому экзамену.

Третье внешнее представление соответствует требованиям студенческого отдела кадров и содержит информацию о студентах.

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 необходи-

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

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

В нашем внешнем представлении участвуют простые типы сущностей: **Преподаватели, Группы, Дисциплины, Должности** и агрегированный тип сущностей – **Экзамены**.

В предметной области объекты **взаимодействуют** друг с другом посредством **связей (relationships)**.

Взаимодействия между двумя сущностями, например, **Преподаватели** и **Экзамены** представлены фразами «*принимает несколько*» и «*принимается одним*». Преподаватель принимает несколько экзаменов, но каждый экзамен принимается лишь одним преподавателем.

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 (рис. 4.1, 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 сущностей В соответствует только один экземпляр типа сущностей А. (рис. 4.1, б).

Связь «**многие – ко – многим**» означает, что каждому экземпляру одного типа сущностей (А) соответствует один или более экземпляров другого типа сущностей (В), и наоборот (рис. 4.1, в). В реляционных БД этот тип связи сводится к связи «**один – ко – многим**».

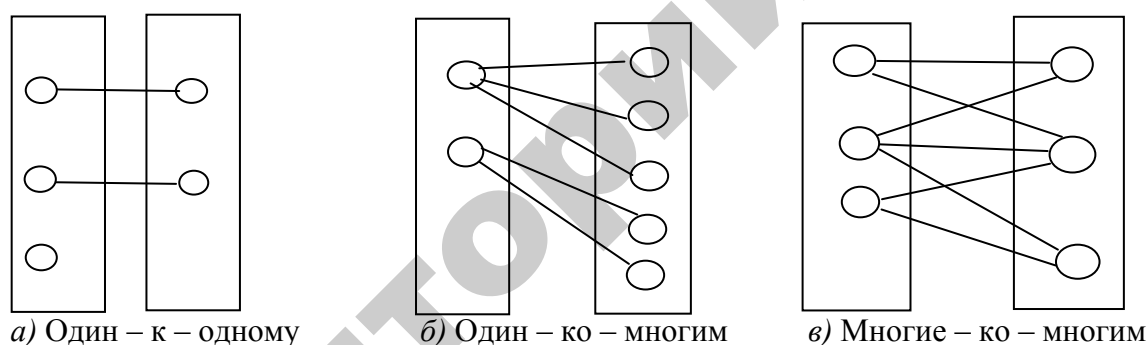



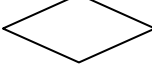


Рис. 4.1. Виды связей между таблицами

В нашем примере каждый преподаватель может принимать несколько экзаменов, но каждый экзамен принимается только одним преподавателем, поэтому связь между сущностями **Преподаватели** и **Экзамены** типа 1:M. Аналогичный тип связи имеют и все остальные связи (**Группы – Экзамены**, **Дисциплины – Экзамены**, **Должности – Преподаватели**).

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. Основные графические примитивы модели «Сущность-Связь» представлены в таблице 4.3.

Основные графические примитивы модели «СущностьСвязь»

Название примитива	Одно из современных обозначений
Сущность	
Свойство	
Первичный Ключ	
Связь	

Получена следующая ER-диаграмма внешнего представления (рис. 4.2).

При определении типов сущностей **Преподаватель**, **Группа**, **Дисциплина**, **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 деканата и соответствует следующей ER-диаграмме (рис. 4.3).

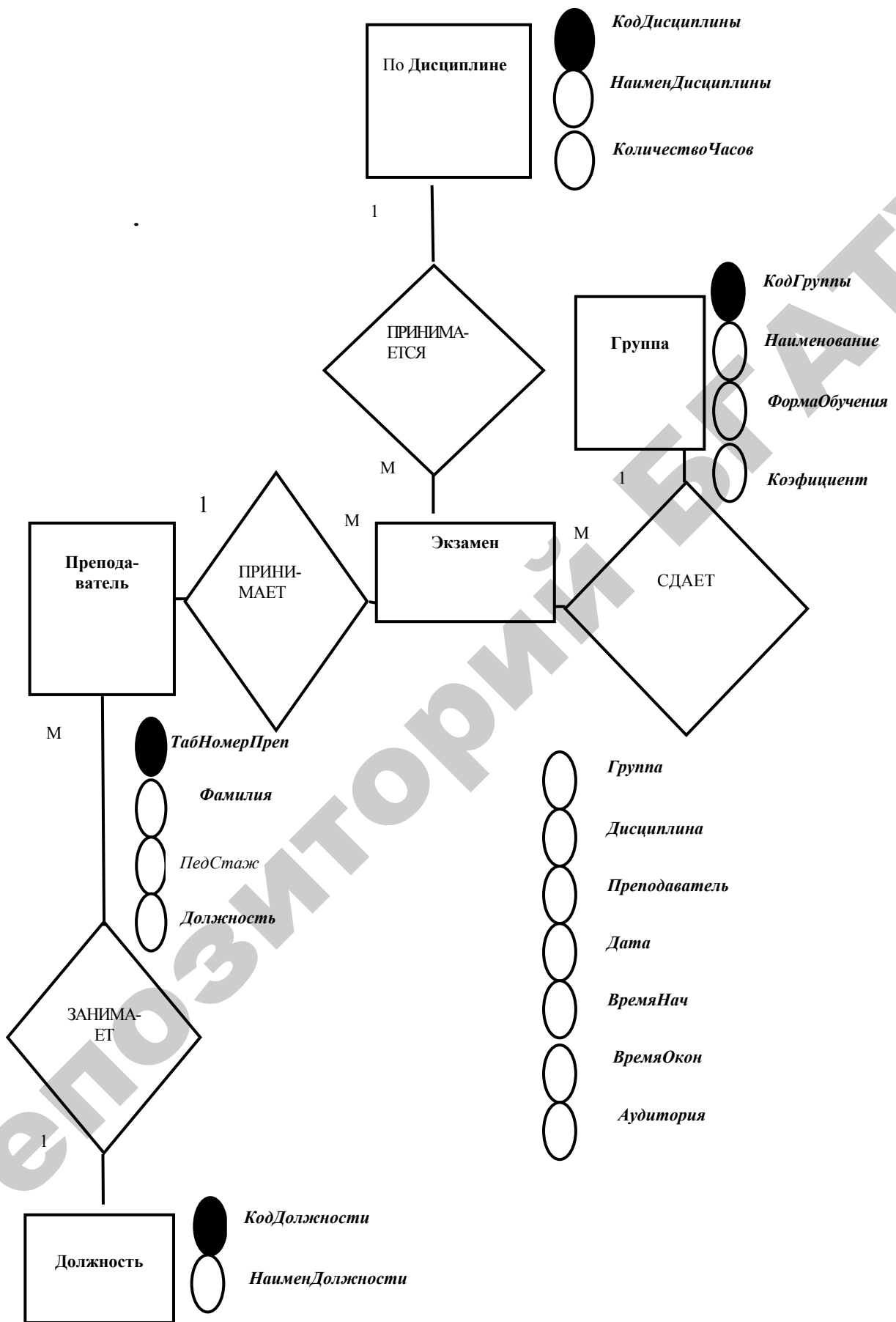


Рис. 4.2. ER-Диаграмма для первого внешнего представления

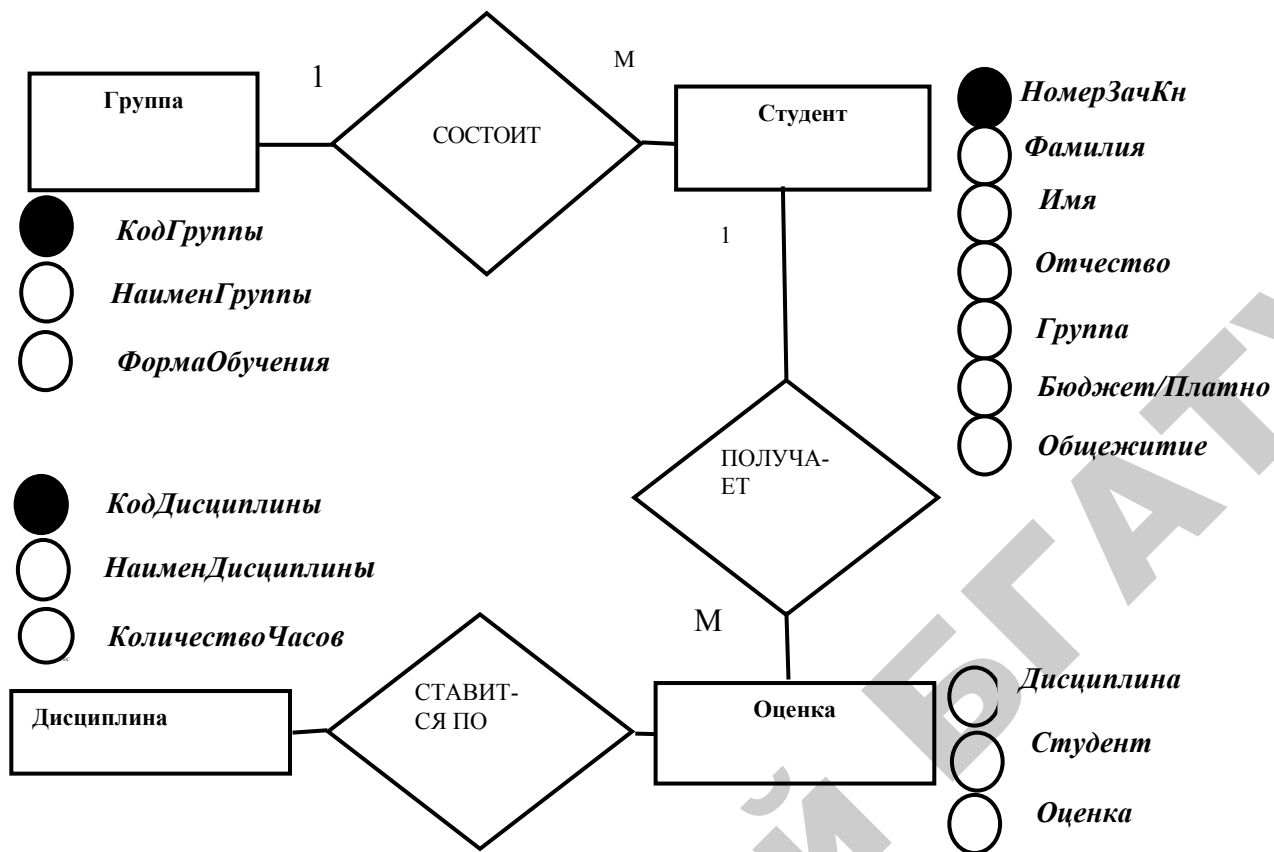


Рис. 4.3. ER-диаграмма для второго внешнего представления

Третье внешнее представление описывает информацию для студенческого отдела кадров и соответствует следующей ER-диаграмме (рис. 4.4).

Объединяя три внешних представления, получаем концептуальную схему проектируемой базы данных. Назовем ее **ВУЗ** (рис. 4.5).

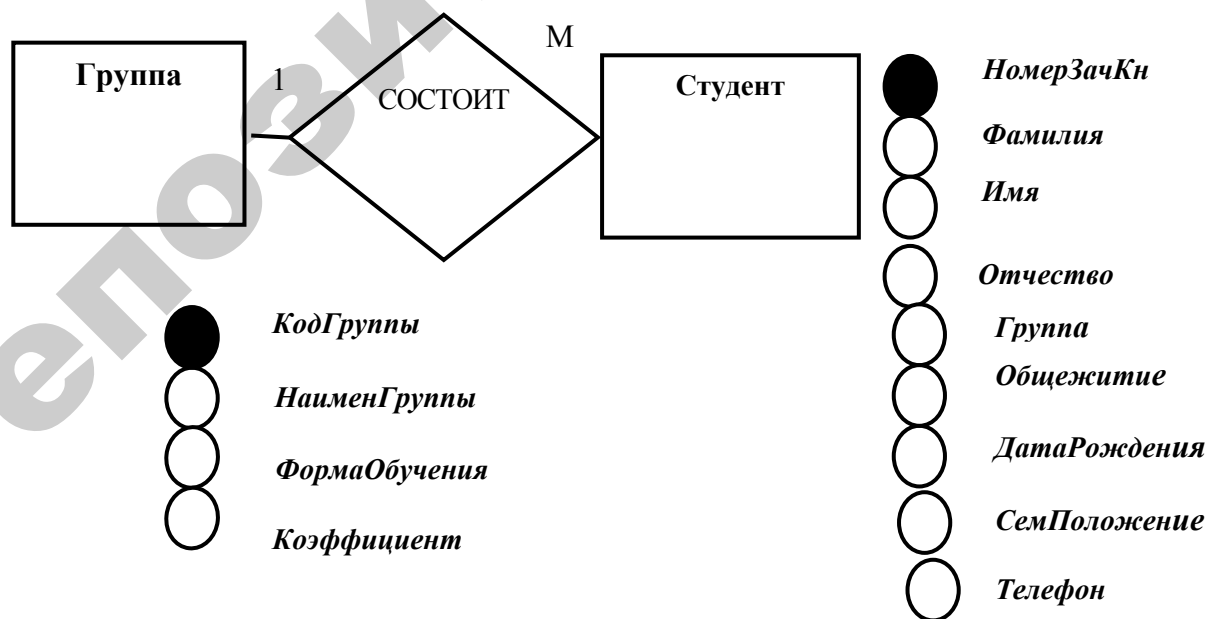


Рис. 4.4. ER-диаграмма для третьего внешнего представления

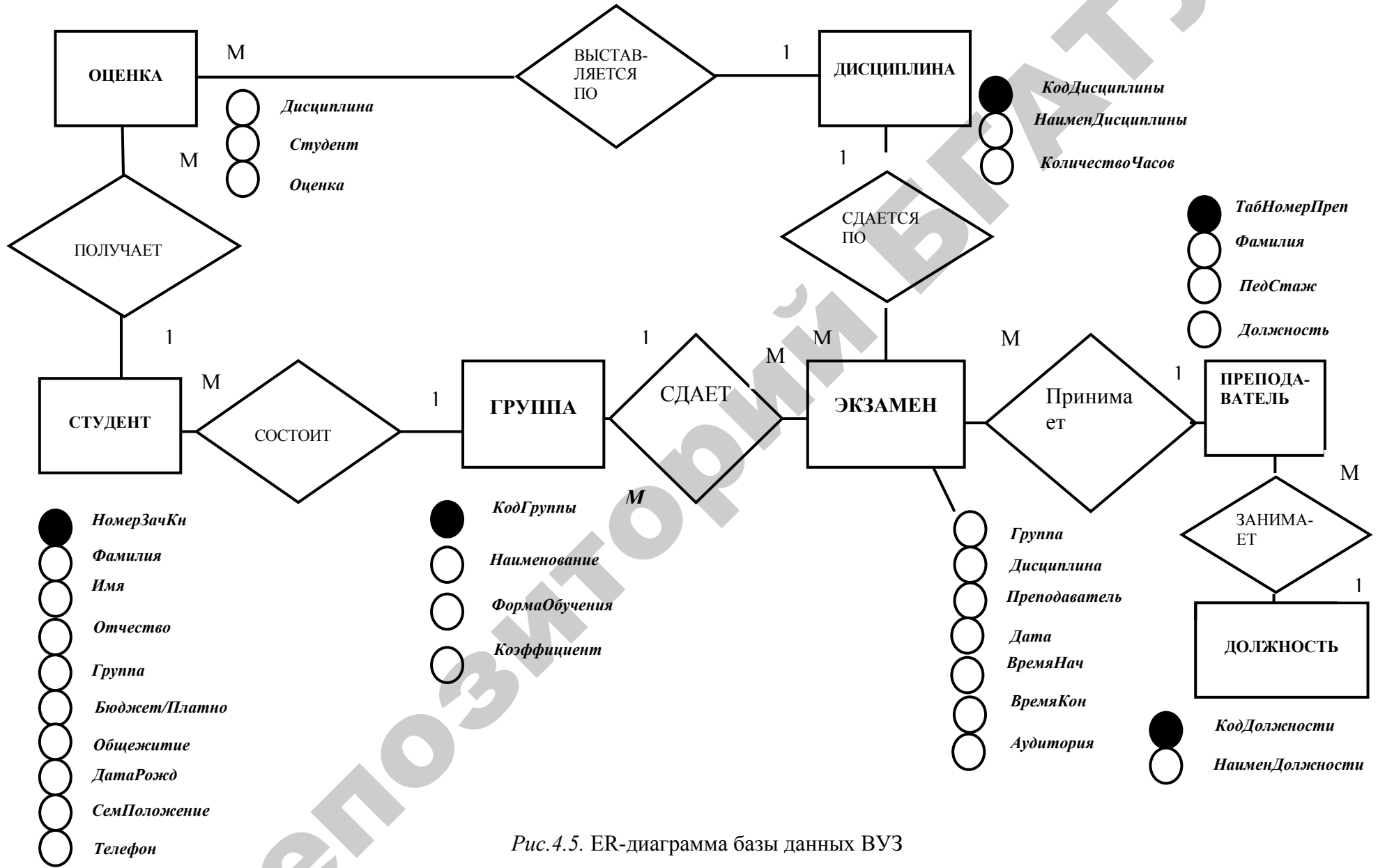


Рис.4.5. ER-диаграмма базы данных ВУЗ

В процессе логического проектирования, с ориентацией на СУБД MS Access и с учетом законов нормализации базы данных, получена следующая совокупность логически связанных таблиц (таблицы 4.4–4.17).

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

1. Таблица **Экзамены** (таблица 4.4) и ее структура записи (таблица 4.5) БД ВУЗ. Таблица **Экзамены** соответствует агрегированной сущности (процессу приема экзаменов), поэтому в ней **не нужно определять первичный ключ!!!**

Таблица 4.4

Таблица **Экзамены** БД ВУЗ

Группа	Дисциплина	Преподаватель	Дата	ВремяНач	ВремяКон	Аудитория
1	1	1	05.06.2010	8:15	12:30	135
1	3	3	10.06.2010	10:00	14:00	133
1	4	3	14.06.2010	8:15	12:30	315a
1	5	4	26.06.2010	8:15	14:00	133
1	6	2	30.06.2010	10:00	14:30	135
2	2	1	04.06.2010	8:30	15:00	137
2	3	3	09.06.2010	10:00	14:30	241
2	4	3	18.06.2010	8:30	12:20	241
2	5	4	24.07.2010	10:00	15:30	315a
2	6	2	29.06.2010	12:00	16:30	135

Таблица 4.5

Структура записи таблицы **Экзамены** БД ВУЗ

Имя поля	Тип данных	Описание
Группа	– числовой (длинное целое)	Код группы
Дисциплина	– числовой (длинное целое)	Код дисциплины
Преподаватель	– числовой (длинное целое)	Код преподавателя
Дата	– дата/время	Дата экзамена
ВремяНач	– дата/время	Время начала экзамена
ВремяКон	– дата/время	Время окончания экзамена
Аудитории	– текстовый	Номер аудитории

2. Таблица **Группы** (таблица 4.6) и ее структура записи (таблица 4.7) БД ВУЗ. Таблица **Группы** соответствует сущности Группа и по смыслу является *справочником групп*, поэтому *первичным ключом* нужно определить поле КодГруппы.

Таблица 4.6

Таблица Группы БД ВУЗ

↔КодГруппы	НаименГруппы	ФормаОбучения	Коэффициент
1	73эи	Дневная	1,2
2	68им	Дневная	1,2
3	72эи	Дневная	1,4
4	5тк	Заочная	0,3
5	64м	Вечерняя	1

Таблица 4.7

Структура записи таблицы Группы БД ВУЗ

Имя поля	Тип данных	Описание
↔ КодГруппы	– числовой (длинное целое)	Код группы (первичный ключ)
НаименГруппы	– текстовый	Наименование группы
ФормаОбучения	– текстовый	Форма обучения
Коэффициента	– числовой (плав. точка)	Коэффициент

3. Таблица **Преподаватели** (таблица 4.8) и ее структура записи (таблица 1.9) БД ВУЗ. Таблица **Преподаватели** соответствует сущности Преподаватель и по смыслу является *справочником преподавателей*, поэтому *первичным ключом* нужно определить поле ТабНомПр.

Таблица 4.8

Таблица Преподаватели БД ВУЗ

↔ТабНомПреп	Фамилия	ПедСтаж	Должность
1	Иванов	25	1
2	Петров	5	3
3	Астрахан	15	2
4	Квачук	30	1
5	Чиж	9	2

Таблица 4.9

Структура записи таблицы Преподаватели БД ВУЗ

Имя поля	Тип данных	Описание
↔ ТабНомПреп	– числовой(длин целое)	Код преподавателя (первич. ключ)
Фамилия	– текстовый	Фамилия преподавателя
ПедСтаж	– числовой (длин целое)	Педагогический стаж преподавателя
Должность	– числовой (длин целое)	Код должности преподавателя

4. Таблица **Студенты** (таблица 4.10) и ее структура записи (таблица 4.11) БД ВУЗ. Таблица **Студенты** соответствует сущности **Студент** и по смыслу является *справочником студентов*, поэтому *первичным ключом* нужно *определить поле НомерЗачКн*.

Таблица 4.10

Таблица Студенты БД ВУЗ

№-НомерЗачКн	Фамилия	Имя	Отчество	Группа	БюджетОбуч	Общезажитие	ДатаРожд	СемПоложение	Телефон
92001	Чиж	Анна	Петровна	1	Да	Нет	11.11.1991	Да	255-64-95
92002	Кисель	Александр	Ивановна	1	Нет	Да	15.06.1992	Нет	288-45-89
92005	Баланович	Алеся	Александровна	1	Да	Да	25.08.1992	Нет	234-56-78
94026	Корень	Владимир	Эдуардович	2	Да	Нет	15.04.1992	Нет	288-50-65
94041	Стах	Оксана	Юрьевна	2	Нет	Да	13.08.1990	Да	678-94-56
94048	Степанов	Сергей	Александрович	2	Нет	Нет	15.07.1991	Нет	288-65-55

Таблица 4.11

Структура записи таблицы Студенты БД ВУЗ

Имя поля	Тип данных	Описание
№-НомерЗачКн	– числовой (длинное целое)	Номер зачетной книжки (первич ключ)
Фамилия	– текстовый	Фамилия студента
Имя	– текстовый	Имя студента
Отчество	– текстовый	Отчество студента
Группа	– числовой (длинное целое)	Код группы
Бюджет	– логический (да/нет)	Форма обучения (бюджет/платное)
Общезажитие	– логический (да/нет)	Номер аудитории
ДатаРождения	– дата/время	Дата рождения студента
СемПоложение	– логический (да/нет)	Семейное положение студента
Телефон	– текстовый	Телефон студента

5. Таблица **Должности** (таблица 4.12) и ее структура записи (таблица 4.13) БД ВУЗ. Таблица **Должности** соответствует сущности **Должность** и по смыслу является *справочником должностей*, поэтому *первичным ключом* нужно *определить поле КодДолжности*.

Таблица 4.12

Таблица Должности БД ВУЗ

КодДолжности	НаименДолжности
1	Профессор
2	Доцент
3	Старший преподаватель
4	Ассистент

Таблица 4.13

Структура записи таблицы Должности БД ВУЗ

Имя поля	Тип данных	Описание
☛ КодДолжности	– числовой (длин. целое)	Код преподавателя (первич ключ)
НаименДолжности	– текстовый	Наименование должности преподавателя

6. Таблица **Дисциплины** (таблица 4.14) и ее структура записи (таблица 4.15) БД ВУЗ. Таблица **Дисциплины** соответствует сущности Дисциплина и по смыслу является *справочником дисциплин*, поэтому *первичным ключом нужно определить поле КодДисциплины*.

Таблица 4.14

Таблица Дисциплины БД ВУЗ

КодДисциплины	НаименДисциплины	КоличествоЧасов
1	КИТ	68
2	ИТ	68
3	Высшая математика	60
4	ЭММ	51
5	Макроэкономика	85
6	История Беларуси	48
7	Всемирная история	64

Таблица 4.15

Структура записи таблицы Дисциплины БД ВУЗ

Имя поля	Тип данных	Описание
☛ КодДисциплины	– числовой (длин целое)	Код дисциплины (первичный ключ)
НаимДисциплины	– текстовый	Наименование дисциплины
КоличествоЧасов	– числовой (длин целое)	Количество часов дисциплины

7. Таблица **Оценки** (таблица 4.16) и ее структура записи (таблица 4.17) БД ВУЗ. Таблица **Оценки** соответствует агрегированной сущности (процессу получения оценок студентами на экзаменах), поэтому в ней *не нужно определять первичный ключ!!!*

Таблица 4.16

Таблица Оценки БД ВУЗ

Дисциплина	Студент	Оценка
1	86015	6
3	86015	5
...
2	94048	6
2	94048	5

Структура записи таблицы **Оценки** БД ВУЗ

Имя поля	Тип данных	Описание
Дисциплины	– числовой (длин целое)	Код дисциплины
Студенты	– числовой (длин целое)	Номер зачетной книжки студента
Оценка	– числовой (длин целое)	Оценка студента на экзамене

ВНИМАНИЕ! Во всех справочниках коды объектов следует объявить **ключевыми полями**. Ниже приведена информационно-логическая модель проектируемой базы данных (рис. 4.6).

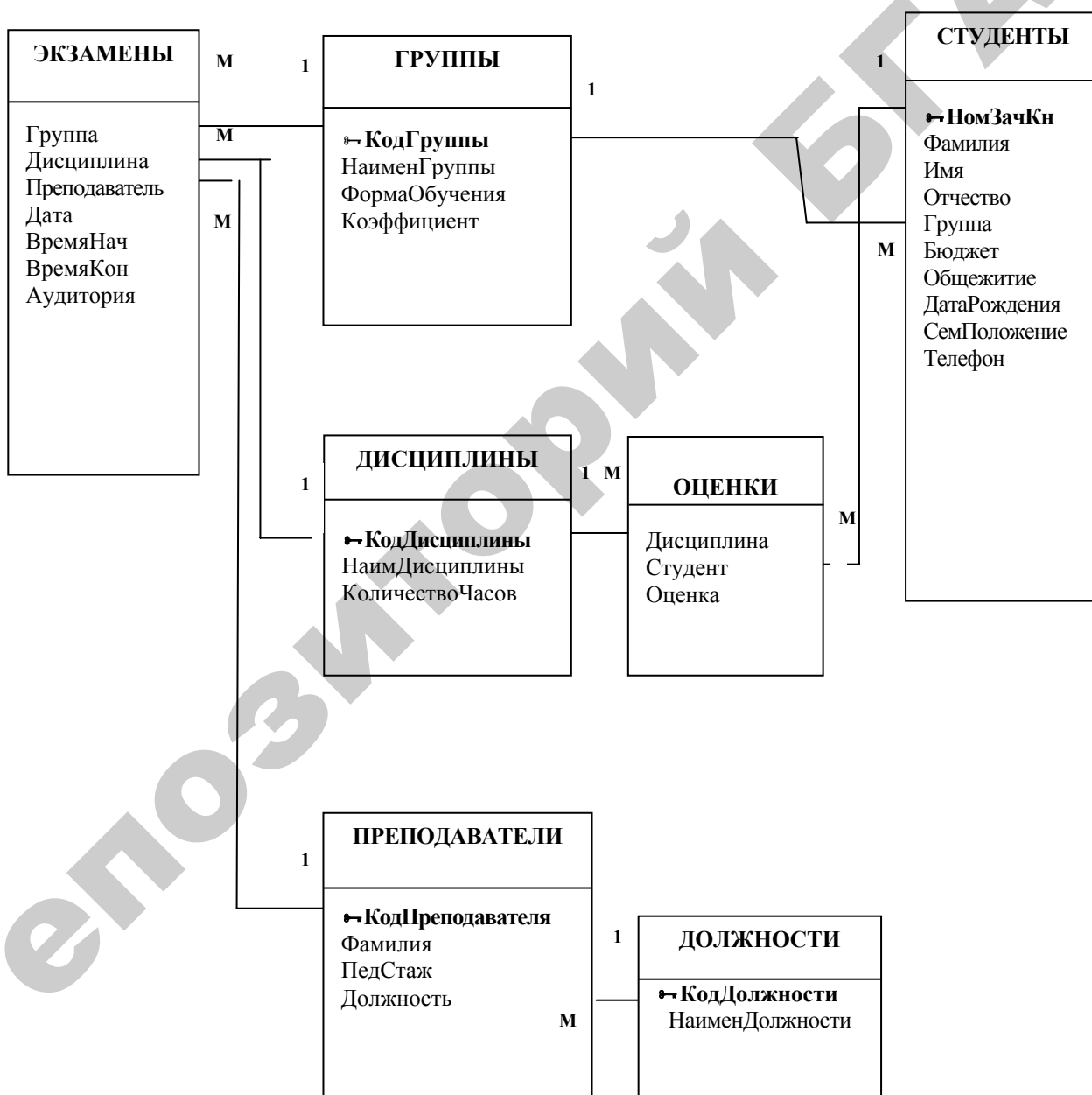


Рис. 4.6. Информационно-логическая модель БД ВУЗ

Далее следует приступить к этапу физического проектирования базы данных.

ЛАБОРАТОРНАЯ РАБОТА № 1

СОЗДАНИЕ БАЗЫ ДАННЫХ В ACCESS. ПОСТРОЕНИЕ ТАБЛИЦ И СХЕМЫ ДАННЫХ

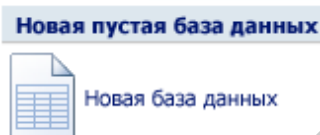
Цель работы: научиться создавать таблицы в БД разными способами, заполнять их данными, определять первичный ключ, грамотно строить схему данных.


Задание 1. Создать БД с именем **ВУЗ** в своей личной папке (в папке **Мои_документы/Фамилия_студента**).

Выполняемые действия

1. Запустить MS Access2007.

2. На странице **Приступая к работе с Microsoft Office Access** в разделе **Новая пустая база данных** выберите команду **Новая база данных**.



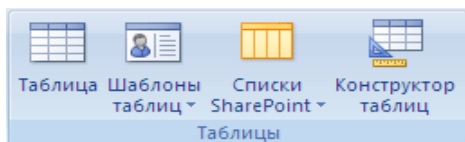
3. В области **Новая база данных** в поле **Имя файла** введите имя файла (**ВУЗ**). Если имя файла указано без расширения, расширение будет добавлено автоматически. Чтобы сохранить файл в другой папке, отличной от используемой по умолчанию, нажмите кнопку **Открыть**  (рядом с полем **Имя файла**), перейдите к нужной папке и нажмите кнопку **ОК**.

4. Нажмите кнопку **Создать**.

Приложение Access создаст базу данных с пустой таблицей с именем **Таблица1** и откроет эту таблицу в режиме таблицы. Курсор находится в первой пустой ячейке столбца **Добавить поле**.

Создание таблиц

Добавление новой таблицы к существующей базе данных осуществляется командами группы **Таблицы** на вкладке **Создать** ленты меню.




Создание таблицы в режиме таблицы

В режиме таблицы можно начать ввод данных сразу, структура таблицы при этом будет создаваться автоматически. Полям присваиваются имена с последовательными номерами (**Поле1**, **Поле2** и т. д.). СУБД Access автоматически задает тип данных для каждого поля в зависимости от вводимых данных. Помните, что можно использовать режим таблицы для выполнения всех задач по разработке таблицы, кроме создания поля подстановок.

Задание 2. Создать таблицу **Дисциплины** в режиме ТАБЛИЦЫ.

Выполняемые действия

а) На вкладке **Создание** в группе **Таблицы** щелкните **Таблица**. 

Приложение Access создаст пустую таблицу с именем **Таблица1** и откроет ее в режиме таблицы. При этом будет выделена первая пустая ячейка столбца **Добавить поле**.


б) Ввести данные (значения полей) в таблицу, начиная со второго столбца. Существующий столбец **Код** данными НЕ ЗАПОЛНЯТЬ!!! Переходить от столбца к столбцу нажатием кнопки **Enter**.


в) Переименовать столбцы таблицы в соответствии с указаниями. Для переименования столбца (поля) дважды щелкните клавишей мыши по заголовку столбца и введите новое имя. Таблица примет вид, подобный рисунку 4.7.

Дисциплины				
	КодДисциплин	НаименДисциплины	КоличествоЧасов	Добавить поле
+	1	КИТ	68	
+	2	ИТ	68	
+	3	Высшая математика	60	
+	4	ЭММ	51	
+	5	Макроэкономика	85	
+	6	История Беларуси	48	
*				

Рис. 4.7. Таблица **Дисциплины** в режиме просмотра

г) Сохранить таблицу с именем **Дисциплины**.

д) Войти в режим **КОНСТРУКТОРА** (можно выделив таблицу **Дисциплины** в области переходов, вызвав контекстное меню (правой клавишей мыши) и нажав кнопку ) , проверить автоматически созданный макет таблицы **Дисциплины** в соответствии с таблицей на рисунке 4.8. Если нужно, то откорректировать макет.

е) Оставаясь в **КОНСТРУКТОРЕ**, выделить и удалить автоматически созданное поле **Код**, объявить поле **КодДисциплины** ключевым полем. Для этого выделить его и, вызвав контекстное меню, щелкнуть клавишей мыши по  кнопке

. Ключевое поле будет помечено таким же значком. Структура таблицы в режиме конструктора изображена на рисунке 4.8.

Рис. 4.8. Таблица **Дисциплины** в режиме **КОНСТРУКТОРА**

ж) Сохранить откорректированную таблицу.

Создание таблицы в режиме **КОНСТРУКТОРА**

В режиме **КОНСТРУКТОРА** сначала создается структура новой таблицы. Затем можно переключиться в режим таблицы для ввода данных или ввести данные, используя другой метод, например вставку или импорт.

Задание 3. Создать таблицу **Группы** с помощью **КОНСТРУКТОРА** таблиц.



Выполняемые действия

1. На вкладке **Создание** в группе **Таблицы** щелкните клавишей мыши по кнопке **Конструктор таблиц** .

2. Для каждого поля в таблице введите имя в столбце **Имя поля**, а затем в списке **Тип данных** выберите тип данных.

При желании можно ввести описание для каждого поля в столбце **Описание**. Это описание будет отображаться в строке состояния, когда в режиме таблицы курсор будет находиться в данном поле. Это описание также отображается в строке состояния для любых элементов управления в форме или отчете, которые создаются путем перетаскивания этого поля из области **списка полей**, и любых элементов управления, которые создаются для этого поля при использовании мастера отчетов или мастера форм.

Для указания типа поля необходимо щелкнуть клавишей мыши в колонке **Тип поля** в правой части описываемого поля, затем щелкнуть клавишей мыши по кнопке

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

Замечание. Размеры текстовых полей **НаименованиеГруппы** и **ФормаОбучения** установить равными 10. Значения свойств поля **Коэффициент**, приведены на рисунке 4.9. Формирование поля **ФормаОбучения** описано в пункте 3. Остальные свойства полей принять по умолчанию.

3. Так как поле **ФормаОбучения** имеет ограниченный список значений, определить для него тип **Подстановка** из списка значений типов данных.


Выполняемые действия

а) В списке типов данных для поля **ФормаОбучения** выбрать **Мастер подстановок**.

б) В открывшемся окне **СОЗДАНИЕ ПОДСТАНОВКИ** выбрать пункт **Будет введен фиксированный набор значений**. Нажать **Далее**.

в) В следующем окне указать число столбцов 1, дважды щелкнуть клавишей мыши по правой границе столбца и ввести в столбец список значений поля: **дневная, заочная, вечерняя**. Нажать **Далее**.

д) В следующем окне задать подпись, которую содержит столбец подстановки: **ФормаОбучения**. Нажать **Готово**.

4. Оставаясь в КОНСТРУКТОРЕ, объявить поле **КодГруппы** ключевым полем. Для этого выделить его и щелкнуть по кнопке . Ключевое поле будет помечено таким же значком. Структура таблицы в конструкторе со значениями свойств для поля коэффициент изображена на рисунке 4.9.

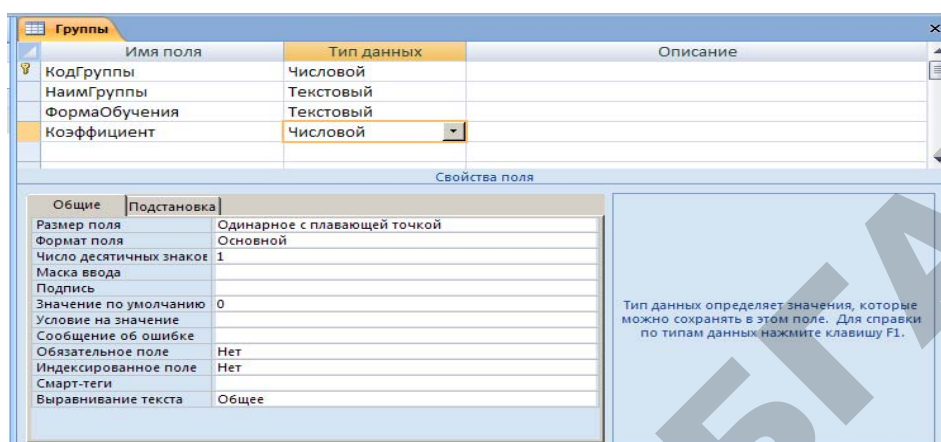



Рис. 4.9. Окно КОНСТРУКТОРА таблиц (структура таблицы **Группы**)

5. Сохраните таблицу.

Щелкните клавишей мыши по значку **Кнопка Microsoft Office** , а затем выберите команду **Сохранить**.

Чтобы ввести (добавить) данные, переключитесь в режим ТАБЛИЦЫ начните их вводить, щелкнув клавишей мыши по первой пустой ячейке. При заполнении поля **ФормаОбучения** значения поля выбирать из списка.

6. После ввода данных таблица **Группы** примет вид, изображенный рисунке 4.10.

КодГруппы	НаименГрупп	ФормаОбуч	Коэффицие	Добавить поле
73эи		дневная	1.2	
2 68им		дневная	1.2	
3 3эи		дневная	1.4	
4 5тк		заочная	0.3	
5 64м		вечерняя	1	
*			0	

Рис. 4.10. Таблица **Группы**

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.

Задание 4. Создать таблицу **Должности** одним из описанных способов (данные взять из таблицы 4.12, структура записи отображена в таблице 4.13).

Импорт, добавление или связывание данных из другого источника

Задание 5. Создать таблицу **Преподаватели**, содержащую сведения о преподавателях. В отделе кадров имеется следующая информация о преподавателях: Табельный номер, Фамилия, Педагогический стаж. Информация организована в виде таблицы Excel. Выполнить импорт информации в таблицу **Преподаватели** из приложения Excel.

Выполняемые действия

1. В группе Внешние данные на вкладке Внешние данные в Access выберите команду для типа файла **Excel**.



2. В диалоговом окне **Внешние данные** нажмите кнопку **Обзор**, чтобы найти файл данных источника, или введите в поле **Имя файла** полный путь к файлу данных источника.

3. Для сохранения данных выбрать пункт

Импортировать данные источника в новую таблицу в текущей базе данных.

4. В окне **Импорт электронной таблицы** указать лист и, если нужно, именованные диапазоны импортируемой информации (у нас *Лист1*). Дважды нажать **Далее**.

5. В следующем окне можно, поочередно выделяя столбцы таблицы, определить имя и тип данных столбца. Для неимпортируемых полей установить соответствующий указатель. Импортировать столбцы **ТабНомПреп** (числовое, формат – длинное целое), **Фамилия** (текстовое), **ПедСтаж** (числовое, формат – длинное целое).

6. В следующем окне выбрать пункт **Определить ключ** и выбрать поле **ТабНомПреп** в качестве ключевого.

7. В следующем окне задать имя создаваемой таблицы в Access **Преподаватели**. Нажать **Готово**.

Задание 6. Добавить в таблицу **Преподаватели** поле **Должность**, определив ему тип Подстановка.

Выполняемые действия

а) Открыть таблицу **Преподаватели** в режиме **КОНСТРУКТОРА**.

Для поля **Фамилия** установить размер 25.

б) Ввести в список полей имя нового поля **Должность**.

в) В списке типов данных для поля **Должность** выбрать **Мастер подстановок**.

г) В открывшемся окне СОЗДАНИЕ ПОДСТАНОВКИ выбрать пункт: *Объект «столбец подстановки» будет использовать значение из таблицы или запроса*. Нажать *Далее*.

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

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

ж) В следующем окне указать поле, по которому следует выполнить сортировку (**КодДолжности**).

з) В следующем окне отключить функцию *Скрыть ключевой столбец*. Нажать *Далее*. Если эту функцию не отключить, то в поле **Должность** таблицы **Преподаватели** будет видно только наименование должности, а не ее код.

и) В следующем окне в списке *доступные поля* выбрать **КодДолжности**.

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

На вопрос *Перед созданием связи следует сохранить эту таблицу, сделать это сейчас?* Ответить *Да*.

Структура таблицы **Преподаватели** приведена на рисунке 4.11.

л) Перейти в режим таблицы и ввести значения поля **Должность**, выбирая их из списка. Сохранить таблицу.

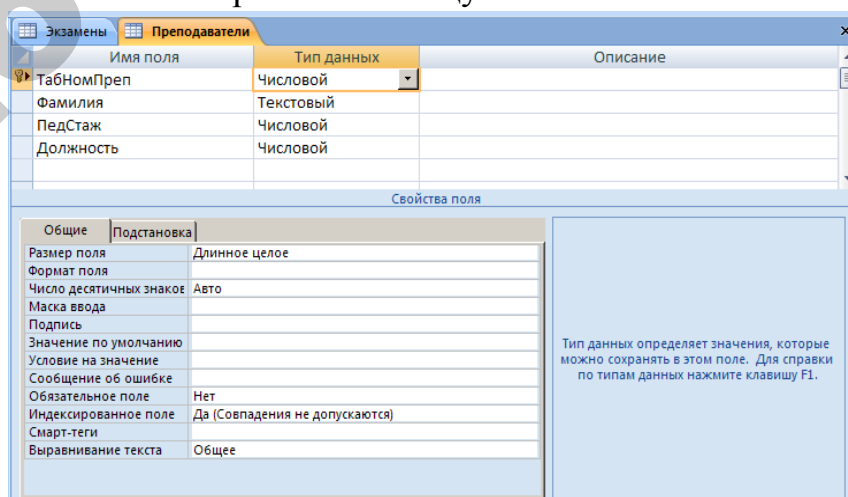


Рис. 4.11. Таблица **Преподаватели** в режиме **КОНСТРУКТОРА**

Задание 7. Создать структуру таблицы **Экзамены** с помощью **КОНСТРУКТОРА**.

Выполняемые действия

1. На вкладке **Создание** ленты меню выбрать **КОНСТРУКТОР таблиц**.
2. Описать имена полей и их типы данных таблицы **Экзамены**, определив полям **Группа**, **Дисциплина**, **Преподаватель** тип *числовой*, поскольку эти объекты будут представлены в конструируемой таблице своими кодами.


Для полей **Дата**, **ВремяНач**, **ВремяКон** определить тип *Дата/Время*.

Структура конструируемой таблицы с измененными значениями свойств полей приведена в таблица 4.18.

Таблица 4.18

Структура таблицы **Экзамены**

Имя поля	Тип данных	Изменяемые свойства поля	Устанавливаемые значения свойств
Группа	Числовой	Обязательное поле	Да
Дисциплина	Числовой	Обязательное поле	Да
Преподаватель	Числовой	Обязательное поле	Да
Дата	Дата/Время	Формат поля	Краткий формат даты
		Маска ввода	99.99.9999
ВремяНач	Дата/Время	Формат поля	Краткий формат времени
		Маска ввода	99:99
ВремяКон	Дата/Время	Формат поля	Краткий формат времени
		Маска ввода	99:99
Аудитория	Текстовый	Размер поля	5

3. Сохранить структуру таблицы, выбрав **Office / Сохранить** или с помощью кнопки . Ввести имя таблицы **Экзамены**.


Замечание. В создаваемой таблице **Экзамены** 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 8. Установить связи между таблицами **Группы-Экзамены**; **Дисциплины-Экзамены**; **Преподаватели-Экзамены**.

Выполняемые действия

- а) Закрывать все таблицы.
- б) На вкладке **РАБОТА С БАЗАМИ ДАННЫХ** ленты меню нажать кнопку . Откроется окно **СХЕМА ДАННЫХ**.

в) Добавить в это окно связываемые таблицы (**Группы, Дисциплины, Преподаватели**). Для этого, нажав кнопку  , открыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ (если оно не открыто), и поочередно дважды щелкнуть клавишей мыши по именам нужных таблиц (или выделить все добавляемые таблицы, удерживая кнопку клавиатуры **Ctrl**, и нажать кнопку **Добавить**). Макеты добавленных таблиц появятся в окне СХЕМА ДАННЫХ. Закройте окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.

Для удаления макета таблицы из окна СХЕМА ДАННЫХ следует щелкнуть клавишей мыши по границе макета и нажать клавишу DEL.

г) Для связывания таблиц **Группы** и **Экзамены** установить курсор в поле **КодГруппы** таблицы **Группы** и, удерживая его левой кнопкой мыши, переместить на поле **Группа** таблицы **Экзамены**.

д) В появившемся окне ИЗМЕНЕНИЕ СВЯЗЕЙ можно изменить некоторые свойства связи, уточнить связь. Обратите внимание, что тип создаваемой связи **один – ко – многим**, поскольку каждая группа сдает много экзаменов. Далее следует 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 **б8им** из таблицы **Группы**, все записи по группе **б8им** автоматически изменяются или удаляются и из таблицы **Экзамены**.

е) Создать связь нажатием кнопки **Создать**.

ж) Аналогично создать еще две связи, переместив поле **КодДисциплины** таблицы **Дисциплины** на поле **Дисциплина** таблицы **Экзамены** и поле **ТабНомПреподавателя** таблицы **Преподаватели** на поле **Преподаватель** таблицы **Экзамены**. Установить свойства **Обеспечение целостности данных, Каскадное обновление записей и Каскадное удаление записей** для всех связей.

з) Добавить в окно СХЕМА ДАННЫХ таблицу **Должности**. Обратите внимание, что связь между таблицами **Должности** и **Преподаватели** установилась автоматически, в результате определения поля **Должность** таблицы **Преподаватели** типа ПОДСТАНОВКА. Установите свойства **Обеспечение целостности данных, Каскадное обновление записей и Каскадное удаление записей** для этой связи.

Общий вид окна схемы данных изображен на рисунке 4.12.

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

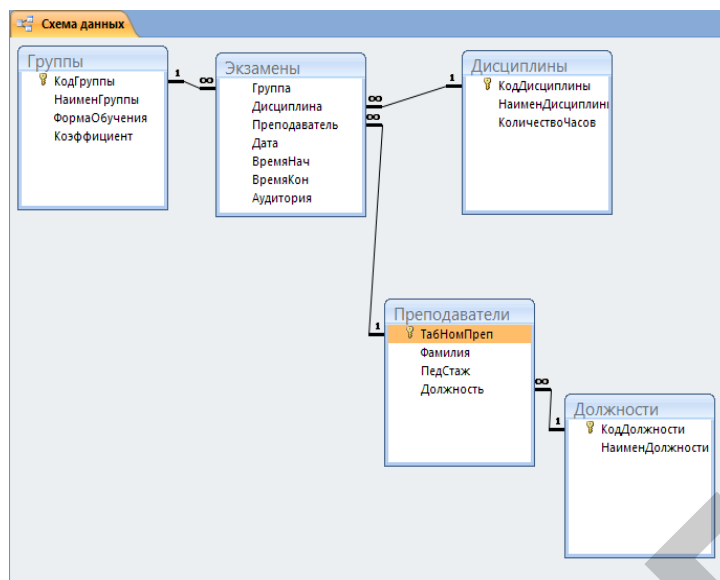



Рис. 4.12. Окно СХЕМА ДАННЫХ

Для вывода на экран всех существующих в БД связей между таблицами следует в окне СХЕМА ДАННЫХ (рис. 4.12) нажать кнопку  . .

Замечание. Кроме описанных выше приемов в окне СХЕМА ДАННЫХ пользователь имеет возможность выполнить следующие действия.

1. Изменить макет таблицы:
 - установить указатель на таблицу, нажать правую клавишу мыши и выбрать **Конструктор таблиц**;
 - внести изменения в макет;
 - 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.

Индивидуальные задания

Выберите по указанию преподавателя вариант индивидуального задания. Разработайте базу данных, выполните задания в соответствии с выбранным уровнем сложности.

Репродуктивный уровень (оценка 4–6)

Для вариантов 13–15 выполнить задания 1–3 из темы **Создание таблиц**.

Продуктивный уровень (оценка 7–9)

Для вариантов 1–12 выполнить задания 1–3 из темы **Создание таблиц**.

Творческий уровень (оценка 10)

Создать базу данных для решения реальной задачи из любой области знаний. Самостоятельно спроектировать базу данных, построить инфологическую и даталогическую модели базы данных. Создать справочники, описать структуры связующих таблиц, определить связи между таблицами.

ЛАБОРАТОРНАЯ РАБОТА № 2

СОЗДАНИЕ ПРОСТЫХ ФОРМ.


ЗАПОЛНЕНИЕ ТАБЛИЦ С ИСПОЛЬЗОВАНИЕМ ФОРМ

Цель работы: научиться создавать простые формы и использовать их для заполнения таблиц данными. На форме уметь применять удобный для заполнения таблиц данными элемент управления поле со списком, уметь создавать на форме кнопку, при нажатии на которую будет открываться другая форма.


Задание 1. Создать форму **ФДолжности** для пополнения таблицы **Должности** новыми данными.

Выполняемые действия

1. В области переходов выделить таблицу **Должности**.

2. Нажать кнопку **Форма**  на вкладке **Создание** ленты меню. Откроется окно макета формы, содержащее все поля таблицы **Должности** с надписями. Поля размещены в столбик и сгруппированы. Обратите внимание, что в нижней части основной формы отражена ленточная подчиненная форма по таблице **Преподаватели**, так как в связи таблиц **Должности** – **Преподаватели** (Один – к – многим) таблица **Преподаватели** считается подчиненной. С помощью создаваемой формы заполнять можно одновременно обе таблицы.

3. Сохранить форму с именем **ФДолжности**.

Замечание 1. Если нужно дополнить форму новыми элементами, то следует перейти в режим КОНСТРУКТОРА с помощью кнопки **Режим** .

Замечание 2. Для разгруппировки полей и надписей в режиме КОНСТРУКТОРА следует выделить все поля и надписи (весь макет), вызвать правой кнопкой мыши контекстное меню и выбрать команды **Макет – Удалить**.

4. Для ввода данных перейти в режим формы с помощью той же кнопки **Режим**.

Задание 2. С помощью формы **ФДолжности** дополнить таблицу **Должности** новой строкой (таблица 4.19).

Данные для добавления в таблицу **Должности**


<i>КодДолжности</i>	<i>НаименДолжности</i>
6	<i>Преподаватель-стажер</i>

Ввести в таблицу **Преподаватели** двух сотрудников, работающих в этой должности (таблица 4.20).

Таблица 4.20


Данные для добавления в таблицу **Преподаватели**



<i>ТабНомПреп</i>	<i>Фамилия</i>	<i>ПедСтаж</i>
6	Крот	3
7	Орда	5

Замечание. Для ввода новой записи с помощью формы следует нажать кнопку  в строке состояния формы (самая нижняя строка).

Задание 3. Создать форму **ФЭкзамены** для заполнения таблицы **Экзамены**.

Выполняемые действия

1. При выделенной таблице **Экзамены** в области переходов нажать кнопку **Конструктор форм**  на вкладке **Создание** ленты меню. Откроется окно конструктора форм, содержащее пустую область данных формы. Область данных служит для размещения полей таблицы.

2. Далее нажать кнопку **Добавить поля** . Если в области **Список полей** не показан список объектов, нажать кнопку  **Показать все таблицы**. Затем, выбрав в открывшемся списке таблиц, перенести в область данных формы поля таблицы **Экзамены** двойным щелчком клавишей мыши по каждому полю. В области данных формы появятся поля (правый столбец) вместе с присоединенными надписями (левый столбец).

Так как поля **Группа**, **Дисциплина** и **Преподаватель** выбираются из соответствующих справочников, то для ввода их значений удобно использовать элемент управления, называемый *поле со списком*.

Задание 3.1. Организовать поле **Группа** на создаваемой форме в виде *поля со списком*.

Выполняемые действия

1. Выделить поле **Группа** в конструкторе формы. В вызванном правой кнопкой мыши контекстном меню выбрать пункт *Преобразовать элемент в Поле со списком*.

2. Снова вызвать контекстное меню правой кнопкой мыши и выбрать пункт *Свойства*. Далее на вкладке *Все* выбрать *Число столбцов 2*, *Тип источника строк Таблица/запрос* (по умолчанию). В свойстве *Источник строк* из раскрывающегося списка выбрать таблицу **Группы**. Закрыть окно свойств.


Задание 3.2. Организовать поля **Дисциплина** и **Преподаватель** в виде поля со списком.

Для наглядности нанесите на форму еще поля **НаимГруппы**, **НаименованиеДисциплины**, **Фамилия** из соответствующих справочников. Сохранить форму с именем **ФЭкзамены**.

Задание 3.3. Добавить в форму области **ЗАГОЛОВОК ФОРМЫ** и **ПРИМЕЧАНИЕ ФОРМЫ**.

Выполняемые действия

1. Нажать кнопку .

2. В заголовок ввести надпись *Заполнение таблицы Экзамены* (перемещением кнопки  с панели элементов, если поле надписи не появилось автоматически).

Открыть окно свойств надписи (с помощью правой кнопки мыши при выделенной надписи). На вкладке *Все* задать следующие значения свойств:

Цвет фона – **Светлый фон заголовка** (по нажатию 

Размер шрифта – **16**;

Выравнивание текста – **По центру**;

Цвет текста – **Темный текст**.

3. В примечание формы добавить надпись *«Поля Группа, Дисциплина, Преподаватель выбираются из списков»*. Для выделения курсивом можно пользоваться соответствующей кнопкой группы *Шрифт* вкладки *Конструктор* ленты меню.

Создаваемая форма приведена на рисунке 4.13 (режим **КОНСТРУКТОРА**) и на рисунке 4.14 (режим **ФОРМЫ**).

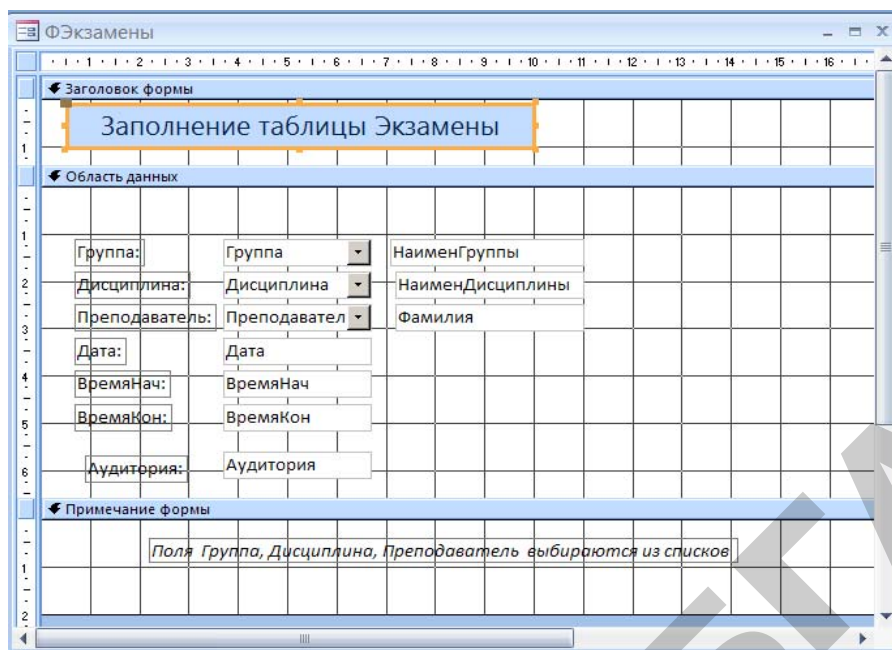


Рис. 4.13. Форма ФЭкзамены в режиме КОНСТРУКТОРА



Замечание. Переход между режимами просмотра объекта осуществляется кнопкой на вкладке **Главная** ленты меню.

Задание 4. Заполнить таблицу Экзамены с помощью созданной формы, выбирая значения полей **Группа**, **Дисциплина** и **Преподаватель** из раскрывающихся списков. Данные таблицы Экзамены представлены в таблице 4.4.

Задание 5. Создать таблицу Студенты, содержащую основные сведения о студентах в соответствии со структурой, приведенной в таблице 4.21.

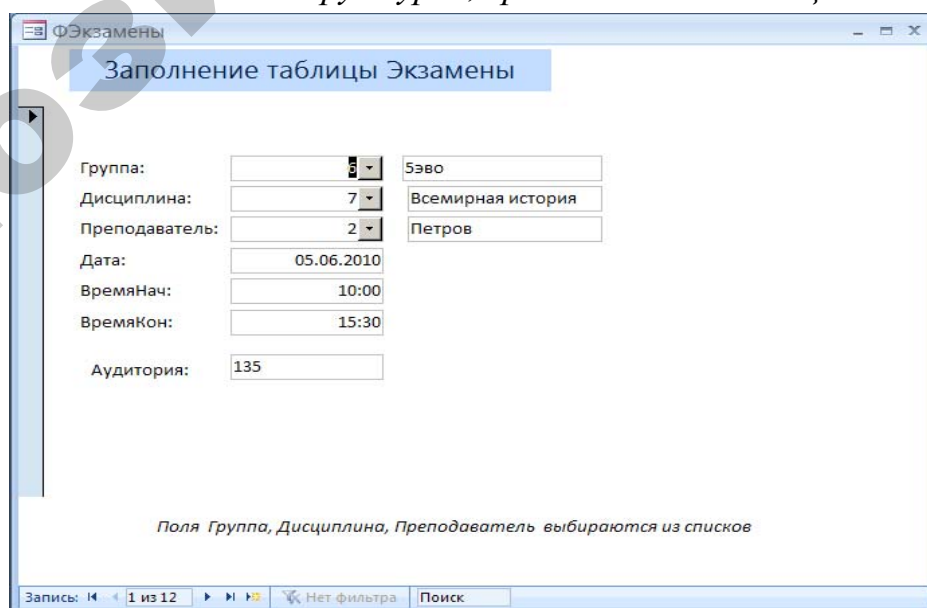


Рис. 4.14. Форма ФЭкзамены в режиме ФОРМЫ

Задание 6. Создайте две формы для заполнения таблицы *Студенты*. Первая форма *ФСтудДеканат* предназначена для ввода данных, обрабатываемых в деканате, и включает поля *НомерЗачКн*, *Фамилия*, *Имя*, *Отчество*, *Группа*, *БюджетОбуч*, *Общежитие*. Так как поле *Группа* заполняется значениями из таблицы *Группы*, организовать его как поле со списком.

Таблица 4.21

Структура таблицы *Студенты*

Имя поля	Тип данных	Изменяемые свойства поля	Устанавливаемые значения свойств
<i>НомерЗачКн</i>	Числовой	Индексированное поле	Да (совпадения не допускаются)
<i>Фамилия</i>	Текстовый	Размер поля	20
<i>Имя</i>	Текстовый	Размер поля	15
<i>Отчество</i>	Текстовый	Размер поля	20
<i>Группа</i>	Числовой	Обязательное поле	Да
<i>БюджетОбуч</i>	Логический		
<i>Общежитие</i>	Логический		
<i>ДатаРожд</i>	Дата/Время	Формат поля	Краткий формат даты
<i>СемПоложение</i>	Логический		
<i>Телефон</i>	Текстовый	Размер поля	10

Вторая форма *ФСтудОК* предназначена для заполнения таблицы *Студенты* данными студенческого отдела кадров и должна содержать поля: *НомерЗачКн* (для соответствия), *ДатаРожд*, *СемПоложение*, *Телефон*.

Указанные формы удобнее создавать в режиме КОНСТРУКТОРА.

Задание 7. Заполните таблицу *Студенты* (таблица 4.9), используя созданные формы.

Указание. Вначале введите данные деканата с помощью формы *ФСтудДеканат*, причем поле *Группа* следует выбирать из раскрывающегося списка. Затем, просматривая уже созданные записи, с помощью фор-

мы **ФСтудОК**, дополните их данными студенческого отдела кадров в соответствии с номерами зачетных книжек.

Задание 8. Для представления информации (данных) о результатах экзаменов создать таблицу **Оценки**, структура которой приведена в таблице 4.22.

Таблица 4.22


Структура таблицы **Оценки**

Имя поля	Тип данных	Изменяемые свойства поля	Устанавливаемые значения свойств
Дисциплина	Мастер подстановок	Обязательное поле	Да
Студент	Мастер подстановок	Обязательное поле	Да
Оценка	Числовой	Условие на значение	≥ 0 And ≤ 10
		Сообщение об ошибке	Значение не является оценкой

Замечание. При определении типа *Подстановка* поля *Дисциплина* функцию *Скрыть ключевой столбец* не отключать!!!

Задание 9. Создать форму **ФОценки** для заполнения таблицы **Оценки**.

Выполняемые действия

1. В области переходов выделить таблицу **Оценки**.
2. Нажать кнопку **Разделенная Форма**  на вкладке **Создание** ленты меню. Откроется окно макета формы, содержащее все поля таблицы с надписями.
3. Для ввода данных перейти в режим **ФОРМЫ**. Заполните таблицу **Оценки** данными, приведенными на рисунке 4.15. Поля *Дисциплина* и *Студент* выбирать из раскрывающихся списков.

Оценки		
Дисциплина	Студент	Оценка
КИТ	92001	9
Высшая математика	92001	8
ЭММ	92001	9
Макроэкономика	92001	10
История Беларуси	92001	7
КИТ	92002	7
Высшая математика	92002	6
ЭММ	92002	8
Макроэкономика	92002	9
История Беларуси	92002	9
КИТ	92005	9
Высшая математика	92005	9
ЭММ	92005	9
Макроэкономика	92005	10
История Беларуси	92005	9
ИТ	94026	9
Высшая математика	94026	8
ЭММ	94026	8
Макроэкономика	94026	8
Всемирная история	94026	7
ИТ	94041	8
Высшая математика	94041	7
ЭММ	94041	6
Макроэкономика	94041	7
Всемирная история	94041	7
ИТ	94048	5
Высшая математика	94048	4
ЭММ	94048	6
Макроэкономика	94048	6
Всемирная история	94048	5



Рис. 4.15. Таблица **Оценки**

Создание кнопки в форме

Задание 10. В форме **ФОценки** создать кнопку для открытия формы **ФЭкзамены**.

Выполняемые действия

1. Перейти в режим **КОНСТРУКТОРА** формы **ФОценки** с помощью кнопки **Режим**.

2. При нажатой кнопке  [МАСТЕР] переместить знак  с панели элементов в область размещения кнопки в форме. Запустится мастер СОЗДАНИЕ КНОПОК.

Далее выполнить следующие действия.

а) В первом окне мастера выбрать:

в области КАТЕГОРИИ – категорию РАБОТА С ФОРМОЙ;

в области ДЕЙСТВИЯ – действие ОТКРЫТИЕ ФОРМЫ, нажать ДАЛЕЕ.

б) Во втором окне мастера выбрать форму **ФЭкзамены**, которая должна открываться при нажатии создаваемой кнопки. Нажать ДАЛЕЕ.

в) В третьем окне оставить положение переключателя ОТКРЫТЬ ФОРМУ И ПОКАЗАТЬ ВСЕ ЗАПИСИ. Нажать ДАЛЕЕ.

г) В четвертом окне поставить переключатель в положение ТЕКСТ, в поле ввода, расположенное справа, набрать текст, помещаемый на кнопку: “Открыть форму **ФЭкзамены**”. Нажать ДАЛЕЕ.


д) В пятом окне задать имя кнопки или оставить предложенное системой. Нажать ГОТОВО. Сохранить новый вариант формы.

3. Перейти в режим ФОРМЫ, убедиться, что кнопка выполняет свою функцию.

Задание 11. Создать связь между таблицами Группы – Студенты.

Выполняемые действия

1. Открыть окно СХЕМА ДАННЫХ, нажав на соответствующую кнопку вкладки Работа с базами данных ленты меню.

2. Добавить таблицу **Студенты**, нажав на кнопку  и дважды щелкнув клавишей мыши по имени Студенты в появившемся списке таблиц.

3. Создать связь между таблицами **Группы – Студенты**, установив свойства *Обеспечение целостности данных*, *Каскадное обновление записей* и *Каскадное удаление записей* для этой связи.

4. Добавить в окно СХЕМА ДАННЫХ таблицу **Оценки**. Обратите внимание, что связи таблиц **Студенты – Оценки**, **Дисциплины – Оценки** установились автоматически в результате определения полям *Дисциплина* и *Студент* таблицы **Оценки** типа *Подстановка*. Установите этим связям свойства *Обеспечение целостности данных*, *Каскадное обновление записей* и *Каскадное удаление записей*.

В результате окно СХЕМА ДАННЫХ примет вид, приведенный на рисунке 4.16.

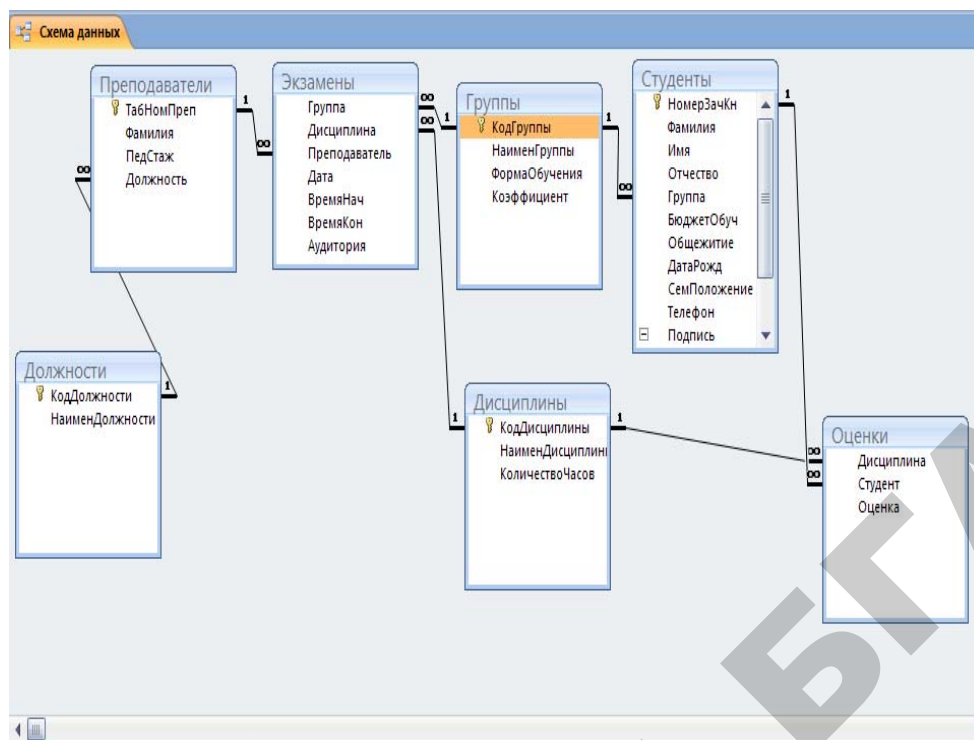


Рис. 4.16. Окно «Схема данных»

Задание 12. Создать таблицу **Стипендия**, содержащую информацию о видах и размерах стипендии в зависимости от успеваемости студента в соответствии со структурой, приведенной в таблице 4.23.

Таблица 4.23

Структура таблицы **Стипендия**

Имя поля	Тип данных	Изменяемые свойства поля	Устанавливаемые значения свойств
СрБаллМин	Числовой	Размер поля	Одинарное с плавающей точкой
		Формат поля	Основной
		Число десятичных знаков	1
		Условие на значение	≥ 0 And ≤ 10
		Сообщение об ошибке	Значение не является оценкой
СрБаллМакс	Числовой	Размер поля	Одинарное с плавающей точкой
		Формат поля	Основной
		Число десятичных знаков	1
		Условие на значение	≥ 0 And ≤ 10
		Сообщение об ошибке	Значение не является оценкой
Вид	Текстовый	Размер поля	20
Размер	Числовой	Формат поля	Денежный
		Число десятичных знаков	0

Таблица **Стипендия** изображена на рисунке 4.17.

СрБаллМин	СрБаллМакс	ВидСтип	РазмерСтип
0	5.5	Социальная	82 503р.
5.5	6	Учебная	102 539р.
6	8	Учебная+10%	112 793р.
8	9	Учебная+20%	133 301р.
9	10	Учебная+50%	153 809р.
*			0р.

Рис. 4.17. Таблица Стипендия

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

Для вариантов 13–15 выполнить задания из темы **Создание таблиц**, начиная с номера 4 и заканчивая последним номером, включая его.

Продуктивный уровень (оценка 7–9)

Для вариантов 1–12 выполнить задания из темы **Создание таблиц**, начиная с номера 4 и заканчивая последним номером, включая его.

Творческий уровень (оценка 10)

В индивидуальной базе данных создать формы для заполнения связующих таблиц, на формах предусмотреть поля со списками. Описать и заполнить все спроектированные таблицы базы данных.

ЛАБОРАТОРНАЯ РАБОТА № 3

РЕДАКТИРОВАНИЕ БАЗЫ ДАННЫХ. ФИЛЬТРАЦИЯ ДАННЫХ

Цель работы: научиться редактировать таблицы и схему данных БД, проводить различные виды отбора данных путем их фильтрации.

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



Чтобы **изменить ширину столбца**, нужно передвинуть с помощью мыши границу столбца. **Высота строки** меняется аналогично перемещением граничной линии.

Чтобы **поменять местами столбцы**, следует выделить перемещаемый столбец и с помощью мыши передвинуть его в то место, где он должен находиться.

Чтобы **скрыть столбец** на экране, следует его правую границу переместить на левую.

Изменение структуры таблицы

Структура таблицы изменяется в режиме КОНСТРУКТОРА. Поля таблицы можно перемещать, удалять, вставлять, добавлять; кроме того, можно изменять их имена, типы, свойства. **Чтобы поменять местами поля в структуре таблицы**, следует перемещаемое поле выделить и передвинуть в нужное место. 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 1. *Открыть базу данных ВУЗ. Сделать копию таблицы Экзамены с именем Экзамены1. Использовать кнопки  (копировать) и  (вставить).*

Выполнить изменения размеров строк и столбцов таблицы Экзамены1 и изменить структуру таблицы Экзамены1: поменять местами поля Группа и Дисциплина, изменить формат поля ВремяНач.

Сортировка и поиск данных в таблице

Задание 2. Таблицу **Преподаватели** отсортировать в порядке убывания поля **ПедСтаж**.


Выполняемые действия

1. Открыть таблицу **Преподаватели** в режиме ТАБЛИЦЫ.
2. Выделить столбец **ПедСтаж** и нажать кнопку . Записи будут отсортированы по возрастанию. Если теперь сохранить таблицу, то она сохранится с новой сортировкой. Для сортировки по убыванию следует нажать кнопку .

Поиск в таблице

Задание 3. Выполнить следующие виды поиска в таблице **Экзамены1**:

а) найти запись с конкретным номером, например, № 7 в таблице. Для этого следует в режиме таблицы в поле **Запись** ввести 7 и нажать клавишу ENTER;

б) найти запись с конкретным значением поля, например, в таблице **Экзамены1** найти записи со значением 3 поля **Дисциплина**. Для этого следует в таблице **Экзамены1** курсор поставить в поле **Дисциплина**; нажать кнопку  (найти) в группе **Найти** вкладки **Главная** ленты меню; ввести искомое значение 3 в поле **Образец**; определить другие необходимые параметры в окне диалога **ПОИСК**:

поиск в поле **Дисциплина**, совпадение – **Поля целиком**, просмотр – **Вниз**.

Нажать **Найти далее**. Найдется первое вхождение данного значения поля. Для поиска следующего вхождения нажать **Найти далее**.

Для удаления таблицы из БД нужно выделить ее имя в окне переходов и нажать клавишу DELETE.

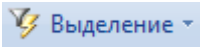

Использование фильтров в таблицах

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.


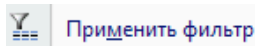

Задание 5. *Используя фильтр по выделенному, из таблицы **Экзамены1** выбрать сведения о группе с кодом 1.*

Выполняемые действия

- а) Открыть таблицу **Экзамены1** в режиме таблицы.
- б) Поместить курсор в колонку **Группа** на значение «1».
- в) Выбрать  в группе *Сортировка и фильтр* вкладки *Главная* ленты меню. Далее выбрать *Равно 1*. На экране отразится таблица в заданном виде.
- г) Удалить фильтр кнопкой  пункта *Дополнительно* в группе *Сортировка и фильтр* вкладки *Главная* ленты меню.

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




Выполняемые действия

1. Открыть таблицу **Группы** в режиме таблицы.
2. Нажать кнопку  пункта *Дополнительно* в группе *Сортировка и фильтр* вкладки *Главная* ленты меню. Появится окно с заголовками полей таблицы.
3. Поставить курсор на поле **ФормаОбучения** и из списка выбрать значение **Дневная**. Затем нажать вкладку *или* в нижней части окна и в том же поле **ФормаОбучения** выбрать значение **Заочная**.
4. Установить курсор в таблице, вызвать правой кнопкой мыши контекстное меню и выбрать пункт .
5. Проанализировать результаты фильтрации.
6. Удалить фильтр кнопкой  пункта *Дополнительно* в группе *Сортировка и фильтр* вкладки *Главная* ленты меню.

Задание 7. *Из таблицы **Преподаватели** выбрать сведения о профессорах, педагогический стаж которых составляет 15 и более лет. Вывести записи в алфавитном порядке фамилий.*

Эту задачу можно решить, используя расширенный фильтр.

Выполняемые действия

1. Открыть таблицу **Преподаватели** в режиме таблицы.
2. Нажать кнопку  **Расширенный фильтр...** пункта *Дополнительно* в группе *Сортировка и фильтр* вкладки *Главная* ленты меню. Появится окно конструктора фильтра.
3. Далее в бланк фильтра в строку **Поле** из списка полей таблицы переместить поля **Фамилия, ПедСтаж, Должность**.
4. Задать сортировку записей в алфавитном порядке фамилий. Для этого щелкнуть клавишей мыши в клетке на пересечении колонки **ФИО** и строки *Сортировка* и из раскрывающегося списка выбрать строку *по возрастанию*.
5. Задать условия отбора записей. Для этого в строку *Условие отбора* в колонку **ПедСтаж** ввести условие ≥ 15 , а в колонку **Должность** ввести **1**.
7. Применить фильтр, установив курсор в таблице, вызвав правой кнопкой мыши контекстное меню и выбрав пункт  **Применить фильтр**.
8. Проанализировать результаты фильтрации.
9. Удалить фильтр кнопкой  **Очистить все фильтры** пункта *Дополнительно* в группе *Сортировка и фильтр* вкладки *Главная* ленты меню.

Задание 8. *Откройте таблицу **Группы** вместе с подчиненными таблицами.*

Выполняемые действия

Обратите внимание на то, что в начале каждой строки размещены значки открытия подчиненных таблиц (две таблицы: **Студенты** и **Экзамены**). При нажатии на значок выводится список таблиц. Следует выбрать одну из подчиненных таблиц, например, **Студенты**. Тогда выведется информация о студентах текущей группы. В каждой строке данных о студентах также содержатся значки раскрытия (+), при нажатии на которые выводятся все оценки данного студента. Таблица в этом случае имеет вид, изображенный на рисунке 4.17. Вводить данные можно одновременно в три таблицы.

Задание 9. *Используя подчиненные таблицы, ввести информацию о новой группе (таблица 1.24), информацию о двух студентах новой группы (таблица 1.25) и для каждого студента информацию о его оценках по двум дисциплинам (таблица 1.26). Отдельной строкой добавить в таблицу **Студенты** сведения о себе.*

Рис. 4.17. Таблица Группы с подчиненными таблицами

Таблица 4.24

Информация о новой группе

КодГруппы	НаименГруппы	ФормаОбучения	Коэффициент
6	5Эво	Заочная	0.3

Таблица 4.25

Информация о студентах новой группы

НомерЗачКн	Фамилия	Имя	Отчество	БюджетОбуч	Общежитие	ДатаРожд	СемПоложен ие	Телефон	Подпись
86020	Грекова	Ирина	Леонидовна	да	нет	12.08.1993	нет	226-15-89	
86015	Никитин	Сергей	Иванович	да	да	10.04.1994	нет	255-25-80	

Таблица 4.26

Информация об оценках новых студентов

Дисциплина	Студент	Оценка
Высшая математика	86020	8
Всемирная история	86020	9
Высшая математика	86015	6
Всемирная история	86015	5

Задание 10. Используя форму **ФЭкзамены**, добавить в таблицу **Экзамены** сведения о сдаче экзаменов по Высшей математике и Всемирной истории группой с кодом 6 (две записи). Дату проведения, время и аудиторию ввести по своему усмотрению.

Задание 11. *Добавить в таблицу Студенты новое поле ЛичныеДанные, определив для него тип Вложение. Вложить в поле фото студента и роспись (выполнить только для записи о себе).*

Выполняемые действия

1. Открыть таблицу **Студенты** в режиме конструктора.
2. Добавить в конец структуры новое поле **ЛичныеДанные**, выбрав для него тип **Вложение**.

Замечание. Следует помнить, что нельзя преобразовать поле типа **Вложение** в поле другого типа данных, однако в случае ошибки это поле можно удалить.

3. Перейти в режим таблицы **Студенты**.
4. Дважды щелкните по полю **ЛичныеДанные** (вложение).
5. В открывшемся окне **Вложения** нажмите кнопку **Добавить**.
6. Откроется диалоговое окно **Выберите файл**, в котором следует выбрать файл вложения.
7. Нажать **ОК**. Файл будет добавлен в поле, а число, указывающее количество вложений, будет соответственно увеличено.
8. По аналогии (п. 4–7) вложить в поле **ЛичныеДанные** файл росписи, который можно предварительно создать в графическом редакторе **Paint**.

**Задания для самостоятельного выполнения
к лабораторной работе № 3**

1. Откорректировать макет таблицы **Экзамены1**, расположив поле **Дата** последним.
2. Рассортировать записи таблицы **Экзамены1** в порядке убывания поля **Дата**.
3. Осуществить поиск в таблице **Экзамены1** по значению **2** поля **Группа**. Просмотреть все вхождения значения.
4. Удалить таблицу **Экзамены1**.
5. Осуществить корректировку БД следующим образом:
 - а) Добавить в таблицу **Преподаватели** запись.

<i>ТабНомПреп</i>	<i>Фамилия</i>	<i>ПедСтаж</i>	<i>Должность</i>
10	Сидоров	18	3

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

б) Изменить в таблице **Преподаватели** табельный номер Сидорова на 11. Закрывать таблицу. Убедиться, что в записях таблицы **Экзамены**, касающихся Сидорова, произошла аналогичная замена (поле **ТабНомПреп** приняло значение 11). Осуществилось каскадное обновление связанного поля, заданное при установке связи между таблицами.

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

6. Закрывать базу данных ВУЗ.

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

Для выбранного варианта выполнить редактирование таблиц, предварительно сделав копии редактируемых таблиц. Поменять размеры, форматы полей, изменить сортировку таблиц, произвести выборки из таблиц с помощью фильтров.

Продуктивный уровень (оценка 7–9)

Для выбранного варианта выполнить редактирование таблиц, предварительно сделав копии редактируемых таблиц. Поменять размеры, форматы полей, изменить сортировку таблиц, произвести выборки из таблиц с помощью фильтров. Проверить (и продемонстрировать) свойства связей *Каскадное обновление и удаление записей*.

Творческий уровень (оценка 10)

В индивидуальной базе данных выполнить редактирование таблиц, предварительно сделав копии редактируемых таблиц. Поменять размеры, форматы полей, изменить сортировку таблиц, произвести выборки из таблиц с помощью фильтров. Проверить (и продемонстрировать) свойства связей *Каскадное обновление и удаление записей*.

ЛАБОРАТОРНАЯ РАБОТА № 4

СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОР ДАННЫХ

Цель работы: научиться выполнять выбор данных из таблиц с помощью запросов различных типов (из одной таблицы, из нескольких таблиц с сортировкой, с условием отбора, с параметром, с созданием вычисляемого поля, с группировкой, перекрестного запроса).

Создание запроса на выбор данных из одной таблице

Задание 1. Вывести следующие сведения о студентах: *НомерЗачКн, Фамилия, Имя, Отчество, БюджетОбуч, Общежитие, Телефон*. Запрос сохранить с именем **ЗапСтуденты**.

Нужные сведения содержатся в таблице **Студенты**. Создавать запрос следует по ней.

Выполняемые действия

1. На вкладке **Создание** ленты меню в группе **Другие** нажать кнопку



В появившемся окне **ДОБАВЛЕНИЕ ТАБЛИЦЫ** (если его нет, то нажать) добавить таблицу **Студенты** (двойным щелчком клавишей мыши по имени таблицы).

2. Переместить поля *НомерЗачКн, Фамилия, Имя, Отчество, БюджетОбуч, Общежитие, Телефон* в бланк запроса (нижняя часть окна).
Закреть окно **ДОБАВЛЕНИЕ ТАБЛИЦЫ**.

3. Проверить наличие флажка **Вывод на экран** (V) для полей запроса.

4. Выполнить запрос с помощью кнопки



или кнопки



5. Сохранить запрос с именем **ЗапСтуденты**.

Создание запроса на основе нескольких таблиц с применением сортировки

Задание 2. С помощью **КОНСТРУКТОРА** создать запрос **ЗапЭкзамены**, результирующая таблица которого имела бы структуру записи, подобную структуре записи таблицы **Экзамены**, но объекты должны быть представлены своими наименованиями (взятыми из справочников). Добавить поле **НаименДолжности** из таблицы **Должности**. В результирующую таблицу вывести все записи таблицы **Экзамены** (см. рис. 4.19). Произвести сортировку по следующим полям:

- 1) **НаименГруппы** – убывание;
- 2) **Дата** – возрастание.

Это означает, что сначала записи будут отсортированы по наименованию группы, а затем среди одинаковых значений наименования группы записи должны следовать в порядке возрастания даты.

Выполняемые действия

1. Перейти в режим **КОНСТРУКТОРА** запросов.
2. В появившемся окне **ДОБАВЛЕНИЕ ТАБЛИЦЫ** добавить таблицы **Экзамены**, **Группы**, **Дисциплины**, **Преподаватели**, **Должности** (двойным щелчком клавишей мыши по именам таблиц). Закрыть окно **ДОБАВЛЕНИЕ ТАБЛИЦЫ**.

3. На экране появилось окно конструктора запросов, в верхней части которого видны структуры всех добавленных таблиц с установленными связями. В нижней части содержится пустой бланк создаваемого запроса. В строку **ПОЛЕ** бланка следует переместить поля, включаемые в результирующую таблицу: **НаименГруппы** – из таблицы **Группы**, **НаименДисциплин** – из таблицы **Дисциплины**, **Фамилия** – из таблицы **Преподаватели**, **НаименДолжности** – из таблицы **Должности**, **Дата**, **ВремяНач**, **ВремяКон**, **Аудитория** – из таблицы **Экзамены**.

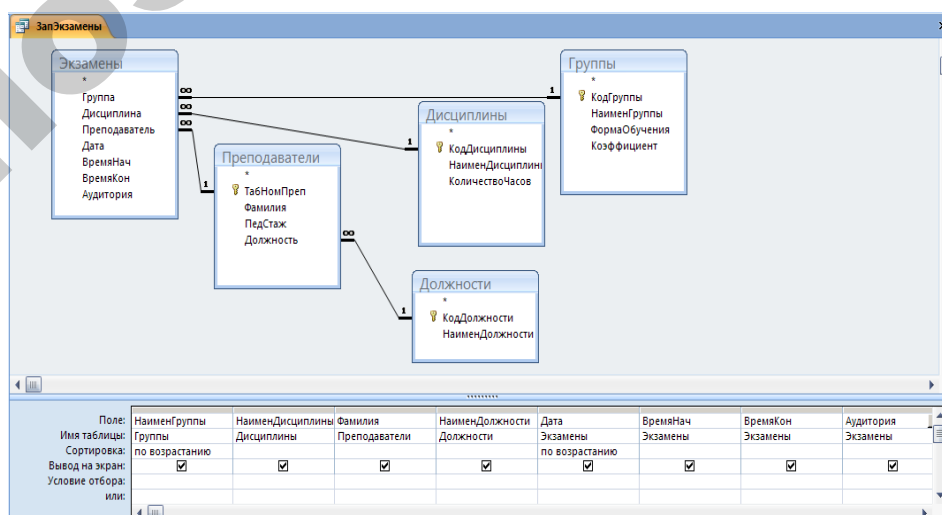


Рис. 4.18. Запрос **ЗапЭкзамены** в режиме **КОНСТРУКТОРА**

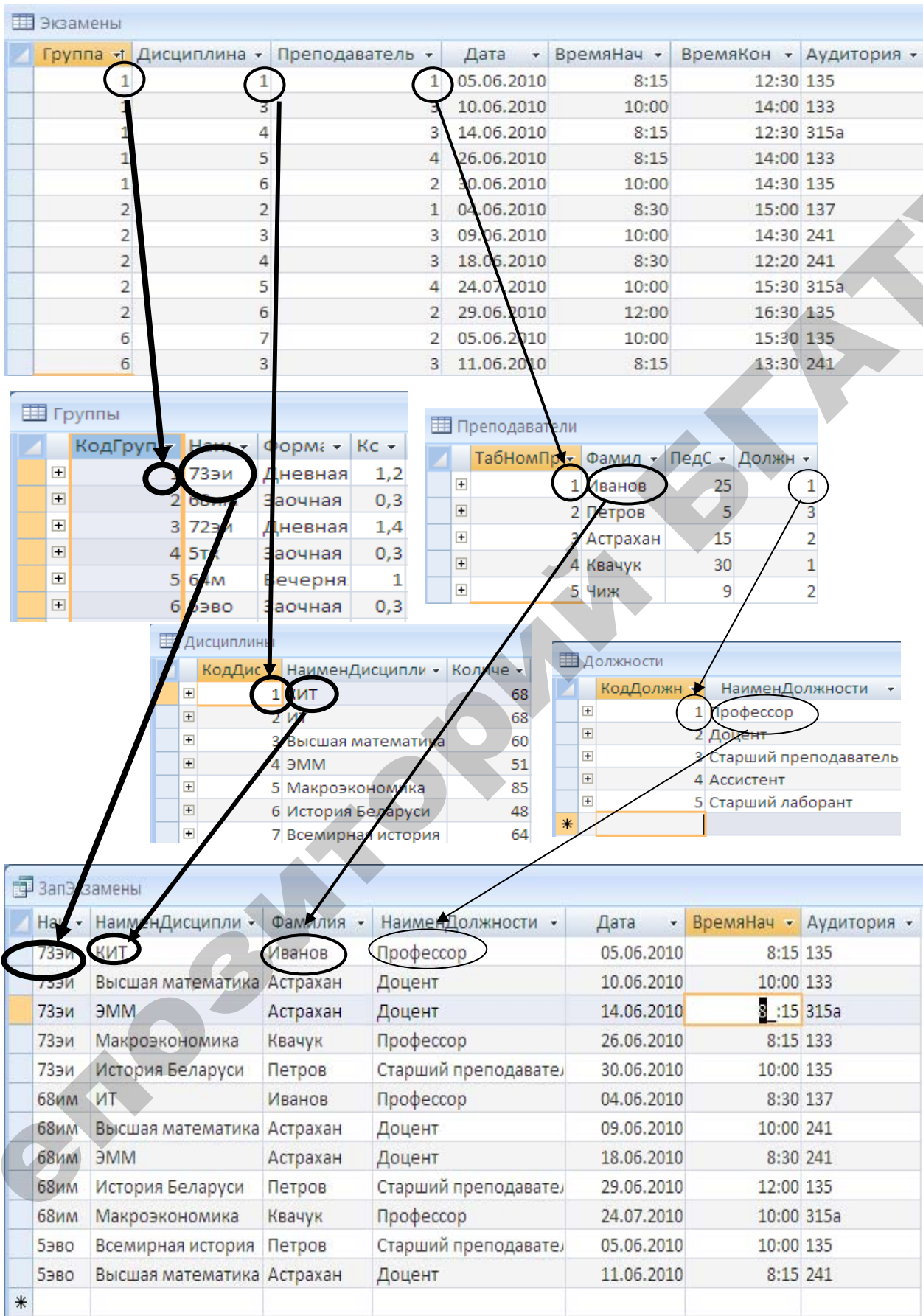


Рис. 4.19 Схема выбора данных по запросу ЗапЭкзамены

4. В строке **Вывод на экран** проверить наличие флажков (V) для всех полей.

5. Сохранить и выполнить запрос.

6. Задание порядка сортировки запроса. В сортировке может участвовать до 10 полей.

- Следует поместить 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 **НаименГруппы** выбрать **По убыванию**, для поля **Дата** – **По возрастанию**. Окончательный вид бланка запроса изображен на рисунке 4.18.

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 собой выражения сравнения. В выражениях могут использоваться логические операторы **NOT**, **AND**, **OR**, а также конструкция **BETWEEN** (Между, см. приложение 4).

Если выражения вводятся в одну строку нескольких столбцов **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 **ЗапЭкзаменыГр**, структура результирующей таблицы которого идентична, **ЗапЭкзамены**, но в таблицу включить только экзамены в группе **73эи**.

Выполняемые действия

1. Скопировать **ЗапЭкзамены** с именем **ЗапЭкзаменыГр** (последовательно нажав кнопки **Копировать** и **Вставить**).

2. Открыть **ЗапЭкзаменыГр** в режиме **КОНСТРУКТОРА** и в строку **Условие отбора** поля **НаименГруппы** ввести надпись **73эи**. Макет **ЗапЭкзаменыГр** изображен на рисунке 1.20.

Задание 4. Создать запрос **ЗапЭкзаменыГр1**, выбирающий экзамены группы **73эи**, назначенные на период с 9-го по 15-е июня.

Выполняемые действия

1. Создать запрос, аналогичный **ЗапЭкзаменыГр** (можно скопировать), но в строку **Условие отбора** конструктора в поле **Дата** добавить условие **>=9.06.2010 AND <=15.06.2010**. Это же условие можно записать так: **BETWEEN 9.06.2010 AND 15.06.2010**.
2. Сохранить запрос с именем **ЗапЭкзаменыГр1**.
3. Сохранить и выполнить запрос.

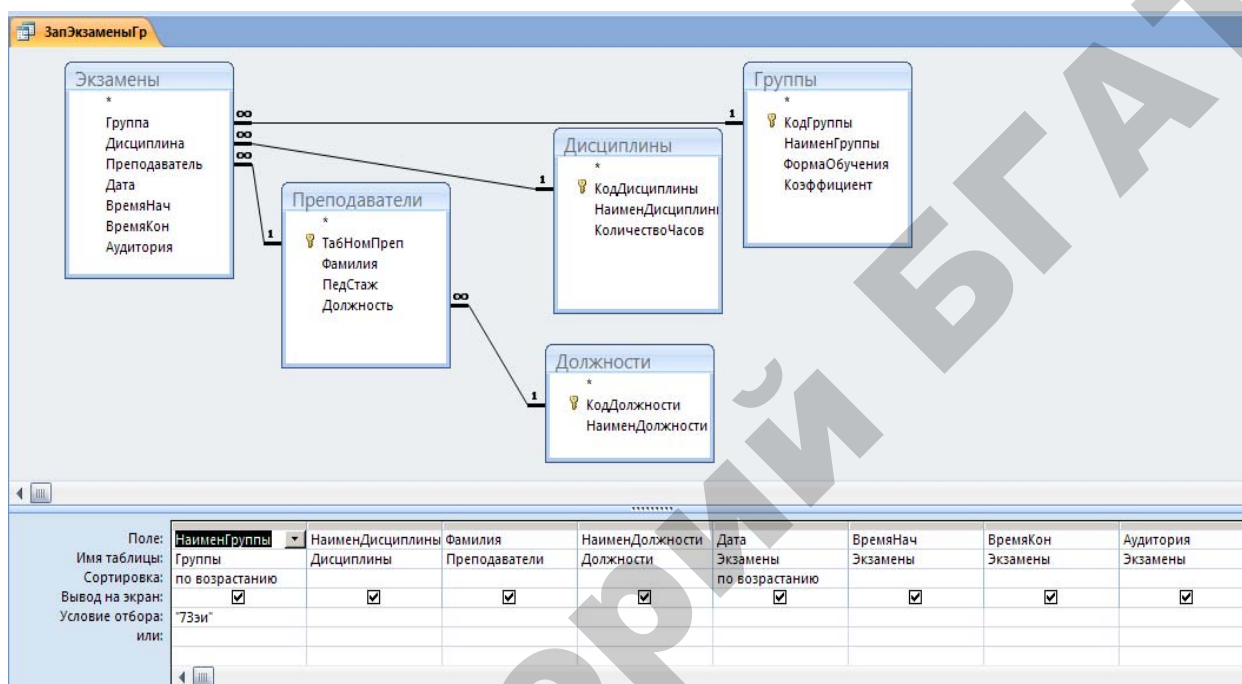


Рис. 4.20. Запрос **ЗапЭкзаменыГр** в режиме КОНСТРУКТОРА

Создание запросов с параметрами

Задание 5. Создать запрос **ЗапЭкзаменыПар**, позволяющий просмотреть расписание экзаменов заданной группы.

Выполняемые действия

1. Создать запрос **ЗапЭкзаменыПар** по аналогии с **ЗапЭкзаменыГр**, но в строку **Условие отбора** поля **НаименГруппы** вместо надписи «73эи» ввести приглашение на ввод условия отбора в квадратных скобках, например, [Введите наименование группы]. Получился запрос с параметром. При выполнении запроса перед формированием таблицы будет выводиться заданное приглашение: «Введите наименование группы». И, вводя нужную

группу, можно получить расписание экзаменов для нее. Запрос **ЗапЭкзаменыПар** в режиме конструктора изображен на рисунке 4.21.

2. Выполнить и сохранить запрос.

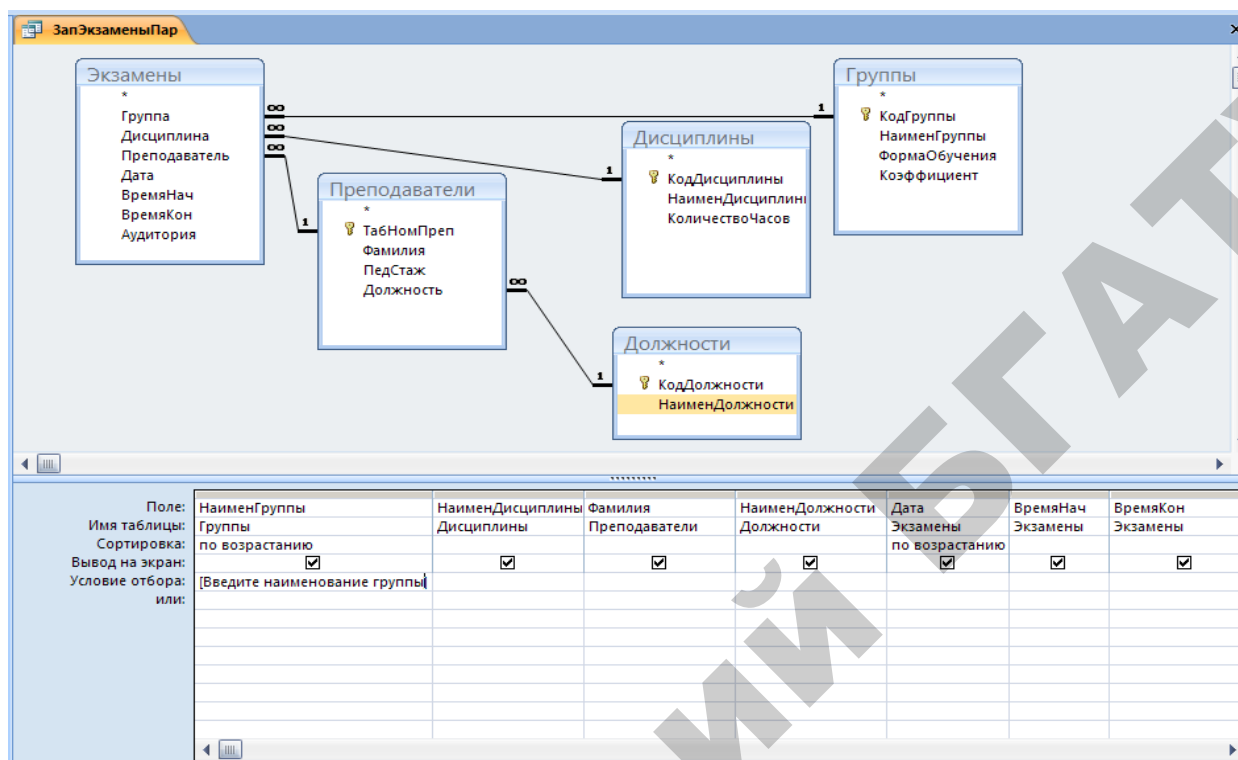





Рис. 4.21. Запрос **ЗапЭкзаменыПар** в режиме КОНСТРУКТОРА

Замечание. Если в строку *Условие отбора* в поле *НаименДисциплины* ввести выражение **Like[Введите первую букву наименования]&"*"**, то можно выбрать экзамены по дисциплинам, начинающимся с введенной буквы. А если ввести выражение **Like"*"&[Введите букву, содержащуюся в наименовании]&"*"**, то будут выводиться экзамены по дисциплинам, наименования которых содержат 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 режиме выполнения запроса кнопками ,  **Дополнительно** или щелчком клавишей мыши по кнопке  в заголовке соответствующей графы таблицы.

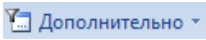
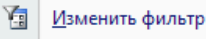
Задание 6. *Применив к запросу **ЗапЭкзамены** фильтр, выбрать экзамены, которые принимают профессора.*

Выполняемые действия

Выполнить запрос **ЗапЭкзамены**.


1 способ

1. Выделить столбец **НаименДолжности**.

2. Нажать кнопки  ,  , в столбце **НаименДолжности** раскрыть список значений и выбрать **Профессор**.

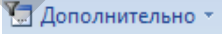

3. Нажать кнопку .

2 способ

При выделенном столбце **НаименДолжности** нажать кнопку  или нажать кнопку  в заголовке столбца **НаименДолжности**

1. В появившемся окне со списком значений поля **НаименДолжности** все отметки снять, отмеченным оставить только значение **Профессор**.

2. Нажать ОК.

Замечание. Для удаления фильтра нажать кнопку , затем .

Задание 7. *Применив к запросу **ЗапЭкзамены** фильтр, выбрать экзамены по истории.*

Выполняемые действия

1) Нажать кнопку  в заголовке столбца **НаименДисциплины**.

2) В открывшемся окне выбрать **Текстовые фильтр, Содержит**.

3) В окне **НАСТРАИВАЕМЫЙ ФИЛЬТР** ввести слово **история**.

Нажать ОК.

Выполнение вычислений в запросах

Задание 8. *С помощью **КОНСТРУКТОРА** создать запрос **ЗапЭкзаменыВыч**, результирующая таблица которого должна содержать расписание в виде, удобном для преподавателя, и содержать сведения в следующем порядке: **Преподаватель (Фамилия), Должность, НаименДисциплины, НаименГруппы, ФормаОбучения, КолЧасовВГруппе, Дата, ВремяНач, ВремяКон, Часы, Аудитория**.*

*Поле **КолЧасовВГруппе** рассчитывается по формуле:*


КолЧасовВГруппе:КоличествоЧасов(Таблица Дисциплины)* Коэффициент (Таблица Группы).

*Поле **Часы** рассчитывается по формуле:*

Часы: DateDiff("h";[ВремяНач];[ВремяКон]).

Сортировка записей должна быть следующей: 1-е поле **Преподаватель (Фамилия)** – по возрастанию; 2-е поле **Дата** – по возрастанию.

Выполняемые действия

1. Перейти в режим КОНСТРУКТОРА запросов.
2. В появившемся окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицы **Экзамены**, **Группы**, **Дисциплины**, **Преподаватели** (двойным щелчком клавишей мыши по именам таблиц). Закрыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.
3. На экране появилось окно конструктора запросов. В строку ПОЛЕ бланка запроса следует переместить поля, включаемые в результирующую таблицу: **Фамилия**, **Должность** – из таблицы **Преподаватели**, **НаименДисципл** – из таблицы **Дисциплины**, **НаименГруппы**, **ФормаОбучения** – из таблицы **Группы**, **Дата**, **ВремяНач**, **ВремяКон**, **Аудитория** – из таблицы **Экзамены**. Для полей **КолЧасовВГруппе** и **Часы** оставить соответствующие колонки пустыми.
4. В строке **Вывод на экран** проверить наличие флажков (V) для всех полей.
5. В строку **Поле** первой пустой колонки ввести выражение:
КолЧасовВГруппе:Дисциплины!КоличествоЧасов*Группы!Коэффициент
Замечание. Чтобы ввести формулу с помощью *Построителя выражений*, следует выполнить действия:
 - а) нажать кнопку  (*построитель*) на панели инструментов;
 - б) в окне Построителя выражений в левом списке раскрыть вкладку **Таблицы** и выбрать таблицу **Дисциплины**, щелкнув клавишей мыши по ее имени;
 - в) в центральном списке дважды щелкнуть клавишей мыши по полю **КоличествоЧасов**, чтобы его имя появилось в верхней части окна построителя;
 - г) ввести операцию *, далее выбрать таблицу **Группы** и поле **Коэффициент**;
 - д) нажать ОК. Формула появится в бланке запроса;
 - е) в режиме КОНСТРУКТОРА выделить построенное поле, вызвать контекстное меню правой кнопкой мыши, выбрать пункт СВОЙСТВА и на вкладке ОБЩИЕ выбрать **Формат поля – Фиксированный**.
6. В строку **Поле** второй пустой строки ввести (можно вручную) выражение:
Часы: DateDiff("h";[ВремяНач];[ВремяКон]),
а в строке **Вывод на экран** установить флажок [V] для обоих вычисляемых полей, чтобы они стали видимыми.
7. Выполнить требуемую сортировку.
8. Выполнить и сохранить запрос.

Окончательный вид запроса в режиме КОНСТРУКТОРА изображен на рисунке 4.22.

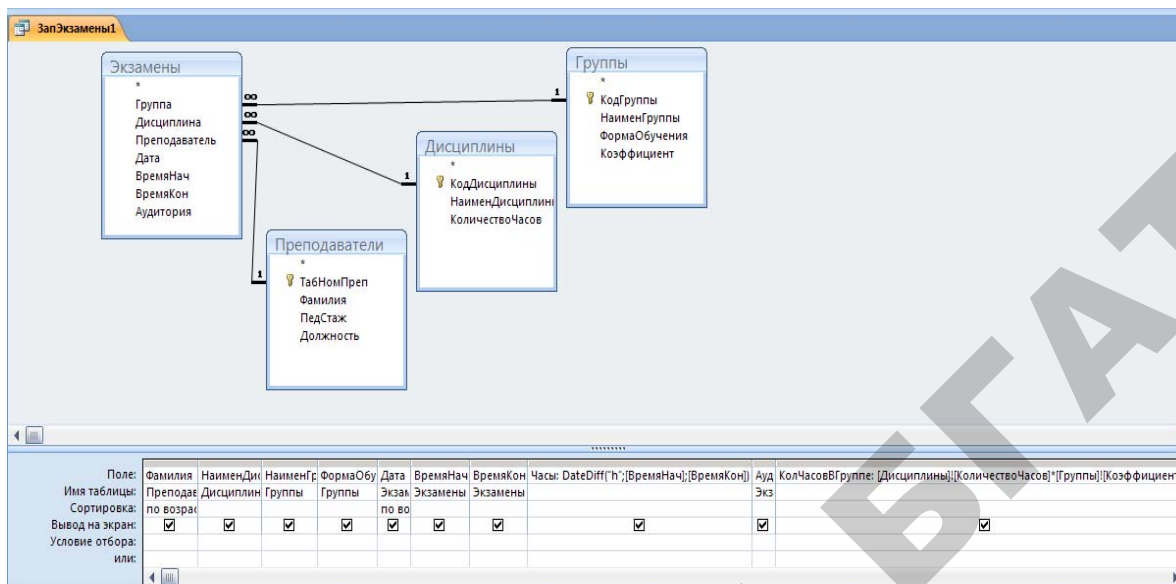


Рис. 4.22. Запрос **ЗапЭкзаменыВыч** в режиме КОНСТРУКТОРА

Создание простого запроса с помощью МАСТЕРА ЗАПРОСОВ


Задание 9. С помощью мастера запросов создать запрос **ЗапУспеваемостьМас**, содержащий сведения об успеваемости студентов, результирующая таблица которого должна содержать сведения об успеваемости студентов в следующем порядке: **Группа, Студент, Дисциплина, КоличествоЧасов, оценка**. Все объекты представляются своими наименованиями.

Выполняемые действия

1. На вкладке **Создание** ленты меню в группе **Другие** нажать кнопку



2. В окне **НОВЫЙ ЗАПРОС** выбрать пункт **Простой запрос**.

3. Из нужных полей соответствующих таблиц скомпоновать макет запроса, перемещая поля кнопкой  между листами. Выбрать поля: **НаименГруппы** (из таблицы **Группы**), **Фамилия, Имя, Отчество** (из таблицы **Студент**), **НаименДисциплины, КоличествоЧасов** (из таблицы **Дисциплины**), **оценка** (из таблицы **Оценки**). Нажать **Далее**.

4. В появившемся окне отметить пункт **Выбрать подробный отчет**, нажать **Далее**.


5. В следующем окне задать имя запроса **ЗапУспеваемостьМас** и выбрать одно из предложенных действий:

- *Открыть результат выполнения запроса;*
- *Изменить структуру запроса. Нажать **Готово**.*

Формирование запросов с группировкой

Задание 10. *Создать запрос **ЗапЭкзаменыГрупп**, показывающий количество экзаменов в каждой группе.*

Выполняемые действия

1. Перейти в режим **КОНСТРУКТОРА** запросов.
2. В появившемся окне **ДОБАВЛЕНИЕ ТАБЛИЦЫ** нажать вкладку *Запросы* и добавить запрос **ЗапЭкзамены**. Закрыть окно **ДОБАВЛЕНИЕ ТАБЛИЦЫ**.
3. В строку *Поле* макета переместить поля **НаименГруппы** и **НаименДисциплины** из **ЗапЭкзамены**.
4. В строке **Вывод на экран** проверить наличие флажков (V) для полей.
5. Нажать кнопку , в бланке запроса появится новая строка **Групповая операция**, в которой для всех полей указано **Группировка**.
6. В поле **НаименДисциплины** вместо надписи **Группировка** задать нужную функцию (Count), выбрав ее из списка функций, появившихся по щелчку клавишей мыши в правой части поля.
7. Сохранить и просмотреть запрос.

Создание перекрестного запроса

Задание 11. *Подсчитать количество часов, потраченное на прием экзаменов каждым преподавателем в каждой группе и вывести результаты в виде перекрестной таблицы. Эту задачу решит перекрестный запрос.*

Выполняемые действия

1. На вкладке **Создание** ленты меню в группе **Другие** нажать кнопку



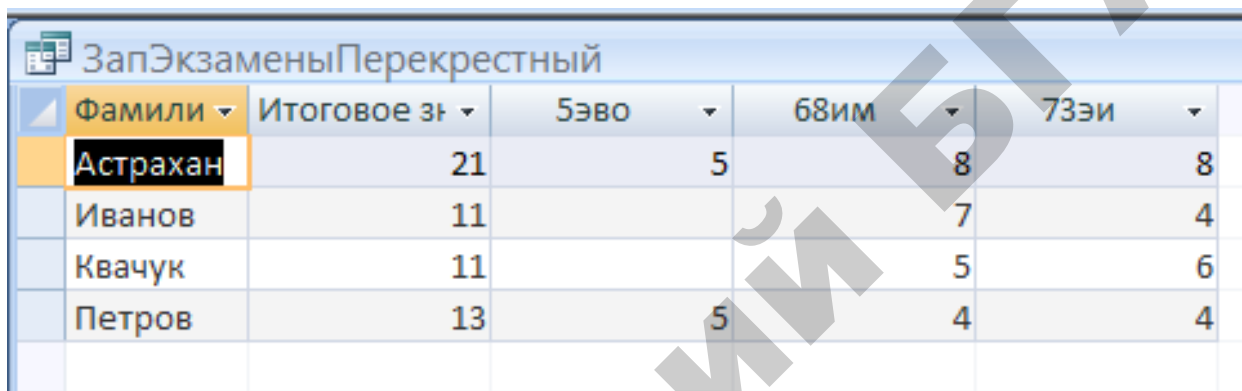
2. В окне **НОВЫЙ ЗАПРОС** выбрать пункт *Перекрестный запрос*.
В появившемся окне выполнить следующие действия.

• Выбрать **Показать запросы**, в качестве источника указать **ЗапЭкзаменыВыч**, нажать **Далее**.

- Для наименования строк выбрать поле **Фамилия**, нажать **Далее**.
- Для наименования столбцов выбрать **НаименГруппы**, нажать **Далее**.
- Выбрать функцию, которую необходимо выполнить для ячеек на пересечении строк и столбцов. В нашем случае выбрать функцию **Сумма(Sum)** и указать поле **Часы**, нажать **Готово**.

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

Результирующая таблица перекрестного запроса **ЗапЭкзаменыПерекрестный** изображена на рисунке 4.23.



Фамилия	Итоговое з...	5эво	68им	73эи
Астрахан	21	5	8	8
Иванов	11		7	4
Квачук	11		5	6
Петров	13	5	4	4

Рис. 4.23. Результирующая таблица перекрестного запроса **ЗапЭкзаменыПерекрестный**

Задания для самостоятельного выполнения к лабораторной работе № 4

Кроме рассмотренных запросов создать следующие запросы в базе данных **ВУЗ**.

1. Создать запрос **ЗапОценкиСамост**, результирующая таблица которого соответствовала бы таблице **Оценки**, но объекты представить наименованиями, добавить поле **НаименГруппы** и добавить поле **ОценкаСтоБал**, вычисляемое по формуле: **Оценка *10**. Определить сортировку по фамилии студента.

2. Создать запрос **ЗапОценкиСамостПар** для вывода экзаменационных оценок заданного студента (по форме **ЗапОценкиСамост**).

3. Сконструировать запрос **ЗапОценкиСамостГрупп** для вывода суммарного балла каждого студента.

4. Построить перекрестный запрос **ЗапОценкиСамостПерекр**, показывающий средний балл каждой группы по каждой дисциплине.

5. Закрыть базу данных **ВУЗ**.

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

В разрабатываемом приложении выполнить задания 1–5 темы **Создание запросов на выборку**.

Продуктивный уровень (оценка 7–9)

В разрабатываемом приложении выполнить задания 1–6 темы **Создание запросов на выборку**.

Творческий уровень (оценка 10)

В индивидуальной базе данных создать запросы на выборку (не менее трех) с различной сортировкой с вычислениями, запросы с параметрами (не менее двух), запросы с группировкой (не менее двух), перекрестные запросы (не менее двух).

Вопросы для самоконтроля

1. Какими средствами в СУБД Access реализуется обеспечение целостности данных?
2. Поддерживает ли СУБД Access связи между таблицами типа «многие-ко-многим»?
3. Какое поле таблицы может быть объявлено первичным ключом? Может ли первичный ключ таблицы состоять из нескольких полей?
4. Перечислите основные понятия методики семантического моделирования.
5. Перечислите этапы проектирования баз данных.
6. На каком этапе проектирования баз данных формируются таблицы БД, ориентируясь на конкретную СУБД?
7. Перечислите основные типы полей таблиц СУБД Access.
8. В чем заключается нормализация таблиц БД? Сколько нормальных форм существует?

ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ ПО МОДУЛЮ 4

1. К какому типу СУБД относится Access?
 - реляционная;

- сетевая;
 - иерархическая.
2. В реляционной СУБД данные представлены в виде:
- графов;
 - таблиц;
 - текстов;
 - схем.
3. Выбрать наименования объектов Access:
- таблицы;
 - формулы;
 - запросы;
 - формы;
 - модели;
 - отчеты;
 - модули;
 - вопросы;
 - табеля;
 - макросы.
4. Запись таблицы содержит информацию:
- об одном экземпляре объекта;
 - о совокупности объектов;
 - о предметной области.
5. В каком режиме осуществляется заполнение данными полей таблицы(ввод значений полей таблицы)?
- в режиме конструктора;
 - в режиме таблицы;
 - в режиме дизайнера;
 - в режиме инженера.
6. Выберите правильные типы запросов в Access:
- запросы на выборку;
 - запросы-действия;
 - перекрестные запросы;
 - запросы с параметрами;
 - многостраничные запросы;
 - сводные запросы.
7. Если в запросе на выборку указана сортировка по двум полям, то производится:

- только сортировка по второму полю;
- осуществляется внутри группы записей с одинаковыми значениями первого поля;

- сортировка по второму полю осуществляется по всему набору записей;
- сортировка по второму полю осуществляется внутри группы записей с нулевым значением первого поля.

8. При создании запроса с параметрами приглашение на ввод параметров вводится:

- в макет запроса в строку “условие отбора”;
- в макет запроса в строку “вывод на экран”;
- в результирующую таблицу запроса;

9. В поле таблицы заносится:

- информация о системе;
- одна характеристика объекта;
- сведения об объекте.

10. В режиме конструктора таблицы видно:

- структура записи таблицы;
- значения полей таблицы;
- столбцы таблицы;
- свойства полей таблицы.

МОДУЛЬ 5

ПРОЕКТИРОВАНИЕ ОБЪЕКТОВ БАЗЫ ДАННЫХ – ЗАПРОСОВ-ДЕЙСТВИЙ, ФОРМ, ОТЧЕТОВ. ЯЗЫК SQL

В результате изучения модуля студент должен

знать: определения, назначение и сущность объектов базы данных: запросов-действий, форм, отчетов, общую характеристику и функциональные возможности языка SQL, типы данных и основные синтаксические конструкции языка SQL; назначение и технологию проектирования форм, отчетов, назначение основных разделов форм и отчетов.

уметь: создавать запросы-действия различных типов, конструировать SQL инструкции для создания запросов, создавать сложные формы различных типов, отчеты с несколькими уровнями группировки данных, использовать при создании форм и отчетов наиболее популярные элементы управления и подходы к оформлению.

КРАТКИЕ СВЕДЕНИЯ О ЗАПРОСАХ-ДЕЙСТВИЯХ И ФОРМАХ

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

- запрос на **создание** таблиц, который создает новую таблицу на основе всех или части данных из одной или нескольких таблиц;
- запрос на **удаление** записей (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 изменения в группу записей одной или нескольких таблиц.

Так как запросы-действия изменяют содержимое таблиц, обновляют, удаляют записи, или добавляют новые, перед выполнением этих запросов следует сделать копии изменяемых таблиц для того, чтобы в случае неудачного завершения операции обновления можно было вернуться к исходному.

Запросы-действия создаются с помощью конструктора. Тип запроса задается соответствующей кнопкой на вкладке **Создание** ленты меню на контекстной вкладке Конструктора запросов.

Запрос-действие, как правило, выполняется один раз, так как их выполнение вызывает изменение в БД.

Формы

Форма – это объект БД, предназначенный для организации интерфейса пользователя.

Назначение форм следующее:

1) для ввода данных в БД. Формы позволяют размещать поля на экране в произвольном порядке. Поэтому пользователь может организовать ввод данных в привычном для себя виде (например, подобно заполнению бланка);

2) форма применяется для просмотра и редактирования данных в БД. ((С помощью форм можно выводить на экран всю информацию или часть ее в представлении, выбранном пользователем.)) Форма позволяет выбрать данные из связанных таблиц (причем не все, а только необходимые пользователю), выполнять вычисления по формулам, скрывать или показывать некоторые данные в зависимости от значений других полей; корректировать поля таблицы.

Формы можно рассматривать как окна, через которые пользователи могут просматривать и изменять базу данных. Рационально построенная форма ускоряет работу с базой данных, поскольку пользователям не требуется искать то, что им нужно. Внешне привлекательная форма делает работу с базой данных более приятной и эффективной, кроме того она может помочь в предотвращении неверного ввода данных;

3) формы используются для управления ходом выполнения приложения. С помощью специальных элементов управления, называемых командными кнопками, можно вызвать открытие других форм, выполнение запросов, распечатку отчетов, выполнение процедур VBA;

4) формы могут предоставлять вспомогательную информацию о работе приложения или о выполняемых действиях. Можно вывести на экран сообщение или предупреждение об ошибке.

ОБЩИЕ СВЕДЕНИЯ ОБ ОТЧЕТАХ

Отчет – это гибкое средство для организации данных при выводе на печать. С помощью отчета можно вывести необходимые сведения в нужном виде. По сравнению с другими средствами распечатки данных (формами, запросами) отчеты обладают некоторыми преимуществами.

1. Отчеты могут быть использованы для получения красиво оформленных счетов, заказов на покупку, почтовых наклеек, материалов для презентаций, и других документов.

2. Они имеют широкие возможности для группировки данных и для вычисления промежуточных и общих итогов.

В отчете каждая группа и итоги по ней могут быть представлены отдельно и имеются следующие возможности:

а) для обеспечения иерархического представления данных можно определить до 10 уровней группировки;

б) можно отдельно определять заголовки и примечания для каждой из групп.

3. Можно произвести сложные вычисления не только внутри некоторой группы, но и по нескольким группам одновременно.

4. Есть возможность задать заголовок и примечание для всего отчета в целом.

5. В любое место отчета можно вставить подчиненный отчет, картинку, диаграмму.

ЭЛЕМЕНТЫ УПРАВЛЕНИЯ В ФОРМАХ И ОТЧЕТАХ

Элементы управления – это графические объекты, которыми заполняются области формы и отчета. Элементы управления определяют внешний вид формы (или отчета), порядок обработки формы (или отчета), а также служат для создания связи между формой и данными таблицы или запроса, являющегося источником данных для формы. Существуют такие элементы управления, как *поля, надписи, списки,; переключатели, кнопки, линии* (рис. 5.1).

Для нанесения элементов управления на форму используются инструменты группы “**Элементы управления**” контекстной вкладки “**Конструктор форм**” вкладки “**Создание**” ленты меню.

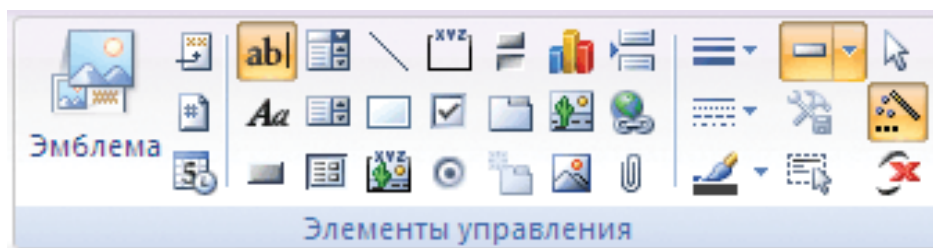


Рис. 5.1. Элементы управления

Элементы управления бывают присоединенные, свободные, вычисляемые.

Присоединенный элемент управления – элемент управления, источником данных которого служит поле таблицы или запроса. Присоединенный элемент управления служит для отображения значений полей базы данных. Значения могут быть текстовыми, числовыми, логическими, датами, рисунками или диаграммами.

Свободные элементы управления – элементы управления, не имеющие источника данных (например, поля или выражения). Свободные элементы управления используются для вывода на экран сведений (надпись), линий, прямоугольников и рисунков. Примером свободного элемента является надпись, которая отображает заголовок формы.

Вычисляемые элементы управления – элементы управления, источником данных которых является выражение, а не поле. Для задания значения, которое должно содержаться в таком элементе управления, необходимо задать *выражение*, служащее источником данных элемента. Выражение – это сочетание операторов (таких, как = и +), имен других элементов управления, имен полей, функций, возвращающих единственное значение, и констант. Например, в следующем выражении цена изделия рассчитывается с 25 % скидкой путем умножения значения поля «Цена за единицу» на константу (0,75), то есть [Цена за единицу] * 0,75.

В выражении могут использоваться данные поля в базовой таблице или запросе формы или данные из другого элемента управления формы.

Назначение наиболее употребляемых элементов управления

Надпись – служит для отображения текста.

Поле – позволяет вводить данные базы и создавать вычисляемые элементы.

Поле со списком – представляет собой раскрывающийся список, который позволяет вводить значения.

Список – позволяет отображать список значений в форме или отчете.


Кнопка – используется для создания кнопки в форме.

Рис. – используется для вставки в форму рисунка из библиотеки рисунков.

Линия – используется для изображения линии.

Прямоугольник – используется для выделения группы элементов формы.

Присоединенный элемент управления – это элемент, связанный с некоторым полем таблицы, на котором базируется форма (отчет). Чтобы вставить его в форму следует открыть список полей соответствующей кнопкой, выделить нужное поле и при нажатой клавише мыши переместить его в область данных макета формы. Образ поля и образ присоединенной надписи появится в области данных макета формы. Оба поля (вместе и по отдельности) можно переместить в любое место области данных. Вставка в форму свободного элемента управления осуществляется перенесением соответствующей кнопки с панели элементов в нужное место области формы.

Многие инструменты в группе **Элементы управления** доступны только тогда, когда форма открыта в режиме конструктора. Чтобы переключиться в режим конструктора, щелкните правой клавишей мыши имя формы в области переходов и выберите команду **Конструктор** .

Замечание. Чтобы появилось название инструмента, следует поместить указатель мыши на инструмент и подождать несколько секунд. Появится название инструмента.

ЛАБОРАТОРНАЯ РАБОТА № 5

СОЗДАНИЕ ЗАПРОСОВ-ДЕЙСТВИЙ

Цель работы: овладеть основными приемами создания запросов-действий.

ВНИМАНИЕ!!! Перед каждым сеансом работы с запросами-действиями программами необходимо нажать на кнопку ПАРАМЕТРЫ, расположенную в строке «Предупреждения системы безопасности...», в открывшемся окне «Оповещение системы безопасности» отметить флажок «Включить это содержимое» и нажать на кнопку ОК.

Построение запросов на создание таблицы

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

Задание 1. *Сконструировать запрос **ЗапСозЭкзаменыПодробно** на создание таблицы **ЭкзаменыПодробно**, аналогичной результирующей таблице запроса **ЗапЭкзамены**, созданного в практической работе № 3.*

Выполняемые действия

1. Перейти в режим КОНСТРУКТОРА запросов.
2. Сформировать макет запроса, аналогичный **ЗапЭкзамены**, или скопировать запрос **ЗапЭкзамены** с именем **ЗапСозЭкзаменыПодробно**.

3. Указать тип запроса, нажав кнопку .

4. В появившемся окне СОЗДАНИЕ ТАБЛИЦЫ указать имя создаваемой таблицы **ЭкзаменыПодробно**. В дальнейшем Access будет запрашивать подтверждение на удаление существующей таблицы **ЭкзаменыПодробно** перед каждым выполнением запроса на создание.

5. Сохранить запрос с именем **ЗапСозЭкзаменыПодробно**, обратить внимание на значок для запросов этого типа.


6. Для создания таблицы выполнить запрос **ЗапСозЭкзаменыПодробно**. Убедиться в том, что новая таблица с именем **ЭкзаменыПодробно** появилась в списке таблиц. Просмотрите созданную таблицу.

Задание 2. С помощью запроса создать таблицу **СтудНазнСтип**, содержащую следующую информацию о студентах: **НомерЗачКн, Фамилия, Имя, Отчество, СреднийБалл, ВидСтип, РазмерСтип**. Записи таблицы должны следовать в порядке возрастания среднего балла.

Замечание. Размер и тип стипендии студента зависит от среднего балла по дисциплинам. Соответствие между средним баллом и стипендией отражено в таблице **Стипендия**. Поэтому сначала необходимо создать запрос, рассчитывающий средний балл каждого студента (запрос с группировкой), а затем на основе этого запроса и таблицы **Стипендия** сконструировать запрос на создание нужной таблицы.

Задание 2.1. Создайте запрос **ЗапСтудСрБалл** с группировкой, рассчитывающий средний балл каждого студента, содержащий поля **НомерЗачКн, Фамилия, Имя, Отчество, СреднийБалл**. Порядок следования записей в результирующей таблице по возрастанию поля **СреднийБалл**.

Выполняемые действия

1. Перейти в режим КОНСТРУКТОРА запросов.
2. В появившемся окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицы **Студенты** и **Оценки**. Закрывать окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.
3. В строку **Поле** макета запроса переместить поля **НомерЗачКн, Фамилия, Имя, Отчество** из таблицы **Студенты**, **Оценка** – из таблицы **Оценки**.
4. В строке **Вывод на экран** проверить наличие флажков (V) для полей.
5. Нажать кнопку , в бланке запроса появится новая строка **Групповая операция**, в которой для всех полей указано **Группировка**.
6. В столбце **Оценка** в строке **поле** добавить надпись, чтобы получилось **СреднийБалл:Оценка**, а в строке **Групповая операция** вместо надписи **Группировка** задать функцию вычисления среднего балла (Avg), выбрав ее из списка функций, появившихся по щелчку клавишей мыши в правой части поля.
7. Определить порядок сортировки по возрастанию среднего балла.

Создаваемый запрос в режиме конструктора приведен на рисунке 5.2.

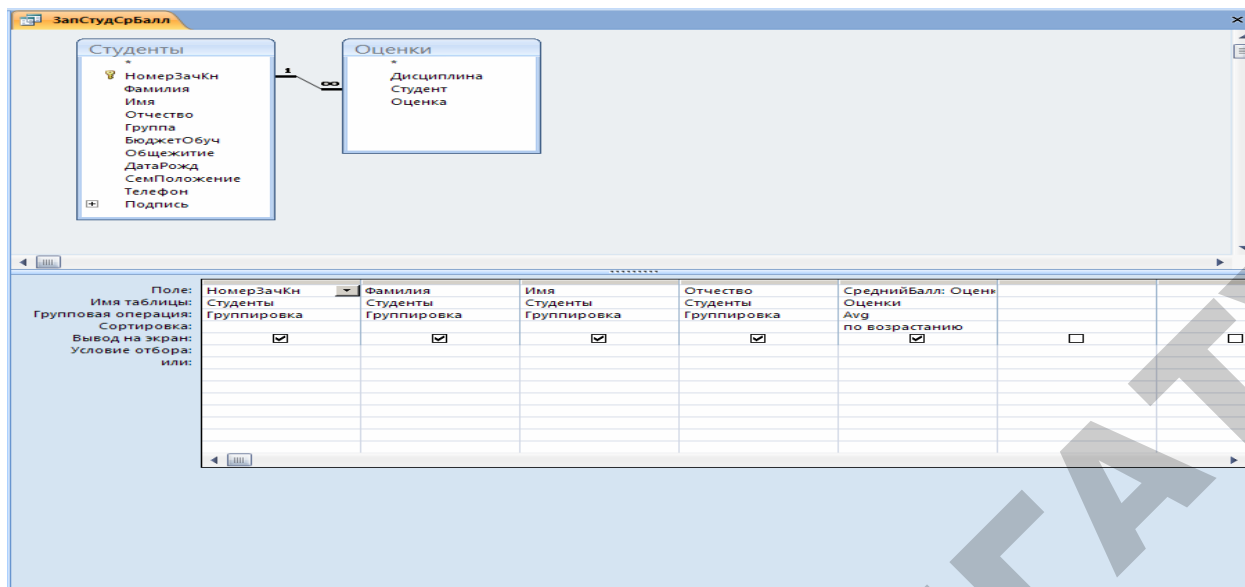


Рис. 5.2. ЗапСтудСрБалл в режиме КОНСТРУКТОРА

8. Выполнить запрос и убедиться, что получена ожидаемая таблица. Сохранить запрос с именем **ЗапСтудСрБалл**.

Задание 2.2. Создать запрос **ЗапСозСтудНазнСтип** на создание таблицы **СтудНазнСтип**.

Выполняемые действия

1. Перейти в режим КОНСТРУКТОРА запросов.
 2. В появившемся окне **ДОБАВЛЕНИЕ ТАБЛИЦЫ** добавить запрос **ЗапСтудСрБалл** и таблицу **Стипендия**. Закрывать окно **ДОБАВЛЕНИЕ ТАБЛИЦЫ**.

3. В строку **Поле** макета запроса переместить поля **НомерЗачКн**, **Фамилия**, **Имя**, **Отчество**, **СреднийБалл** из запроса **ЗапСтудСрБалл**, **ВидСтип**, **РазмерСтип** – из таблицы **Стипендия**.

4. В строке **Вывод на экран** проверить наличие флажков (V) для полей.

5. В колонку **СреднийБалл** в строку **Условие отбора** ввести выражение:

>[Стипендия]![СрБаллМин] And <=[Стипендия]![СрБаллМакс]

Определить сортировку поля **СреднийБалл** по возрастанию.

6. Выполнить запрос.

7. Для назначения создаваемому запросу типа **Создание таблицы** нажать

кнопку . В окне **СОЗДАНИЕ ТАБЛИЦЫ** указать имя создаваемой таблицы **СтудНазнСтип**.

8. Сохранить запрос с именем **ЗапСозСтудНазнСтип**, обратить внимание на значок для запросов этого типа. Созданный запрос в режиме КОНСТРУКТОРА изображен на рисунке 5.3.

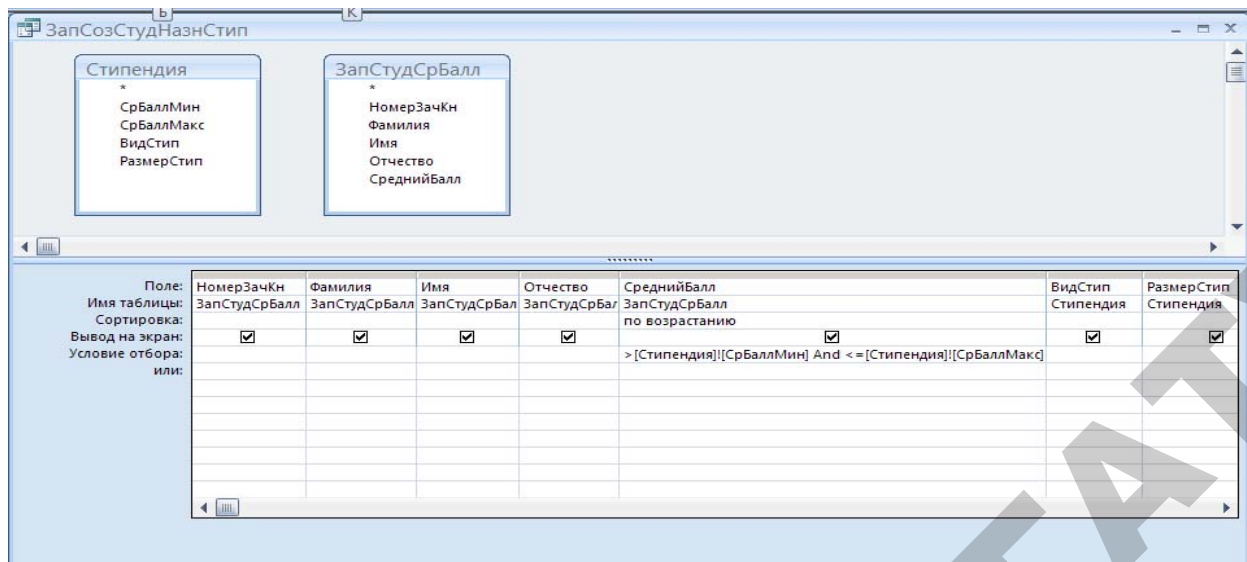


Рис. 5.3. Запрос ЗапСозСтудНазнСтип в режиме КОНСТРУКТОРА

9. Для создания таблицы выполнить запрос **ЗапСозСтудНазнСтип**. Убедитесь в том, что новая таблица появилась в списке таблиц с именем **СтудНазнСтип**. Просмотрите созданную таблицу.

Конструирование запросов на обновление и на добавление

Задание 3. *Принято решение принимать экзамены по подгруппам. Экзамен во второй подгруппе принимается в тот же день, что и в первой через 10 минут после окончания экзамена в первой подгруппе и длится 3 часа. Создать расписание для принятия экзамена по подгруппам.*

Задание 3 разобьем на более мелкие задания.


Задание 3.1. *В режиме КОНСТРУКТОРА добавить в таблицу **ЭкзаменыПодробно** поле **Подгруппа** (числовое) в конец макета.*

Создание запроса на обновление

Задание 3.2. *Создать запрос на обновление для заполнения нового поля **Группа** значением 1.*

Выполняемые действия

1. Перейти в режим КОНСТРУКТОРА запросов.
2. В появившемся окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицу **ЭкзаменыПодробно**. Закрыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.

3. В строку *Поле* макета запроса переместить поле **Подгруппа**.
4. В строке **Вывод на экран** проверить наличие флажков (V) для полей.
5. Определить тип запроса, нажав на кнопку . В появившуюся строку **Обновление** в обновляемую колонку ввести значение 1. Макет запроса изображен на рисунке 5.4.

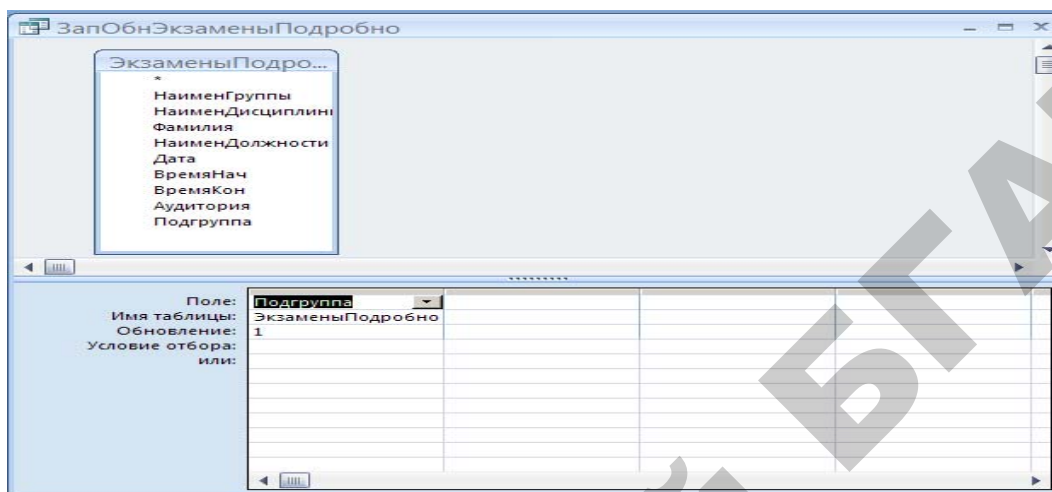


Рис. 5.4. Бланк запроса **ЗапОбнЭкзаменыПодробно**

Замечание. Если новое значение обновляемого поля должно рассчитываться по формуле, то формула вводится в строку **Обновление** обновляемого поля.


6. Сохранить созданный запрос с именем **ЗапОбнПодгрЭкзамены-Подробно** и закрыть его. Следует обратить внимание на значок запросов типа **ОБНОВЛЕНИЕ**.

7. Обновить таблицу **ЭкзаменыПодробно**, выполнив запрос **ЗапОбн-ПодгрЭкзаменыПодробно**. На оба появляющиеся вопроса о подтверждении обновления ответить «ДА». Результат работы запроса – занесение значения 1 в поле **Подгруппа** таблицы **ЭкзаменыПодробно**. В этом следует убедиться, открыв упомянутую таблицу. Теперь таблица **ЭкзаменыПодробно** содержит расписание экзаменов в первой подгруппе.

Создание запроса на добавление

Задание 3.3. Создать запрос на добавление **ЗапДобавЭкзамены-Подробно**, с помощью которого добавить в таблицу **ЭкзаменыПодробно** строки экзаменов во второй подгруппе, установив в качестве времени начала экзамена во второй подгруппе время окончания экзамена в первой подгруппе плюс 10 минут, а в качестве времени окончания экзамена – время начала экзамена плюс 3 часа.

Выполняемые действия

1. Перейти в режим КОНСТРУКТОРА запросов.
2. В появившемся окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицу **ЭкзаменыПодробно**. Закрывать окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.
3. В строку *Поле* макета запроса переместить все поля таблицы (можно выделив все поля при нажатой кнопке *Shift* и переместив левой кнопкой мыши).
4. В строке **Вывод на экран** проверить наличие флажков (V) для полей.
5. Определить тип запроса, нажав на кнопку . В открывшемся окне выбрать таблицу **ЭкзаменыПодробно**.
6. 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:

- в колонке **ВремяНач** в строке *Поле* вместо прежней надписи ввести выражение **DateAdd("n";10;[ВремяКон])**. Это функция добавления 10 временных интервалов “минута” (“n” означает минута) к времени **ВремяКон**. Результат выполнения функции – время начала экзамена во второй подгруппе;
- аналогично в колонке **ВремяКон** в строке *Поле* вместо прежней надписи ввести выражение **DateAdd("n";190;[ВремяКон])**. Функция добавит 190 минут (то есть три часа 10 минут) к времени **ВремяКон** и определит время окончания экзамена во второй подгруппе;
- в колонке **Подгруппа** в строке *Поле* вместо прежней надписи ввести выражение **[Подгруппа]+1**. Окончательный вид КОНСТРУКТОРА запроса изображен на рисунке 5.5.

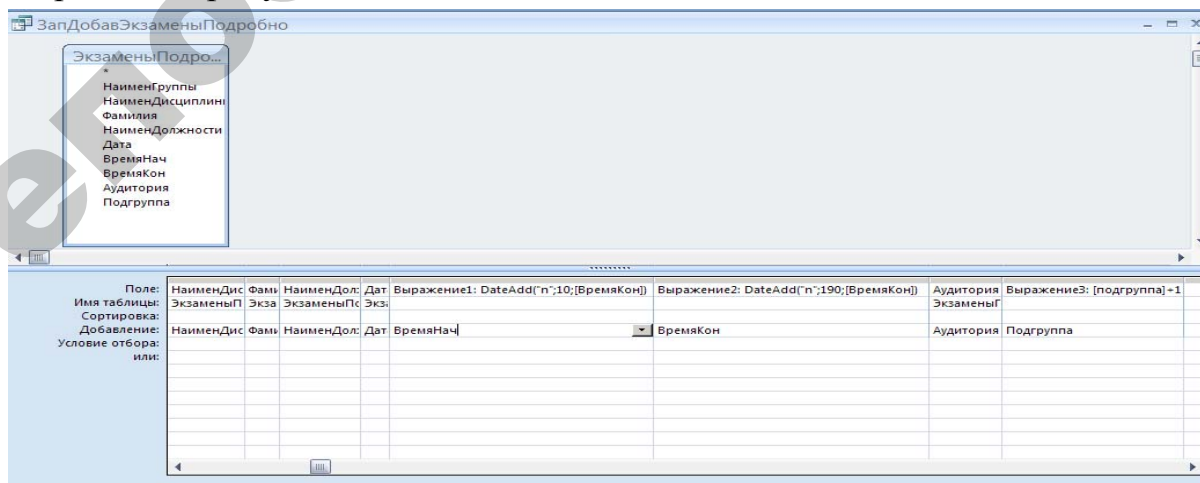


Рис.5.5. Бланк запроса ЗапДобавЭкзаменыПодробно

7. Сохранить запрос с именем **ЗапДобавЭкзаменыПодробно** и закрыть запрос. Следует обратить внимание на значок запросов типа ДОБАВЛЕНИЕ.

8. Создать копию таблицы **ЭкзаменыПодробно** с именем **ЭкзаменыПодробно1** для того, чтобы в случае неудачного обновления таблицы можно было восстановить ее первоначальное состояние.

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

9. Добавить необходимые записи в таблицу **ЭкзаменыПодробно**, выполнив запрос **ЗапДобавЭкзаменыПодробно**. На оба появляющиеся вопроса о подтверждении обновления ответить «ДА».

10. Открыв таблицу **ЭкзаменыПодробно**, убедиться, что добавление записей выполнилось правильно.

Замечание. Для удобства анализа результирующей таблицы следует создать расширенный фильтр. Для этого в режиме таблицы:

- выбрать **Главная/Сортировка и фильтр/Дополнительно/Расширенный фильтр**;

- в открывшийся бланк запроса перенести все поля таблицы;

- установить сортировку: первое поле – **НаименГруппы**, второе поле – **НаименДисциплины**;

- нажать кнопку *Применить фильтр*. В результате в таблице строки двух подгрупп по каждому экзамену выведутся друг за дружкой. Сравнить их и проанализировать правильность вычисления значений полей **ВремяНач** и **ВремяКон** для второй подгруппы будет удобно.

Создание запроса на удаление

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

Задание 4. Удалить из таблицы **ЭкзаменыПодробно** записи, касающиеся преподавателя Петрова.

Выполняемые действия

1. Перейти в режим КОНСТРУКТОРА запросов.
2. В появившемся окне ДОБАВЛЕНИЕ ТАБЛИЦЫ добавить таблицу **ЭкзаменыПодробно**. Закрыть окно ДОБАВЛЕНИЕ ТАБЛИЦЫ.
3. В строку *Поле* макета запроса переместить поле **Фамилия**.
4. В строке **Вывод на экран** проверить наличие флажков (V) для полей.
5. В строке УСЛОВИЕ для поля **Фамилия** ввести «Петров».

6. Определить тип запроса, нажав на кнопку .

7. Сохранить запрос с именем **УдалЭкзаменыПодробно**, обратив внимание на значок запросов этого типа. Закрыть запрос.

8. Выполнить запрос **УдалЭкзаменыПодробно**, подтвердить удаление ответом «ДА».

9. Для просмотра результата выполнения запроса открыть таблицу **ЭкзаменыПодробно** и убедиться, что заданные записи удалились из запроса.

Задания для самостоятельного выполнения к лабораторной работе № 5

В базе данных **ВУЗ** выполнить следующие задания.

1. В соответствии с Указом Республики Беларусь № 711 от 29 декабря 2009 года «Об увеличении всех видов государственных стипендий учащейся молодежи» Совету Министров Республики Беларусь дано указание увеличить на 5 % размеры всех видов государственных стипендий. Создать запрос **ЗапОбнСамост** на обновление таблицы **Стипендия**.

2. Удалить из таблицы **ЭкзаменыПодробно** экзамены в группах, в наименовании которых есть буква Э, создав запрос **ЗапУдЭкзаменыПодробно**.

Рекомендация: Ввести условие отбора: `Like "*" & "Э" & "*"`

3. Закрыть базу данных **ВУЗ**.

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

В разрабатываемом приложении выполнить задания 7, 8 темы **Создание запросов-действий**.

Продуктивный уровень (оценка 7–9)

В разрабатываемом приложении выполнить задания 7–10 темы **Создание запросов на выборку**.

Творческий уровень (оценка 10)

В индивидуальной базе данных создать запросы-действия: запрос на создание таблицы, запрос на обновление полей таблицы, запрос на добавление записей в таблицу, запрос на удаление записей из таблицы.

ЛАБОРАТОРНАЯ РАБОТА № 6

СОЗДАНИЕ ЗАПРОСОВ НА ЯЗЫКЕ SQL

Цель работы: изучить общую характеристику и функциональные возможности языка SQL, типы данных и основные синтаксические конструкции языка SQL; научиться конструировать команды на языке SQL, позволяющие выполнять простейшие манипуляции с данными в БД.

Общие сведения о языке SQL

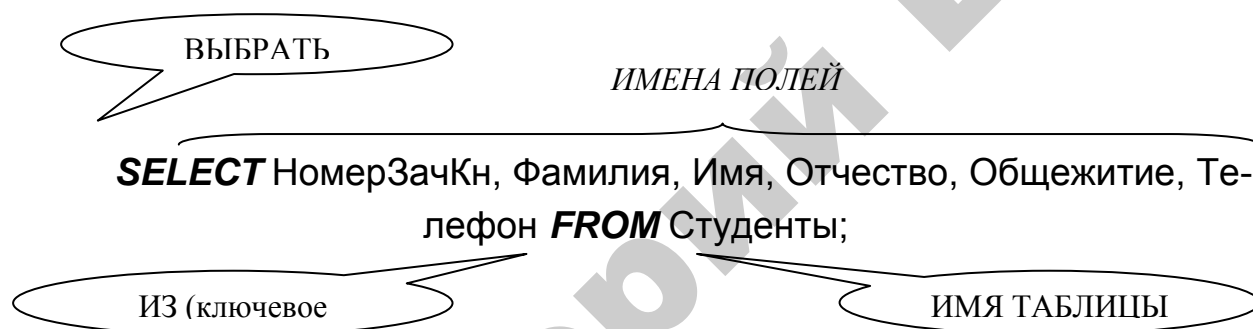
Структурированный язык запросов SQL (Structured Query Language) предназначен для создания запросов в реляционных базах данных. Он позволяет выполнять операции над таблицами (создание, удаление, изменение структуры) и над данными таблиц (выборка, изменение, добавление и удаление), а также производить некоторые сопутствующие операции (управление доступом, обеспечение целостности данных и т. п.). SQL основан на реляционном языке исчисления кортежей и является непроцедурным языком, т. е. не содержит операторов управления, организации подпрограмм, ввода-вывода и т. п. В связи с этим SQL автономно не используется, обычно он погружен в среду встроенного языка программирования СУБД. В различных СУБД состав операторов SQL может несколько отличаться. В данном изложении синтаксиса мы будем придерживаться стандарта, принятого в СУБД Access.

Структура команды языка SQL

Каждая команда SQL начинается с **глагола**, т. е. ключевого слова, описывающего **действие**, выполняемое командой. После глагола идет одно или несколько **предложений**. Предложение описывает данные, с которыми работает команда или содержит уточняющую информацию о действии, выполняемом ко-

мандой. Каждое предложение начинается с **ключевого слова**, такого как **WHERE** (где), **FROM** (откуда), **INTO** (куда), **HAVING** (имеющий). Одни предложения в команде являются обязательными, а другие – нет. Предложения могут содержать имена полей или имена таблиц БД, константы и выражения. Имена должны содержать от 1 до 18 символов, начинаться с буквы и не содержать пробелы и специальные символы пунктуации.


Запрос **ЗапСтуденты** на выборку следующих сведений о студентах (номер зачетной книжки – поле **НомерЗачКн**, фамилии – поле **Фамилия**, имени – поле **Имя**, отчества – поле **Отчество**, проживает ли в общежитии – поле **Общежитие**, телефон – поле **Телефон**) из таблицы **Студенты** на языке SQL будет представлен следующей командой:



Любой запрос, сформированный в режиме конструктора ACCESS, может быть представлен в виде инструкций на языке SQL, перейдя из режима **КОНСТРУКТОРА** в режим SQL.

Создание запроса на языке SQL

Для создания нового запроса в режиме SQL выполнить следующие команды на управляющей ленте:

Создание → **Конструктор запросов**, закрыть появившееся окно **Добавление таблицы** и перейти в SQL, нажав на кнопку . На экране появится окно текстового редактора для ввода и редактирования инструкций SQL. Выполнение и сохранение запроса осуществляется стандартными способами.

Создание таблиц и манипулирование данными в таблицах

Создание структуры новой таблицы. Команда CREATE TABLE

Задание 1. Запрос на создание новой таблицы Группы1, подобной по структуре таблице Группы.

CREATE TABLE Группы1 (КодГруппы **INT**, НаименГруппы **CHAR(5)**, ФормаОбучения **CHAR(10)**, Коэффициент **REAL**).

Будет создана новая таблица с именем Группы1, состоящая из следующих полей:

КодГруппы – тип данных: короткое целое – ключевое слово **INT**;

НаименГруппы – тип данных: строка, длиной до 5 символов **CHAR(5)**;

ФормаОбучения – тип данных: строка длиной до 10 символов **CHAR(10)**;

Коэффициент – тип данных: одинарное с плавающей точкой **REAL**.

Манипулирование данными

Ввод данных в таблицу. Команда INSERT

Задание 2. Запрос на заполнение таблицы Группы1 данными. Заполнить таблицу Группы1 данными о группах своего потока. Значение поля Коэффициент задать произвольно.

INSERT INTO Группы1 **VALUES** (1, “72эи“, “Дневная“, 1.5);

Каждое выполнение запроса заполняет одну строку таблицы. Если нужно заполнить несколько строк, то запрос можно выполнить несколько раз с разными значениями полей. Например,

INSERT INTO Группы1 **VALUES** (2, “91зэи“, “Заочная“, 0.8);

Изменение значений полей таблицы. Команда UPDATE

Задание 3. Запрос на обновление в таблице Группы1 коэффициента сложности обучения с 0.8 на 1.0 в группе 91зэи.

UPDATE Группы1 **SET** Коэффициент = 1.0 **WHERE**

НаименГруппы = “91зэи“;

Удаление записей из таблицы. Команда DELETE

Задание 4. Запрос на удаление из таблицы **Группы1** данных о группах заочного отделения.

DELETE FROM Группы1 **WHERE** ФормаОбучения = “Заочная“;
Из таблицы **Группы1** будут удалены все строки, у которых значение поля **ФормаОбучения** будет **Заочная** (соответствие должно быть полным).

Выбор данных. Команда **SELECT**

Выбор данных из одной таблицы

Задание 5. Запрос на выборку из таблицы **Группы1** следующих сведений о группах дневной формы обучения: код группы, наименование. Отсортировать полученные данные по коду группы.

SELECT КодГруппы, НаименГруппы **FROM** Группы1
WHERE ФормаОбучения = “Дневная“ **ORDER BY** КодГруппы;

Выбор данных из нескольких таблиц

Задание 6. Запрос на выборку из таблицы **Студенты** следующих сведений: номер зачетной книжки, фамилия, имя, наименование группы, телефон. Наименование группы следует взять из таблицы **Группы**.

SELECT Студенты.НомерЗачКн, Студенты.Фамилия,
Студенты.Имя, Группы.НаименГруппы, Студенты.Телефон
FROM Группы **INNER JOIN** Студенты **ON** Группы.КодГруппы =
Студенты.Группа;

Задания для самостоятельного выполнения к лабораторной работе № 6

1. Создать запрос на языке SQL на создание структуры таблицы **Студенты1**, подобной таблице **Студенты**. Включить в новую таблицу следующую информацию: номер зачетной книжки, фамилия, имя, группа, телефон.

2. Создать запрос на языке SQL на заполнение таблицы **Студенты1** конкретными значениями. Заполнить таблицу **Студенты1** пятью записями (среди которых должно быть 2 записи о студентах Вашей группы и 1 запись о себе). Можно выполнять один запрос несколько раз с разными значениями полей.

3. Создать запрос на языке SQL на **удаление** записи, содержащей информацию о студентах Вашей группы из таблицы **Студенты1**.

4. Создать запрос на языке SQL на **обновление**, выполняющий замену значения поля **группа** на номер Вашей группы записи с конкретным номером зачетки таблицы **Студенты1**.

5. Создать запрос на языке SQL на **выборку** из таблицы **Студенты** следующих сведений о студентах группы с кодом 1: номер зачетной книжки, фамилия, имя, телефон с сортировкой по возрастанию по полю **НомЗачКн**.

6. Создать запрос на языке SQL на **выборку** из таблицы **Преподаватели** сведений: фамилия, должность (причем наименование должности взять из справочника должностей).

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

В разрабатываемом приложении выполнить задания 11–13, 16 темы **Создание SQL запросов**.

Продуктивный уровень (оценка 7–9)

В разрабатываемом приложении выполнить задания 11–18 темы **Создание SQL запросов**.

Творческий уровень (оценка 10)

Для вариантов 6–20 выполнить задания 11–17 темы **Создание SQL запросов**. Или в индивидуальной базе данных создать на языке SQL все виды запросов, рассмотренных в лабораторной работе № 6. Кроме того, создать запрос на выборку данных из трех связанных таблиц.

ЛАБОРАТОРНАЯ РАБОТА № 7

КОНСТРУИРОВАНИЕ СЛОЖНЫХ ФОРМ

Цель работы: научиться конструировать сложные формы: форму с разным количеством подчиненных форм, сводную форму, создавать кнопку на форме.

Задание 1. Создать форму для просмотра результатов сессии.

Предусмотреть отображение экзаменов по группам и результатов сдачи каждого экзамена (экзаменационной ведомости).

Замечание. Для выполнения поставленной задачи удобно использовать форму с подчиненной формой. Главную форму следует создать на базе информации таблицы **Экзамены** (запрос **ЗапЭкзамены**). Подчиненную форму – по данным таблицы **Оценки**.


Задание 1.1. С помощью мастера форм создать ленточную форму **ФОценкиДляПодч** для отображения информации о результатах сдачи экзаменов студентами в следующем порядке: **КодГруппы**, **КодДисциплины**, **НомерЗачКн**, **Фамилия**, **Имя**, **Отчество**, **Оценка**.

Эта форма будет использована в качестве подчиненной при создании сложной формы.


Выполняемые действия

1. На вкладке **Создание** ленты меню в группе **Формы** нажать кнопки



В окне **СОЗДАНИЕ ФОРМЫ** выбрать нужные поля из соответствующих таблиц, переключая поля кнопкой  между листами. Выбрать поля: **КодГруппы** (из таблицы **Группы**), **КодДисциплины** (из таблицы **Дисциплины**), **НомерЗачКн**, **Фамилия**, **Имя**, **Отчество** (из таблицы **Студенты**), **Оценка** (из таблицы **Оценки**). Нажать *Далее*.

2. В следующем окне выбрать вид представления данных – **Оценки**. Нажать *Далее*.

3. Далее выбрать внешний вид формы – ленточный.
 4. Выбрать стиль, например, Стандартная.
 5. В следующем окне задать имя формы **ФОценкиДляПодч** и выбрать дальнейшее действие – Открыть форму для ввода данных. Нажать Готово. На экране появится созданная форма в режиме ФОРМЫ.
 6. Перейти в режим КОНСТРУКТОРА.
 7. Уменьшить ширину полей, поместив курсор на правую границу и переместив ее влево.
 8. В заголовке формы изменить сформировавшуюся запись на **РЕЗУЛЬТАТЫ ЭКЗАМЕНА**.
 9. Увеличить высоту области *Примечание формы* и поместить туда вычисляемое поле для расчета среднего балла перемещением кнопки  с панели элементов. В образовавшееся поле (свободное) ввести формулу: **=Avg([Оценка])**. Вызвать свойства поля и на вкладке Все выбрать Формат поля *Фиксированный*, Число десятичных знаков 1. В качестве присоединенной надписи, ввести **Средний балл**.
 10. Сохранить форму с именем **ФОценкиДляПодч**.
- Созданная форма в режиме КОНСТРУКТОРА приведена на рисунке 5.6.

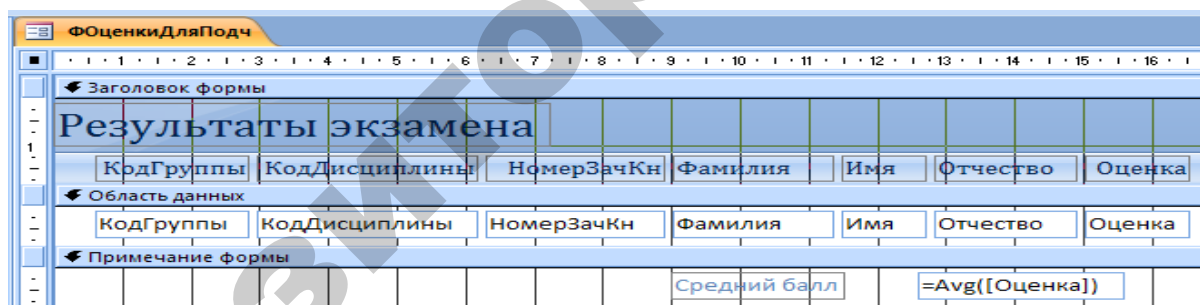



Рис. 5.6. Форма **ФОценкиДляПодч** в режиме КОНСТРУКТОРА

Задание 1.2. Создать форму **ФЭкзаменыСПодч** для отображения информации об экзаменах (главная форма). Кроме полей таблицы **Экзамены** на форму нанести поля **НаименГруппы**, **НаименДисциплины**, **Фамилия** (преподавателя), а так же рассчитать количество часов занятий по дисциплине в группе.

Выполняемые действия

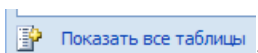
1. При выделенной таблице **Экзамены** на вкладке **Создание** ленты меню в группе **Формы** нажать кнопку . На экране появилась форма по таблице экзамены.

2. Перейти в режим КОНСТРУКТОРА. Уменьшить ширину полей, поместив курсор на правую границу и переместив ее влево.

3. Нанести на форму поля **НаименГруппы**, **Коэффициент** (из таблицы **Группы**), **НаименДисциплины**, **КоличествоЧасов** (из таблицы **Дисциплины**), **Фамилия** (из таблицы **Преподаватели**) с помощью кнопки



, используя вкладку



4. Создать вычисляемое поле для расчета количества часов текущей дисциплины в текущей группе. Для этого перенести значок с панели элементов в соответствующее место области данных. В образовавшееся поле (свободное) ввести формулу: $=[\text{КоличествоЧасов}] * [\text{Коэффициент}]$. Вызвать свойства поля и на вкладке **Все** выбрать **Формат поля** **Фиксированный**, **Число десятичных знаков** **1**. Присоединенную надпись перенести после вычисляемого поля и ввести слово «часов».

5. Сохранить форму с именем **ФЭкзаменыСПодч**.

6. Просмотреть форму, убедиться, что выводится ожидаемая информация.

Задание 1.3. Построить сложную форму **ФЭкзаменыСПодч**, объединив главную форму с подчиненной.

Выполняемые действия

1. Открыть форму **ФЭкзаменыСПодч** в режиме КОНСТРУКТОРА.

2. Удерживая левую клавишу мыши, перетащить имя формы **ФОценкиДляПодч** из области переходов в свободное место области данных формы **ФЭкзаменыСПодч**. На форме появится (очертится) образ (область) подчиненной формы, называемый элементом управления подчиненной формы.

3. Выделить область подчиненной формы (чтобы маркеры находились на границе области). Вызвать свойства правой кнопкой мыши и на вкладке **Данные** установить значения свойств **Основные поля** и **Подчиненные поля**. Для этого нажать кнопку и в окне **Связь с полями подчиненной формы** выбрать из списка **Основные поля** из основной формы и **Подчиненные поля** из подчиненной формы, как указано на рисунке 5.7.

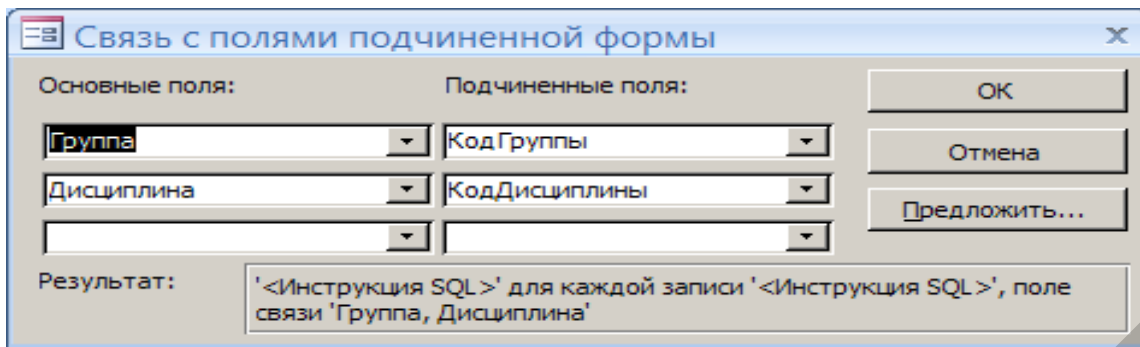


Рис. 5.7. Окно связи полей главной и подчиненной формы

Окно свойств подчиненной формы приведено на рисунке 5.8.

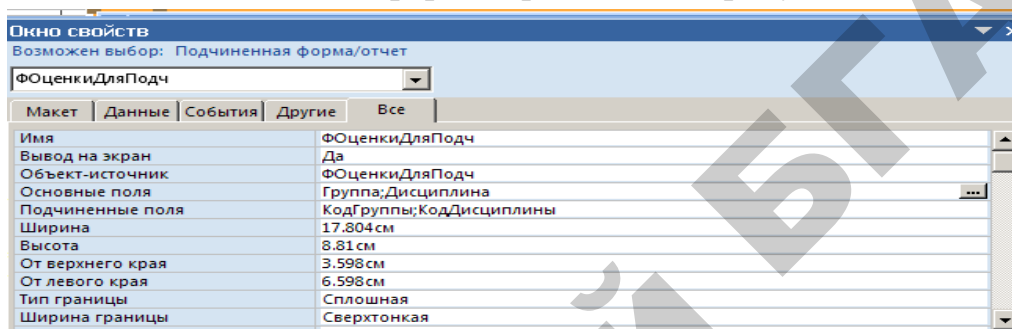


Рис. 5.8. Окно свойств подчиненной формы

Сложная форма в режиме конструктора приведена на рисунке 5.9.

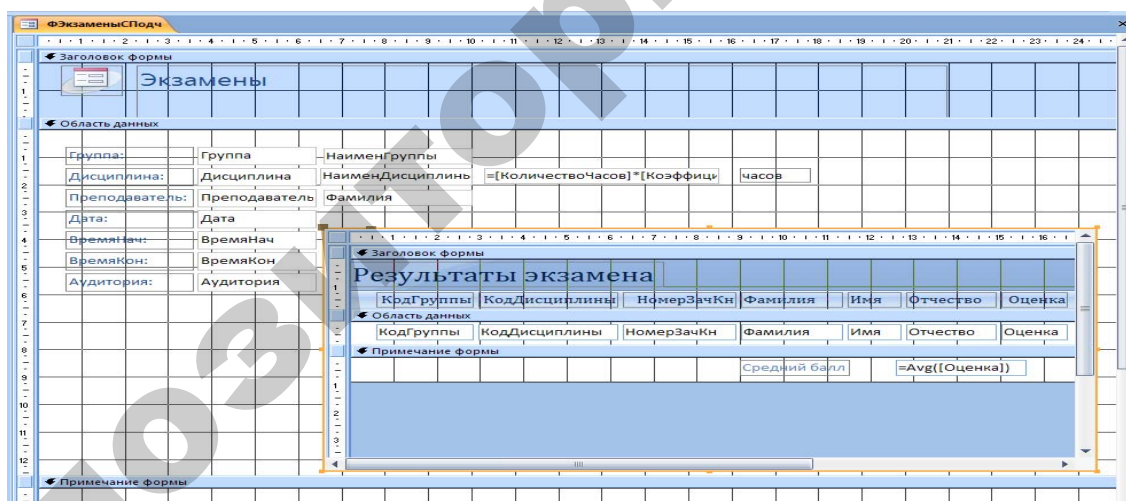



Рис. 5.9. Форма ФЭкзаменыСПодч в режиме КОНСТРУКТОРА

Задание 2. Создать форму для просмотра полной информации по каждому студенту. Предусмотреть вывод всей информации о студенте из справочника, результаты сдачи им всех экзаменов, средний балл, вид и размер назначенной стипендии.

Замечание. Для выполнения задания удобно использовать форму (по таблице Студенты) с двумя подчиненными формами (по информации таблицы Оценки и по таблице СтудНазнСтип).

Задание 2.1. Создать форму **ФСтудентыСПодч**, отображающую всю информацию о студенте (главную форму по таблице **Студенты**).

Выполняемые действия


1. При выделенной таблице **Студенты** на вкладке **Создание** ленты меню в группе **Формы** нажать кнопку  **Форма**. На экране появилась форма по таблице **Студенты** с подчиненной формой по таблице **Оценки**.


2. Можно, перейдя в режим **КОНСТРУКТОРА**, изменить внешний вид формы, например, уменьшить ширину полей главной формы, поместив курсор на правую границу и переместив ее влево, или нанести на форму оформительские элементы.

Задание 2.2. С помощью конструктора форм создать форму **ФСтипДляПодч** для отображения информации о назначении стипендии студентам в следующем порядке: **НомерЗачКн, СреднийБалл, ВидСтип, РазмерСтип**.

Эта форма будет использована в качестве подчиненной при создании сложной формы **ФСтудентыСПодч**.

Выполняемые действия

1. При выделенной таблице **СтудНазнСтип** в области переходов нажать кнопку **Конструктор форм**  на вкладке *Создание ленты* меню в группе **Формы**. Откроется окно конструктора форм, содержащее пустую область данных формы.

2. Нажать кнопку **Добавить поля**  и из открывшегося списка полей перенести в область данных формы поля таблицы **СтудНазнСтип**: **НомерЗачКн, СреднийБалл, ВидСтип, РазмерСтип** (двойным щелчком по каждому полю). В области данных формы появятся поля (правый столбец) вместе с присоединенными надписями (левый столбец).

3. Откорректировать размеры формы.

4. Выполнив форму, убедиться, что выводится нужная информация.

5. Сохранить форму с именем **ФСтипДляПодч**.

Задание 2.3. Построить сложную форму **ФСтудентыСПодч**, объединив главную форму с подчиненной.

Выполняемые действия

1. Открыть форму **ФСтудентыСПодч** в режиме **КОНСТРУКТОРА**.

2. Удерживая левую кнопку мыши, перетащить имя формы **ФСтипДляПодч** из области переходов в нужное место области данных

формы **ФСтудентыСПодч.** На форме появится (очертится) образ (область) подчиненной формы, называемый элементом управления подчиненной формы.

3. Сохранить форму с именем **ФСтудентыСПодч.**

4. Связь главной формы с подчиненной установилась автоматически по одинаковому имени полей связи (**НомерЗачКн**). Чтобы убедиться в этом, следует:

- выделить область подчиненной формы (чтобы маркеры находились на границе области);
- вызвать свойства правой клавишей мыши, активизировать вкладку *Данные*;
- убедиться, что автоматически установились значения свойств *Основные поля* (**НомерЗачКн**) и *Подчиненные поля* (**НомерЗачКн**).

Окно свойств подчиненной формы приведено на рисунке 5.10.

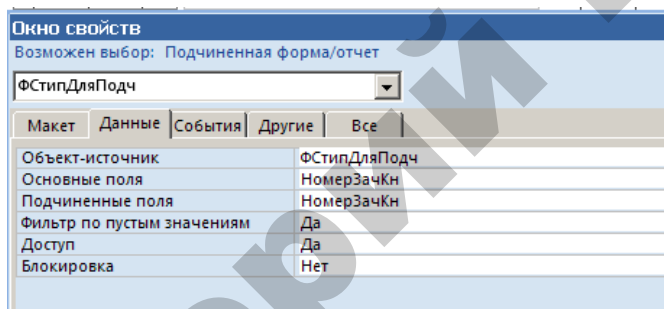


Рис. 5.10. Окно свойств подчиненной формы

Сложная форма в режиме КОНСТРУКТОРА приведена на рисунке 5.11.

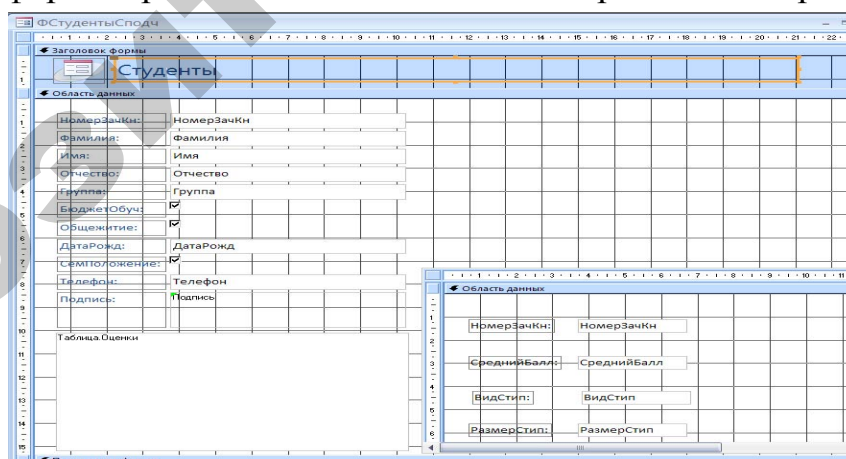


Рис. 5.11. Форма **ФСтудентыСПодч** в режиме КОНСТРУКТОРА

Задание 3. Создать форму в виде сводной таблицы, содержащей информацию о сдаче экзаменов студентами всех групп по всем дисциплинам с расчетом среднего балла по каждой дисциплине в каждой группе и по всем группам.


Выполняемые действия

1. В области переходов выделить запрос **ЗапУспеваемостьМас**.
2. На вкладке *Создание* ленты меню в группе *Формы* в пункте *Другие формы* нажать кнопку *Сводная таблица*.
3. В область полей строк перетащить поле **НаименГруппы**.
4. В область полей столбцов перетащить поле **НаименДисциплины**.
5. В область полей итогов или деталей перетащить поля **Фамилия** и **Оценка**.
6. Выделив столбец оценок любой группы по любой дисциплине в образовавшейся таблице, вызвав контекстное меню (правой клавишей мыши) выбрать пункты *Автовычисления-Среднее*. В таблице появятся итоговые области, содержащие значения среднего балла по каждой группе по каждой дисциплине и значения среднего балла по дисциплинам среди всех сдавших (нижняя строка), значения среднего балла по всем дисциплинам по каждой группе (крайний столбец справа).
7. Нажимая кнопки скрытия, открытия деталей (+,-), можно менять общий вид сводной таблицы. Выполните разные варианты.
8. Вызвав контекстное меню (правой клавишей мыши) при выделенном столбце заголовков (крайний столбец слева), можно менять вид и содержание итоговых полей и скрывать их. Выполните разные варианты.

Создание кнопки в форме

Задание 4. Создать кнопочную форму, содержащую кнопки для открытия двух созданных ранее форм.

Выполняемые действия

1. Нажать кнопку *Конструктор форм* на вкладке **Создание** ленты меню в группе *Формы*. Откроется окно конструктора форм, содержащее пустую область данных формы.
2. Нажать кнопку  Для добавления заголовка и примечания формы. Вызвать окно свойств заголовка (правой клавишей мыши), на вкладке *Все* задать *Цвет фона* – **Светлый фон заголовка**.
3. В заголовок ввести надпись «Открытие форм». Открыть окно свойств надписи (с помощью правой клавиши мыши при выделенной надписи). Задать следующие значения свойств:



Цвет фона – **желтый**;

Размер шрифта – **18**;

Выравнивание текста – По центру;

Цвет текста – Темный текст.

4. Создать кнопку для открытия формы **ФЭкзаменыСПодч**.

При нажатой кнопке  [использовать мастер] переместить знак  с панели элементов в область размещения кнопки в форме. Запустится мастер СОЗДАНИЕ КНОПОК.

Далее выполнить следующие действия.

а) В первом окне мастера выбрать:

- в области КАТЕГОРИИ – категорию *работа с формой*;
- в области ДЕЙСТВИЯ- действие *открытие формы*;
- нажать *далее*.

б) Во втором окне мастера выбрать форму **ФЭкзаменыСПодч**, которая откроется при нажатии создаваемой кнопки. Нажать *Далее*.

в) В третьем окне оставить положение переключателя *Открыть форму и показать все записи*. Нажать *Далее*.

г) В четвертом окне поставить переключатель в положение ТЕКСТ, в поле ввода, расположенное справа, набрать текст, помещаемый на кнопку: Открыть форму **ФЭкзаменыСПодч**. Нажать *Далее*.

д) В пятом окне задать имя кнопки или оставить предложенное системой. Нажать *Готово*.

5. Создать кнопку для открытия формы **ФСтудентыСПодч** (согласно рекомендациям п. 4).

6. Установить желаемые размеры формы.

7. В примечание формы ввести текст *Формы открываются по нажатию кнопок*.

8. Сохранить форму с именем **Формы**.

9. Выполнить форму, убедиться, что кнопки выполняют свои функции.

Задания для самостоятельного выполнения к лабораторной работе № 7

Создать форму с подчиненной, где в главной форме просматриваются Дисциплины, а в подчиненной – вся информация по всем экзаменам по текущей дисциплине.

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

В разрабатываемом приложении выполнить задания 1–2 темы **Создание форм**.

Продуктивный уровень (оценка 7–9)

В разрабатываемом приложении выполнить задания 1–4 темы **Создание форм**.

Творческий уровень (оценка 10)

В индивидуальной базе данных создать все виды форм, рассмотренных в лабораторной работе № 7. Кроме того создать форму в виде сводной таблицы.

ЛАБОРАТОРНАЯ РАБОТА № 8



ПРОЕКТИРОВАНИЕ ОТЧЕТОВ

Цель работы: научиться создавать отчеты с несколькими уровнями группировки данных и разнообразным оформлением, необходимым для формирования выходной информации информационных систем.

Создание отчета с помощью средства «Отчет»

Задание 1. Создать отчет, отражающий список дисциплин с указанием количества часов по каждой дисциплине.


Выполняемые действия


1. В области переходов выделить таблицу **Дисциплины**.
2. На вкладке **Создание** в группе **Отчеты** нажать кнопку **Отчет** . Приложение Access создаст отчет и отобразит его в режиме макета.
3. Просмотреть отчет с помощью кнопки .
4. Сохранить отчет с именем **ОтчетДисциплины**.

Создание отчета с помощью КОНСТРУКТОРА

Задание 2. Создать отчет, отражающий список преподавателей по должностям с подсчетом их количества по каждой должности. Так же рассчитать общее количество преподавателей.

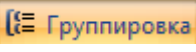
Выполняемые действия

1. Нажать кнопку  на вкладке **Создание** ленты меню в группе **Отчеты**. Откроется окно конструктора отчетов, содержащее пустую область данных отчета и области нижнего и верхнего колонтитулов.

2. Нажать кнопку  и из открывшегося списка полей перенести в область данных отчета поля **КодДолжности**, **НаименДолжности** (из таблицы **Должности**), **ТабНомПреп**, **Фамилия**, **ПедСтаж** (из таблицы **Преподаватели**) двойным щелчком клавишей мыши по каждому полю. В области данных отчета появятся поля связи (правый столбец) вместе с присоединенными надписями (левый столбец).

3. Маркировать надписи полей (удерживая клавишу Shift), с помощью правой кнопки мыши “вырезать их” из области данных и вставить в верхний колонтитул.

4. Разместить поля связи в строку и соответственно присоединенные надписи в верхнем колонтитуле. Просмотреть отчет при помощи кнопки **Режим**.

5. Выполнить группировку строк отчета по полю **КодДолжности**. Для этого нажать кнопку , в нижней части экрана в области **Группировка, сортировка и итоги** выбрать поле группировки **КодДолжности**, далее нажать вкладку **Больше**. В указанной области появится дополнительная информация для выбора. Выбрать значения **С разделом заголовка**, **С разделом примечания**, **С итогами**. Область **Группировка, сортировка и итоги** приведена на рисунке 5.12.

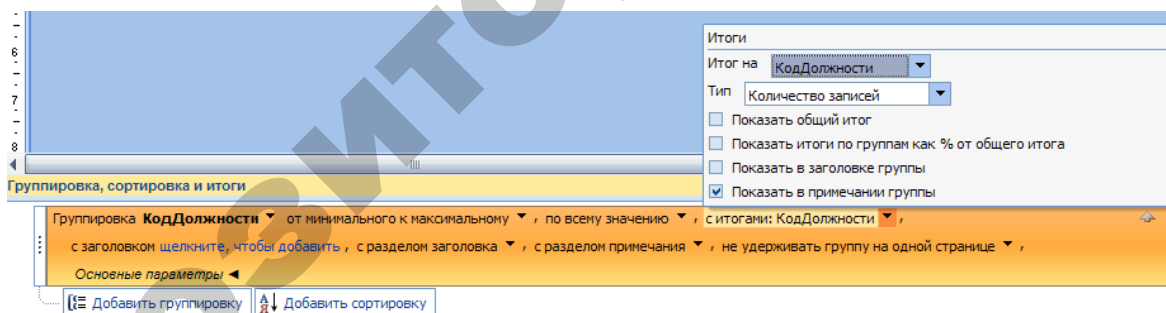



Рис. 5.12. Области Группировка, сортировка и итоги

Замечание. Если необходимо добавить группировку по второму полю, то следует нажать соответствующую кнопку в области **Группировка, сортировка и итоги**.

В конструкторе отчетов появятся области **Заголовок группы** и **Примечание группы** с формулой расчета количества записей группы (то есть количества преподавателей в данной должности), которую следует переместить в правую часть области. Этот расчет можно выполнить и создав в области **Примечание группы** вычисляемое поле(**abl**) с формулой **=Count(Фамилия)**.

6. В области **Заголовок группы** и **Примечание группы** следует еще вставить поле **НаименДолжности** для наглядности.

7. С помощью кнопки  добавить в отчет области *Заголовок отчета* и *Примечание отчета*. В заголовок ввести надпись **ОТЧЕТ ПО ДОЛЖНОСТЯМ**. В примечание – скопировать из примечания группы вычисляемое поле для подсчета количества преподавателей.

Замечание. Обратите внимание, что одна и та же формула в области *Примечание группы* считает количество записей в группе (количество преподавателей по профессии), а в области *Примечание отчета* считает общее количество записей в отчете (Общее количество преподавателей).

8. Оформить «шапку» отчета. Для этого:

а) в верхнем колонтитуле унифицировать размеры полей надписей и соединить боковые границы полей;

б) выделить надписи полей в верхнем колонтитуле, щелкнув слева от строки заголовка;


с) вызвав правой клавишей мыши контекстное меню, выровнять поля надписей сверху;

д) снова вызвав контекстное меню, открыть окно свойств и установить значения:

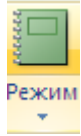
- *Тип границы* – **сплошная**;
- *Ширина границы* – **2 пункта**;
- *Размер шрифта* – **14**.

9. Кроме того, в окне свойств можно изменить тип шрифта, цвет текста, стиль линии сетки и другие свойства заголовков.

10. Добавить элементы оформления в отчет. Нанести разделительные линии в области *Заголовка* и *Примечания группы*.


11. Вставить сегодняшнюю дату в Заголовок отчета, перетащив кнопку  из группы *Элементы управления*.

12. Сохранить отчет с именем **ОтчетДолжности**.

13. Просмотреть отчет с помощью кнопки . Созданный отчет в режиме КОНСТРУКТОРА приведен на рисунке 5.13.

Задание 3. С помощью мастера отчетов создать отчет **ОтчетОценки**, отражающий результаты сдачи экзаменов (оценки студентов) по каждой группе по всем дисциплинам. Рассчитать средний балл каждого студента.

Выполняемые действия

1. На вкладке **Создание** ленты меню в группе *Отчеты* нажать кнопку  **Мастер отчетов**. Далее следует отвечать на вопросы мастера для построения нужного отчета.

2. В первом окне СОЗДАНИЕ ОТЧЕТОВ следует выбрать нужные поля из соответствующих таблиц (кнопкой **>**): **НаименГруппы** (из таблицы **Группы**), **Фамилия**, **Имя**, **Отчество** (из таблицы **Студенты**), **НаименДисциплины** (из таблицы **Дисциплины**), **Оценка** (из таблицы **Оценки**). Нажать **Далее**.

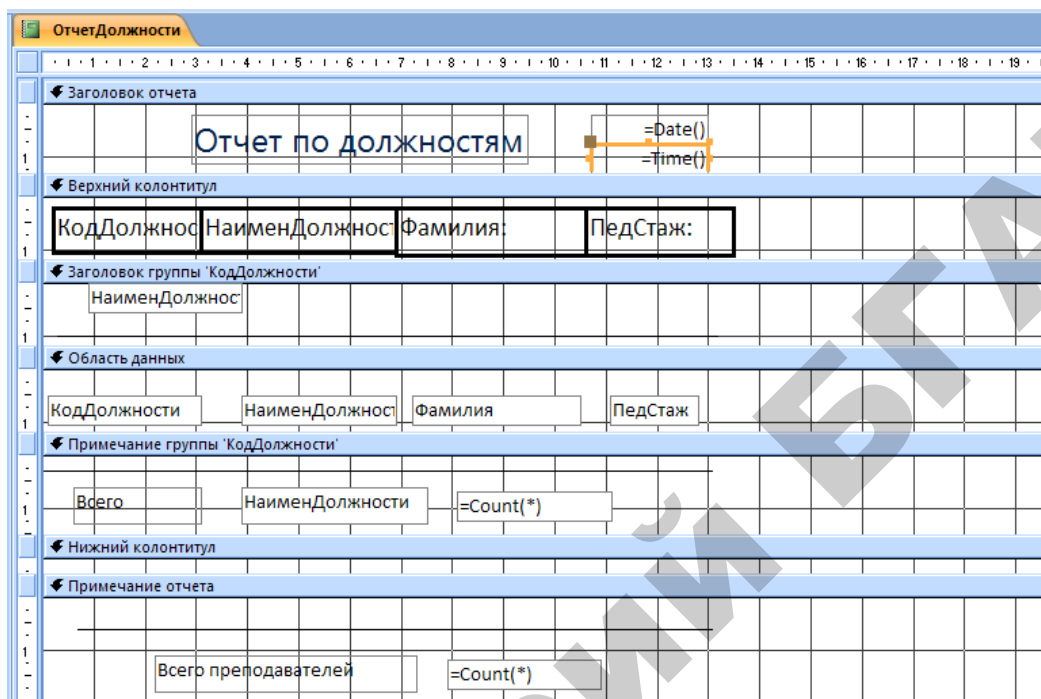


Рис. 5.13. ОтчетДолжности в режиме КОНСТРУКТОРА

3. В следующем окне **Выбрать** вид представления данных **Группы**.
4. В следующем окне можно добавить уровни группировки, а в нашем примере оставить лишь автоматически сформированные.
5. Далее задать первое поле группировки – **Оценка** возростание.
6. Щелкнуть клавишей мыши по вкладке **Итоги**. В появившемся окне на вопрос *Какие итоговые значения необходимо вычислить?* в строке **Оценка** выбрать функцию **Avg**, нажать **ОК**, нажать **ДАЛЕЕ**.
7. Выбрать макет, например, **Ступенчатый**, и ориентацию – **Книжная**. Флажок *Настроить ширину полей для размещения на одной странице* оставить включенным. Нажать **ДАЛЕЕ**.
8. Выбрать **Стиль**, например, **Стандартный**. Ввести имя отчета **ОтчетОценки**.
9. Оставить значение переключателя **Посмотреть отчет**, нажать **Готово**.
10. Проанализировать созданный отчет. Если он верный, то сохранить с именем **ОтчетОценки**.

Замечание. Структуру и другие свойства отчета модно изменить, войдя в режим **КОНСТРУКТОР**.

Публикация данных в Интернете

Задание 4. Данные отчета **ОтчетОценки** представить для публикации в Интернете (в виде веб-страницы).

Выполняемые действия

1. В области переходов выделить отчет **ОтчетОценки**. Вызвать его контекстное меню правой клавишей мыши и выбрать пункт **Экспорт → Документ HTML**. Откроется первое окно мастера экспорта данных.

2. В окне мастера рядом с полем **Имя файла** нажать кнопку **Обзор** и выбрать папку, куда будет сохраняться создаваемый HTML-документ (желательно указать ту папку, в которой находится база данных **УчебныйПроцесс**). После выбора папки в окне **Сохранение файла** имя и тип файла не менять, нажать кнопку **Сохранить**.

3. После возвращения в окно мастера выделить пункт (поставить «галочку») **Открыть целевой файл после завершения операции экспорта**. Нажать кнопку **ОК**.

4. В следующем окне мастера **Параметры ввода в формате HTML** в пункте **Выберите кодировку для сохранения файла** отметить пункт **По умолчанию**. Остальные пункты не выделяйте. Нажать кнопку **ОК**.

5. Откроется окно интернет-браузера для просмотра данных отчета в виде веб-страницы. Посмотреть результат. Файл с именем **ОтчетОценки.html** находится в папке базы данных **УчебныйПроцесс** и может быть просмотрен через любой интернет-браузер. В последнем окне нажать кнопку **ОК**.

Замечание. Созданная веб-страница является **статической**, обновленные данные из базы **УчебныйПроцесс** не будут автоматически отображаться в созданной веб-странице. Для этого нужно создать новую веб-страницу, проделав заново все действия задания 3.

Задания для самостоятельного выполнения к лабораторной работе № 8

1. Создать отчет, отражающий список преподавателей с указанием стажа.

2. Создать отчет, отражающий расписание сдачи экзаменов в сессию с расчетом количества экзаменов, принимаемых преподавателями каждой должности.

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

В разрабатываемом приложении выполнить задания 1, 3 темы **Создание отчетов**.

Продуктивный уровень (оценка 7–9)

В разрабатываемом приложении выполнить задания 1–3 темы **Создание отчетов**.

Творческий уровень (оценка 10)

В индивидуальной базе данных создать все виды отчетов, рассмотренных в лабораторной работе № 8.

Вопросы для самоконтроля

1. Перечислите виды запросов-действий.
2. Какие виды запросов-действий приводят к изменению значений полей таблицы?
3. Для каких целей предназначен объект «форма»?
4. Каким образом форма может влиять на ход выполнения приложения? Приведите пример.
5. Для каких целей предназначен объект «отчет»?
6. Объясните значение понятия «группировка данных», широко используемое при конструировании отчетов.
7. Перечислите наиболее популярные элементы управления, применяемые в формах и отчетах.
8. Охарактеризуйте разные типы элементов управления.

ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ ПО МОДУЛЮ 5

1. В Access существуют следующие типы запросов-действий:
 - запросы на создание таблицы;
 - запросы на добавление;
 - запросы на сравнение;

- запросы на изменение;
- запросы на удаление;
- запросы на поиск.

2. Допишите запрос на SQL, по которому из таблицы будут выбраны фамилии студентов, имеющих имя «Татьяна».

```
SELECT _____
FROM _____
WHERE (Имя="Татьяна");
```

Студент
Код
Фамилия
Имя
ДатаРождения
Общежитие

3. Какой тип имеет следующий запрос:

```
DELETE FROM Предмет
WHERE (Предмет.НомерСеместра)=3);
```

- удаление;
- обновление;
- перекрестный;
- групповой.

4. Какую подпись будет содержать второй столбец следующего запроса:

```
SELECT Клиенты.Фамилия, Клиенты.Телефон , Клиенты.Адрес
FROM Клиенты;
```

- клиенты;
- фамилия;
- адрес;
- телефон.

5. Какие из перечисленных запросов являются запросами-действиями?

- параметрический;
- вычисляемый;
- запрос на обновление;
- перекрестный;
- запрос на создание таблицы.

6. Для связи с полем таблицы – источника данных формы (отчета) служит:

- присоединенный элемент управления;
- свободный элемент управления;
- вычисляемый элемент управления.

7. Выберите правильный вариант записи формулы в вычисляемом поле формы (отчета)

- =[Kolch/2]
- = Kolch/2

- $=[Kolch]/2$

- $[Kolch]/2$

8. Выберите правильный вариант записи формулы в вычисляемом поле отчета для вычисления итогового количества часов

- $=Sum([Kolch])$

- $=Sum[Kolch]$

- $Sum(Kolch)$

- $=Sum[(Kolch)]$

9. При создании итогов в отчетах для получения итога по группе формула расчета итога помещается в поле:

- примечание отчета;

- примечание группы;

- нижний колонтитул.

10. Какую функцию применить для вставки в отчет текущей даты и времени?

- $Now()$

- $Year()$

- $Month()$

МОДУЛЬ 6

АВТОМАТИЗАЦИЯ ОБРАБОТКИ БАЗЫ ДАННЫХ В СУБД СИСТЕМЫ ОБРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ БАЗ ДАННЫХ

В результате изучения модуля студент должен

знать: средства автоматизации работы с БД в MS Access; основы языка VBA, принципы построения модулей; эволюцию концепций обработки данных, принципы передачи данных по сети, системы удаленной обработки, компьютерные архитектуры *Файл/сервер* и *Клиент/сервер*, основные современные серверные СУБД; технологию проектирования и обработки многопользовательских баз данных;

уметь: характеризовать объекты, методы, события, модули VBA; применять полученные теоретические и практические знания при конструировании макросов и модулей в процессе решения экономических задач и задач управления, конструировать пользовательские источники данных, запросы к серверу базы данных, процедуры экспорта и импорта данных в системах обработки многопользовательских баз данных; применять полученные теоретические и практические знания для постановки и решения задач экономики и управления на предприятии, требующие централизованной организации и обработки информации;

моделировать и конструировать пользовательские источники данных, запросы к серверу базы данных, процедуры экспорта и импорта данных в системах обработки многопользовательских баз данных.

ПРОГРАММИРОВАНИЕ НА VBA (*VISUAL BASIC FOR APPLICATION*)

ВНИМАНИЕ!!! Перед каждым сеансом работы с программами необходимо нажать на кнопку ПАРАМЕТРЫ, расположенную в строке «Предупре-

ждения системы безопасности...», в открывшемся окне «Оповещение системы безопасности» отметить флажок «Включить это содержимое» и нажать на кнопку ОК.

Алгоритм обработки данных в программах СУБД Access

При использовании объекта **Recordset** (*набор записей (англ.)*) для доступа к данным таблицы в буфере размещаются строки данных. Из набора для обработки можно выбрать только одну строку, которая называется *текущей записью*. Она является единственной доступной записью для модификации или извлечения из нее данных. После того, как запись стала *текущей*, необходимо с помощью метода **Edit** (*редактировать (англ.)*) скопировать ее в буфер копии, создаваемый ядром СУБД Jet для редактирования. Теперь данные *текущей записи* доступны для модификации. Для сохранения изменений буфера копии в наборе записей используется метод **Update** (*обновить (англ.)*).

Для **физического** перемещения в наборе записей применяются методы **Move**, которые позволяют сделать *текущей* нужную запись. Если необходим полный перебор всех записей набора, можно использовать метод **MoveNext** (переход к следующей записи набора). Для определения конца набора перед каждой итерацией цикла проверяется значение характеристики **EOF** (*end of file (англ.)* – конец файла). Если она равна **TRUE** (*истина (англ.)*), то цикл завершается.

Для **логического** перемещения в наборе записей применяется метод **Seek** (*искать (англ.)*). Поиск осуществляется в наборе записей по номеру записи в индексированном поле.

Обобщенная блок-схема алгоритма обработки данных в программах СУБД Access представлена на рисунке 6.1.

Создание процедуры обработки события «Нажатие на кнопку»

Для создания программ на языке VBA можно использовать модули классов, связанные с формой. Такие модули хранятся вместе с формами.

1. Создать простую форму с именем **Программы** в режиме КОНСТРУКТОРА без указания источника данных: **Создание – Конструктор форм**.

2. При отключенной кнопке ИСПОЛЬЗОВАТЬ МАСТЕРА создать кнопку в области данных формы (по умолчанию имя кнопки будет **Кнопка0**).

3. Щелкнуть на форме **Программы** по кнопке **Кнопка0** правой клавишей мыши и выбрать в контекстном меню пункт **Обработка событий** и далее – **Программы**. Откроется окно для ввода текста программы – редактор VBA.

4. Набрать текст программы, сохранить ее, выполнив команду **File – Save <имяБазы>**. Закрыть редактор VBA.

5. Для выполнения программы открыть форму **Программы** в режиме ФОРМЫ, нажать на кнопку **Кнопка0**.

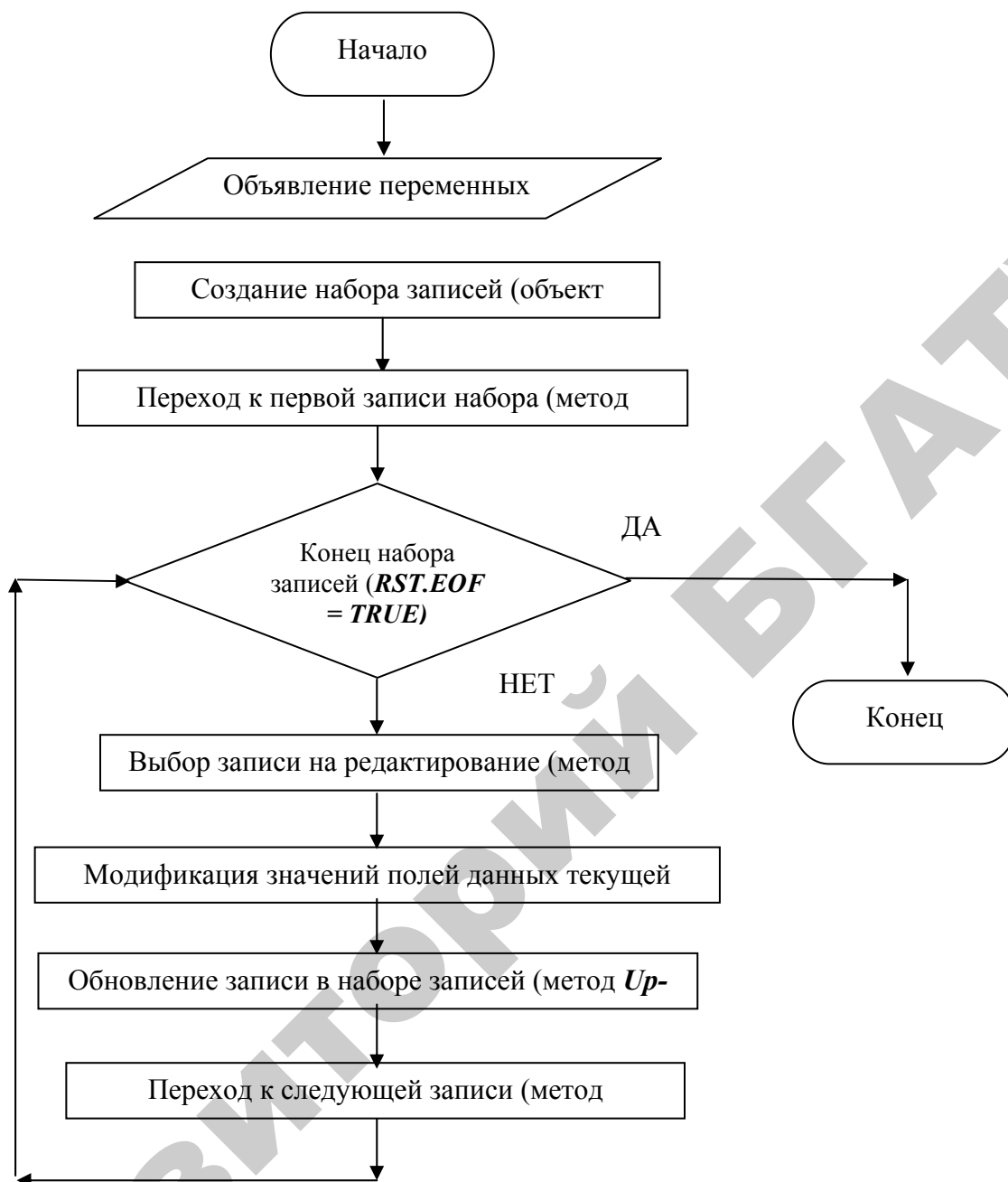


Рис. 6.1. Обобщенная схема алгоритма обработки данных в программах СУБД Access

Замечание.

1. Для каждой программы нужно создавать новую кнопку на форме. Это можно сделать на уже существующей форме – тогда имя новой кнопки по умолчанию будет **Кнопка1** и т. д., а можно создать новую форму и на ней кнопку. В первом случае все программы будут храниться в одном модуле класса, связанном с формой.
2. Чтобы просмотреть уже созданную программу, нужно открыть созданную форму в режиме КОНСТРУКТОРА, щелкнуть по кнопке **Кнопка0** правой клавишей мыши и выбрать в контекстном меню **Обработка событий**.

Отладка программ

Если в программе нет ошибок, то после нажатия на кнопку **Кнопка0** программа будет выполнена. Никаких сообщений при этом выводиться не будет.

Если же в программе есть ошибки, то ее выполнение будет остановлено на первой же ошибке. Сообщение о ней будет выведено на экран, нужно его прочитать и запомнить, нажать на кнопку ОК в окне. Управление будет передано в редактор VBA. Ошибочный оператор будет выделен красным шрифтом. Исправив ошибку, выполните команду **Run – Reset**, сохраните обновленную программу, закройте редактор VBA. Переходите к выполнению п. 5. Повторяйте приведенные действия до тех пор, пока не устраните все ошибки в программе.

СОВРЕМЕННЫЕ АРХИТЕКТУРЫ СУБД

По функциональным возможностям СУБД подразделяются на персональные (настольные) и многопользовательские (серверные).

Персональные СУБД предназначены для создания и обработки недорогих однопользовательских баз данных. Однако, в настоящее время появились сетевые конфигурации однопользовательских СУБД, позволяющие работать нескольким пользователям с одной базой данных. Основной принцип работы таких систем – это обработка данных внутри пользовательского приложения, т. е. данные по запросу пользователя загружаются из хранилища данных на компьютер пользователя, где они и обрабатываются. В этом случае реализуется компьютерная архитектура **файл-сервер**. В качестве сервера используется дополнительный винчестер, к которому могут обращаться несколько пользователей.

Серверные же СУБД изначально ориентированы на многопользовательскую обработку базы данных. Они базируются на компьютерной архитектуре **Клиент/Сервер**, которая обеспечивает наиболее эффективную работу с централизованной БД. Централизация хранения и обработки данных является базовым принципом этой компьютерной архитектуры. В этом случае сервер – это ЭВМ, на которой выполняется программа, предоставляющая определенные услуги другим программам, называемым клиентами.

Многопользовательские (серверные) СУБД включают в себя сервер СУБД и клиентскую часть. Такие СУБД, как правило, могут работать в неоднородной вычислительной среде (с разными типами ЭВМ и ОС). К многопользовательским СУБД относятся: Informix, Oracle, MS SQL Server и др.

На сервере сети размещается БД и устанавливается мощная серверная СУБД – *сервер БД*.

На компьютере-клиенте клиентское приложение (приложение-клиент) формирует запрос к БД. Серверная СУБД обеспечивает интерпретацию запроса, его выполнение, формирование результата запроса и пересылку его (результата) по сети на клиентский компьютер. Клиентский компьютер интерпретирует его необходимым образом и представляет пользователю. Клиентское приложение может также посылать запрос на обновление БД, и серверная СУБД внесет необходимые изменения в БД.

Сервер баз данных осуществляет целый комплекс действий по управлению данными. *Основными функциями сервера* являются:

1) прием запросов от клиента, их интерпретация, выполнение пользовательских запросов на выбор и модификацию данных и метаданных, получаемых от клиентских приложений, функционирующих на персональных компьютерах;

2) поддержка ссылочной целостности данных согласно определенным в базе данных правилам;

3) обеспечение авторизованного доступа к данным в соответствии с правом пользователя, т. е. обеспечение авторизованного доступа к данным на основе проверки прав и привилегий пользователей;

4) хранение данных и их резервное копирование;

5) протоколирование операций и ведение журнала транзакций.

Функции клиента следующие:

1) посылка запросов к СУБД на сервер;

2) интерпретация и представление полученных результатов запроса;

3) реализация пользовательского интерфейса.

Подводя итоги, отметим, что информационные системы, использующие архитектуру «клиент – сервер», обладают рядом преимуществ по сравнению с их аналогами, созданными на основе сетевых версий настольных СУБД.

СУБД MICROSOFT SQL SERVER

Microsoft SQL Server представляет собой СУБД, обеспечивающую создание информационных систем с архитектурой «клиент – сервер». Эта СУБД

поддерживает тиражирование данных, параллельную обработку, создание и обработку больших баз данных на недорогих аппаратных платформах, отличается простотой управления и использования, а также обеспечивает тесную интеграцию баз данных SQL Server в Web.

Инструменты SQL Server

Основными инструментами администрирования SQL Server 2000 являются следующие:

- SQL Server Enterprise Manager;
- SQL Server Service Manager;
- SQL Server Profiler;
- Query Analyzer;
- Upgrade Wizard (Мастер обновления);
- Import and Export Data (Мастер экспорта/импорта);
- утилиты Client Network Utility и Server Network Utility;
- утилиты командной строки;
- специальные Мастера (Wizards).

Большая часть административных задач SQL Server может быть выполнена тремя различными способами: с использованием средств Transact-SQL (требует высокой квалификации, но позволяет решать наиболее сложные задачи), с помощью графического интерфейса SQL Server Enterprise Manager (высокая функциональность наряду с простотой использования) и использованием мастера (простота использования при ограниченном круге решаемых задач).

SQL Server Enterprise Manager является важнейшим инструментом, позволяющим выполнять следующие действия:

- управлять системой безопасности;
- создавать БД и ее объекты;
- создавать и восстанавливать резервные копии;
- конфигурировать подсистему репликации;
- управлять параметрами работы служб SQL Server 2000;
- управлять подсистемой автоматизации;
- запускать, останавливать и приостанавливать службы;
- конфигурировать связанные и удаленные серверы;
- создавать, управлять и выполнять пакеты службы трансформации данных DTS (Data Transformation Services).

SQL Server Service Manager предоставляет пользователю удобный механизм запуска, остановки и приостановки служб SQL Server 2000. Кроме того, утилита позволяет разрешать или запрещать автоматический запуск той или иной службы при загрузке операционной системы.

SQL Server Profiler является графическим инструментом администратора для наблюдения за работой SQL Server 2000. При выполнении пользовательских запросов, хранимых процедур, подключения и отключения к серверу и других действий программы ядра сервера записывают в системные таблицы информацию о результатах выполнения этих действий. Утилита SQL Server Profiler с помощью специальных хранимых процедур выбирает эту информацию и представляет ее в удобном для анализа виде. Мониторинг сервера заключается в наблюдении за *событиями* (events), каждое из которых является минимальным контролируемым объемом работы ядра сервера. Каждое событие принадлежит какому-либо *классу событий* (event classes), которые, в свою очередь, для удобства объединены в двенадцать *категорий* (category).

Query Analyzer предназначен для выполнения запросов и анализа их результатов. По важности этот инструмент сопоставим с SQL Server Enterprise Manager. В этой утилите появился обозреватель объектов (Object Browser), с помощью которого можно просмотреть список всех объектов, имеющихся в любой БД, а также перечень встроенных функций и системных типов данных.

Более существенным нововведением является возможность трассировки выполнения хранимых процедур.

Мастера предназначены для автоматизации и упрощения решения различных административных задач. Большинство мастеров обладает достаточно ограниченными возможностями. Однако, некоторые мастера, например, мастер конфигурирования подсистемы репликации таковым не является. Примеры мастеров: Backup Wizard (создание резервных копий базы данных), Create Database Wizard (создание БД), Create Diagram Wizard (создание диаграммы БД), Create Index Wizard (создание индекса), Create Login Wizard (создание учетной записи SQL Server для пользователя), Create Stored Procedure Wizard (создание хранимой процедуры), Create View Wizard (создание представления), Index Tuning (оптимизация индексов).

ЛАБОРАТОРНАЯ РАБОТА 9

СОЗДАНИЕ ПРОГРАММ НА ЯЗЫКЕ VBA

Цель работы: изучить основные приемы программирования на языке VBA и научиться писать простые программы обработки данных в СУБД Access.

Задание 1. Написать программу на языке VBA, выполняющую следующую задачу:

Время начала экзаменов переносится на полчаса раньше запланированного времени. Составить процедуру корректировки расписания (в каждой записи таблицы **Экзамены1** в поле **ВремяНач** значение времени уменьшить на 30 минут).

Таблица **Экзамены1** является копией таблицы **Экзамены**.

Блок-схема алгоритма выполнения задания 1 представлена на рисунке 6.2.

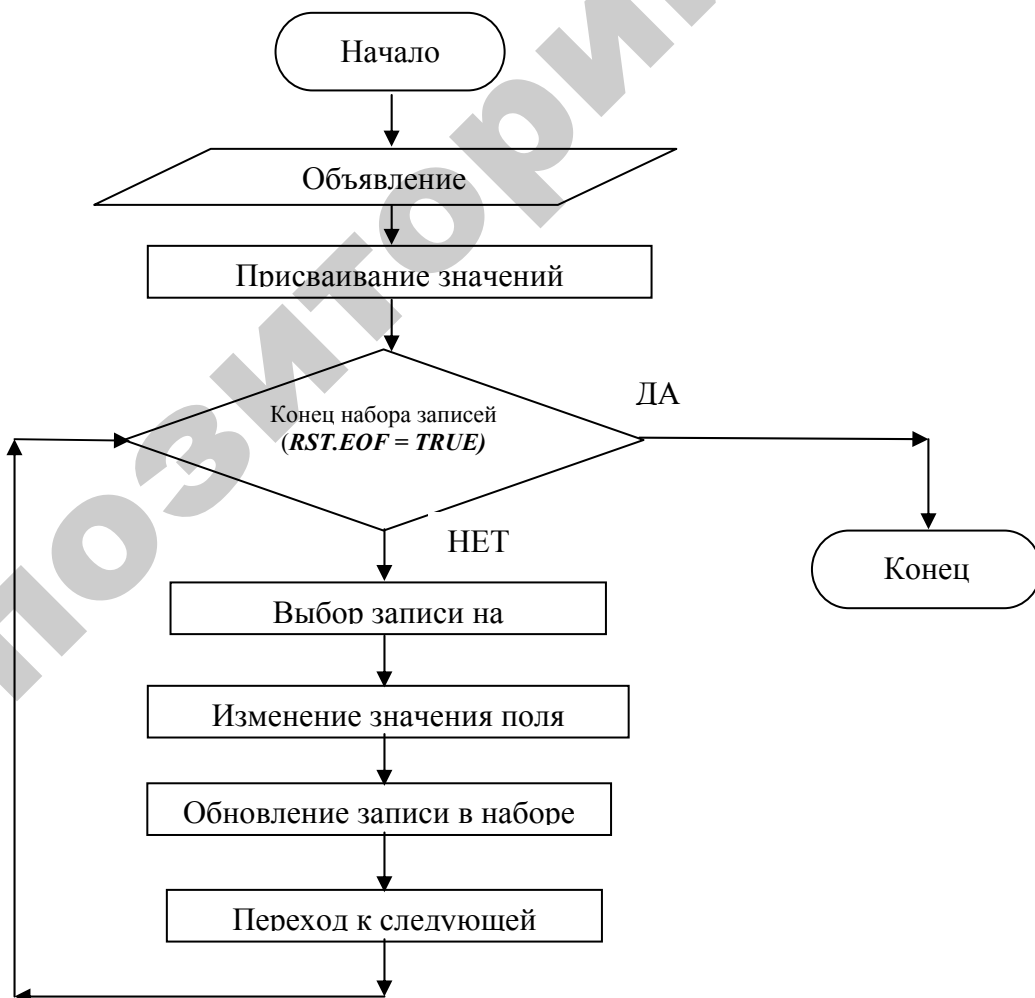


Рис. 6.2. Схема алгоритма задания 1

Текст программы представлен на рисунке 6.3.

```
Private Sub Кнопка0_Click()  
Dim db As Database, rst As Recordset  
Set db = DBEngine(0)(0)  
Set rst = db.OpenRecordset("Экзамены1", dbOpenTable)  
Do Until rst.EOF  
rst.Edit  
rst!ВремяНач = DateAdd("n", -30, rst!ВремяНач)  
rst.Update  
rst.MoveNext  
Loop  
End Sub
```

Рис. 6.3. Текст программы задания 1

Поясним действие каждого оператора языка VBA:

Private Sub Кнопка0_Click() – заголовок подпрограммы (ключевое слово *Sub*) с именем *Кнопка0_Click()*, имя указывает на то, что подпрограмма будет вызываться при нажатии (*Click*) на Кнопку0;

Dim db As Database, rst As Recordset – объявление переменной *db*, имеющей тип данных – база данных (*Database*) и переменной *rst*, имеющей тип данных – указатель на набор записей;

Set db = DBEngine(0)(0) – переменной *db* присваивается указатель на открытую базу данных;

Set rst = db.OpenRecordset("Экзамены1", dbOpenTable) – переменной *rst* присваивается указатель на набор записей таблицы **Экзамены1**;

Do Until rst.EOF – оператор начала цикла *Do Until*, цикл будет выполняться до тех пор, пока указатель на набор записей *rst* не выйдет за нижнюю границу набора записей (пока не закончатся записи таблицы **Экзамены1**);

rst.Edit – текущая запись таблицы **Экзамены1** копируется в буфер для редактирования;

rst!ВремяНач = DateAdd("n", -30, rst!ВремяНач) – значение поля *ВремяНач* текущей записи уменьшается на 30 минут, применяется стандартная функция СУБД Access;

rst.Update – сохранение измененной записи в наборе записей;

rst.MoveNext – переход к следующей записи набора записей таблицы **Экзамены1**;

Loop – оператор завершения цикла *Do Until*;

End Sub – конец процедуры.

Задание 2. Написать программу на языке VBA, выполняющую следующую задачу.

В таблицу **Экзамены1** в режиме КОНСТРУКТОРА добавить новое поле **ЧасыДисциплинаГруппа** (тип данных поля – *числовой*, размер поля – *одинарное с плавающей точкой*). Создать модуль VBA для вычисления для каждого экзамена значения нового поля по формуле:

КоличествоЧасов (из таблицы **Дисциплины**) * **Коэффициент** (из таблицы **Группы**).

Пояснение. Значение поля **КоличествоЧасов** из таблицы **Дисциплины**, соответствующее дисциплине, по которой проводится экзамен, умножить на значение поля **Коэффициент** из таблицы **Группы**, соответствующее группе, у которой проводится экзамен (рис. 6.4).

Так для первой строки таблицы **Экзамены1** (рис. 6.4) номер группы, сдающей экзамен равен **6**, а номер дисциплины, по которой сдается экзамен, – **7**. Нужно найти соответствующие записи в таблицах **Группы** и **Дисциплины**, выбрать значения полей **Коэффициент** из таблицы **Группы** и **КоличествоЧасов** из таблицы **Дисциплины**, перемножить их, и занести результат в первую строку таблицы **Экзамены1**. Аналогичные действия нужно провести для каждой строки таблицы **Экзамены1**, так как значение нового поля должно вычисляться для каждого экзамена.

Группа	Дисциплина	Преподаватель	Дата	ВремяН	ВремяКон	Аудитори	ЧасыДисциплинаГруппа
1	7	2	05.06.2010	9:30	15:30	135	19,2
1	3	3	11.06.2010	7:45	13:30	241	18
1	1	1	05.06.2010	7:45	12:30	135	81,60001
1	3	3	10.06.2010	9:30	14:00	133	72
1	4	3	14.06.2010	7:45	12:30	315a	61,2
1	5	4	26.06.2010	7:45	14:00	133	102
1	6	2	30.06.2010	9:30	14:30	135	57,6
2	2	1	04.06.2010	8:00	15:00	137	20,4
2	3	3	09.06.2010	9:30	14:30	241	18
2	4	3	18.06.2010	8:00	12:20	241	15,3
2	5	4	24.07.2010	9:30	15:30	315a	25,5
2	6	2	29.06.2010	11:30	16:30	135	14,4

КодДисциплины	НаименДисциплины	КоличествоЧасов
1	КИТ	68
2	ИТ	68
3	Высшая математика	60
4	ЭММ	51
5	Макроэкономика	85
6	История Беларуси	48
7	Всемирная история	64

КодГруппы	НаименГрупп	ФормаОбуч	Кoeffицие
1	73эи	Дневная	1,2
2	68им	Заочная	0,3
3	72эи	Дневная	1,4
4	5тк	Заочная	0,3
5	64м	Вечерняя	1
6	Бэво	Заочная	0,3

$$64 * 0,3 = 19,2$$

Рис. 6.4. Пояснение к заданию 2

Текст программы представлен на рисунке 6.5.

```

Private Sub Кнопка1_Click()
Dim db As Database, rst As Recordset, rst1 As Recordset
Set db = DBEngine(0)(0)
Set rst = db.OpenRecordset("Экзамены1", dbOpenTable)
Set rst1 = db.OpenRecordset("Дисциплины", dbOpenTable)
Set rst2 = db.OpenRecordset("Группы", dbOpenTable)
rst.MoveFirst
Do Until rst.EOF
rst.Edit
rst1.Index = "Primarykey"
rst1.Seek "=", rst!Дисциплина
rst1.Edit
rst2.Index = "Primarykey"
rst2.Seek "=", rst!Группа
rst2.Edit
rst!ЧасыДисциплинаГруппа = rst1!КоличествоЧасов * rst2!Кoeffициент
rst.Update
rst.MoveNext
Loop
End Sub

```

Рис. 6.5. Текст программы задания 2

Блок-схема алгоритма выполнения задания 2 представлена на рисунке 6.6.

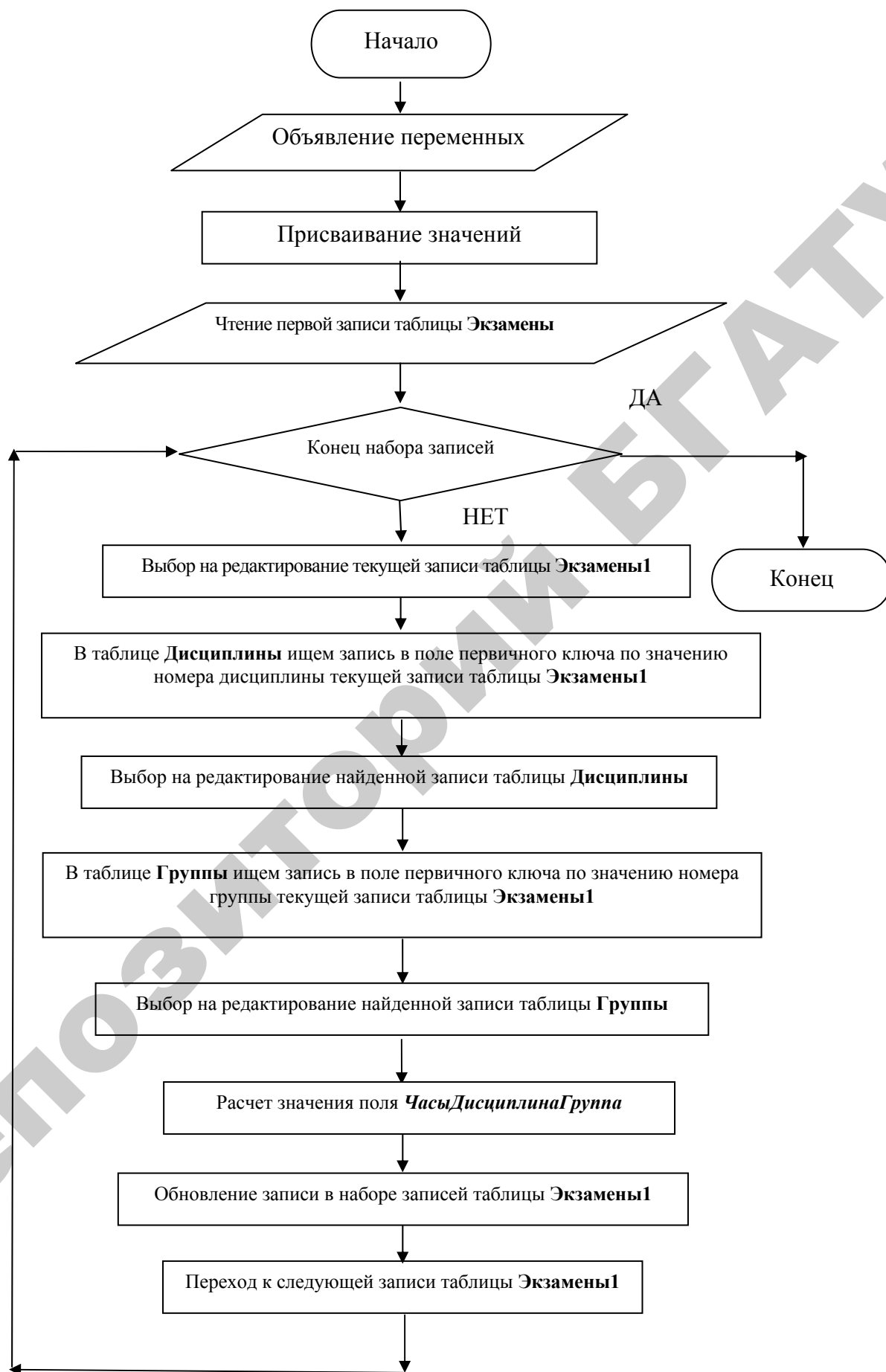


Рис. 6.6. Схема алгоритма задания 2

Поясним действие тех операторов языка VBA, которые не описаны в предыдущем примере:

```
Private Sub Кнопка1_Click()  
Dim db As Database, rst As Recordset, rst1 As Recordset  
Set db = DBEngine(0)(0)  
Set rst = db.OpenRecordset("Экзамен1", dbOpenTable)  
Set rst1 = db.OpenRecordset("Дисциплины", dbOpenTable)  
Set rst2 = db.OpenRecordset("Группы", dbOpenTable)  
rst.MoveFirst – переход к первой записи набора записей таблицы Экзамены1;  
Do Until rst.EOF  
rst.Edit  
rst1.Index = "PrimaryKey" – активизация индекса таблицы Дисциплины по полю  
первичного ключа;  
rst1.Seek "=", rst!Дисциплина – поиск в таблице Дисциплины записи с номером в  
поле первичного ключа равным номеру дисциплины текущей записи таблицы Экзамены1;  
rst1.Edit  
rst2.Index = "PrimaryKey" – активизация индекса таблицы Группы по полю  
первичного ключа;  
rst2.Seek "=", rst!Группа  
rst2.Edit  
rst!ЧасыДисциплинаГруппа = rst1!КоличествоЧасов * rst2!Коэффициент –  
расчет значения поля ЧасыДисциплинаГруппа;  
rst.Update  
rst.MoveNext  
Loop  
End Sub.
```

Задание 3. Написать программу на языке VBA, выполняющую следующую задачу:

Одним из оснований снижения оплаты за обучение является наличие у студента оценок «девять» и «десять» в количестве не менее 50 % от всех оценок сессии. Добавить в таблицу **СтудНазнСтип** новое поле **ОтличОценки** (тип данных поля – *числовой*, размер поля – *целое*). Создать модуль VBA для вычисления значения нового поля **ОтличОценки** для каждого студента, которое соответствует количеству оценок «девять» и «десять» в таблице **Оценки**. Пояснения к заданию 3 представлены на рисунке 6.7.

Так как в таблице **Оценки** в каждой строке записана оценка одного студента по одному предмету, то для того, чтобы подсчитать, сколько оценок «девять» и «десять» получил каждый студент во время сессии, нужно **отсортировать** эту таблицу по полю **Студент**. В программировании это можно сделать, сформировав индекс по нужному полю.

Дисциплина	Студент	Оценка
КИТ	86015	9
Макроэкономика	86015	10
Высшая математика	86020	8
Всемирная история	86020	9
КИТ	92001	9
Высшая математика	92001	8
ЭММ	92001	9
Макроэкономика	92001	10
История Беларуси	92001	7
КИТ	92002	7
Высшая математика	92002	6
ЭММ	92002	8
Макроэкономика	92002	9
История Беларуси	92002	9

Студент с номером зачетки 92001 получил 5 оценок, среди которых 3 оценки «девять» и «десять»

НомерЗачКн	Фамилия	Имя	Отчество	СреднийБа.	ВидСтип	РазмерСтиг	ОтличОцен
86015	Иванов	Сергей	Иванович	5,5	Социальная	82503	2
86020	Иванова	Ирина	Леонидовна	8,5	Учебная+20%	133301	1
92001	Петрова	Анна	Петровна	8,6	Учебная+20%	133301	3
92002	Кисель	Александра	Ивановна	7,8	Учебная+10%	112793	2
92005	Баланович	Алеся	Александровна	9,2	Учебная+50%	153809	5
94026	Корень	Владимир	Эдуардович	8	Учебная+10%	112793	1
94041	Стах	Оксана	Юрьевна	7	Учебная+10%	112793	0
94048	Степанов	Сергей	Александрович	5,2	Социальная	82503	0

Рис. 6.7. Пояснения к заданию 3

Создание индекса по полю Студент в таблице Оценки

1. Открыть таблицу **Оценки** в режиме КОНСТРУКТОРА, нажать кнопку и в открывшееся окно ввести в столбец *Индекс* наименование индекса **ИндОценки**, в столбце *Имя поля* выбрать из выпадающего списка **Студент**, в столбце *Порядок сортировки* выбрать **По возрастанию**, в нижней части окна Свойства индекса все поля должны иметь значение **НЕТ**, далее закрыть окно, сохранить таблицу.

2. В режиме КОНСТРУКТОРА в таблице **Оценки** выделить поле **Студент**, в *Свойствах поля* в строке *Индексированное поле* выбрать **Да (Совпадения допускаются)**.

3. Убедиться в наличии созданного индекса **ИндОценки**, нажав на кнопку ИНДЕКСЫ (рис. 6.8).

Аналогичным образом создать в таблице **СтудНазнСтип** индекс по полю **НомЗачКн**.

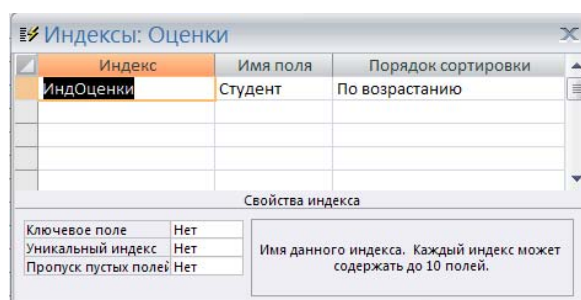


Рис. 6.8. Окно Индексы таблицы Оценки

Блок-схема алгоритма выполнения задания 3 представлена на рисунке 6.9. Детализация заштрихованных блоков представлена на рисунке 6.10.

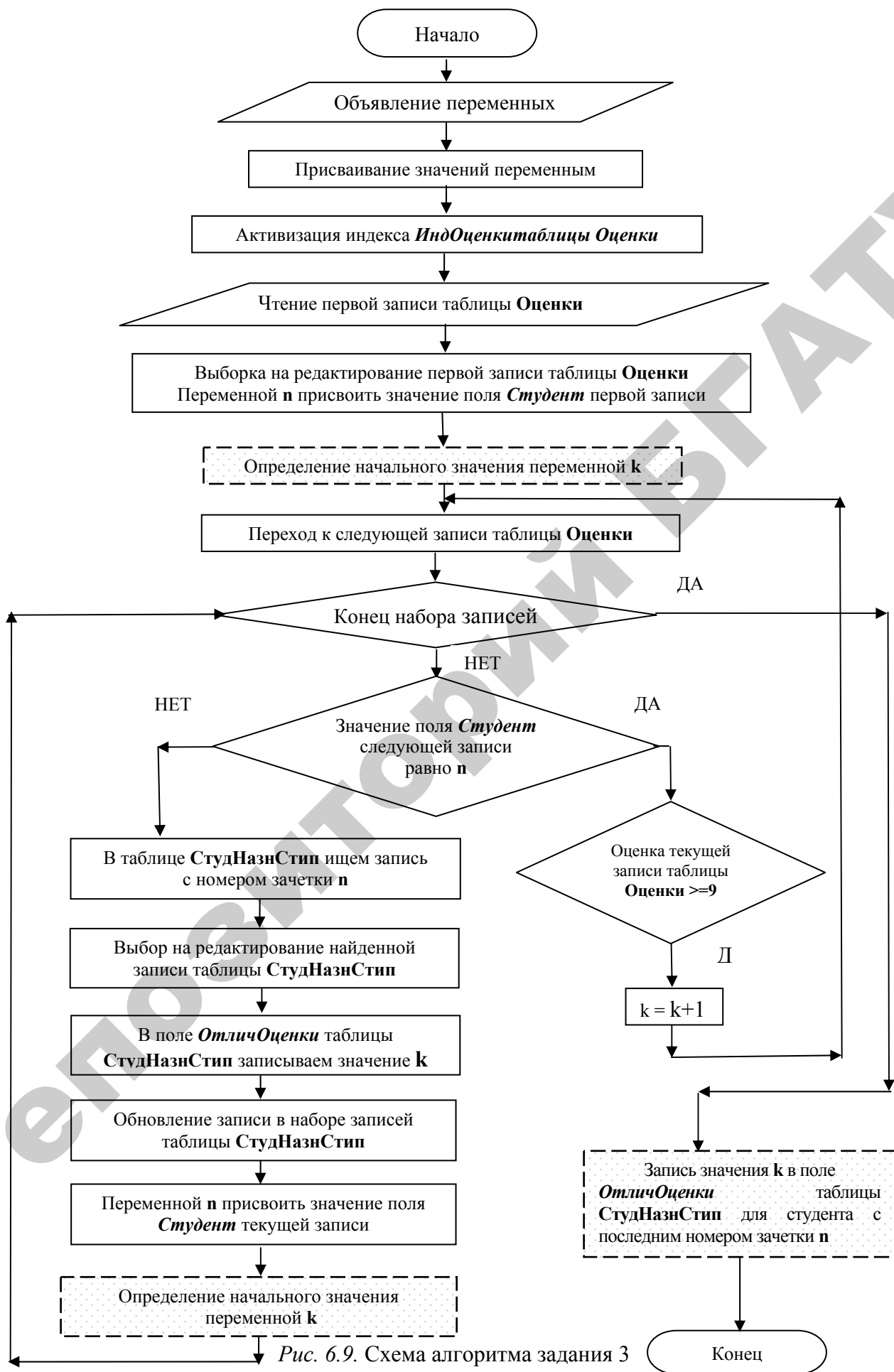


Рис. 6.9. Схема алгоритма задания 3

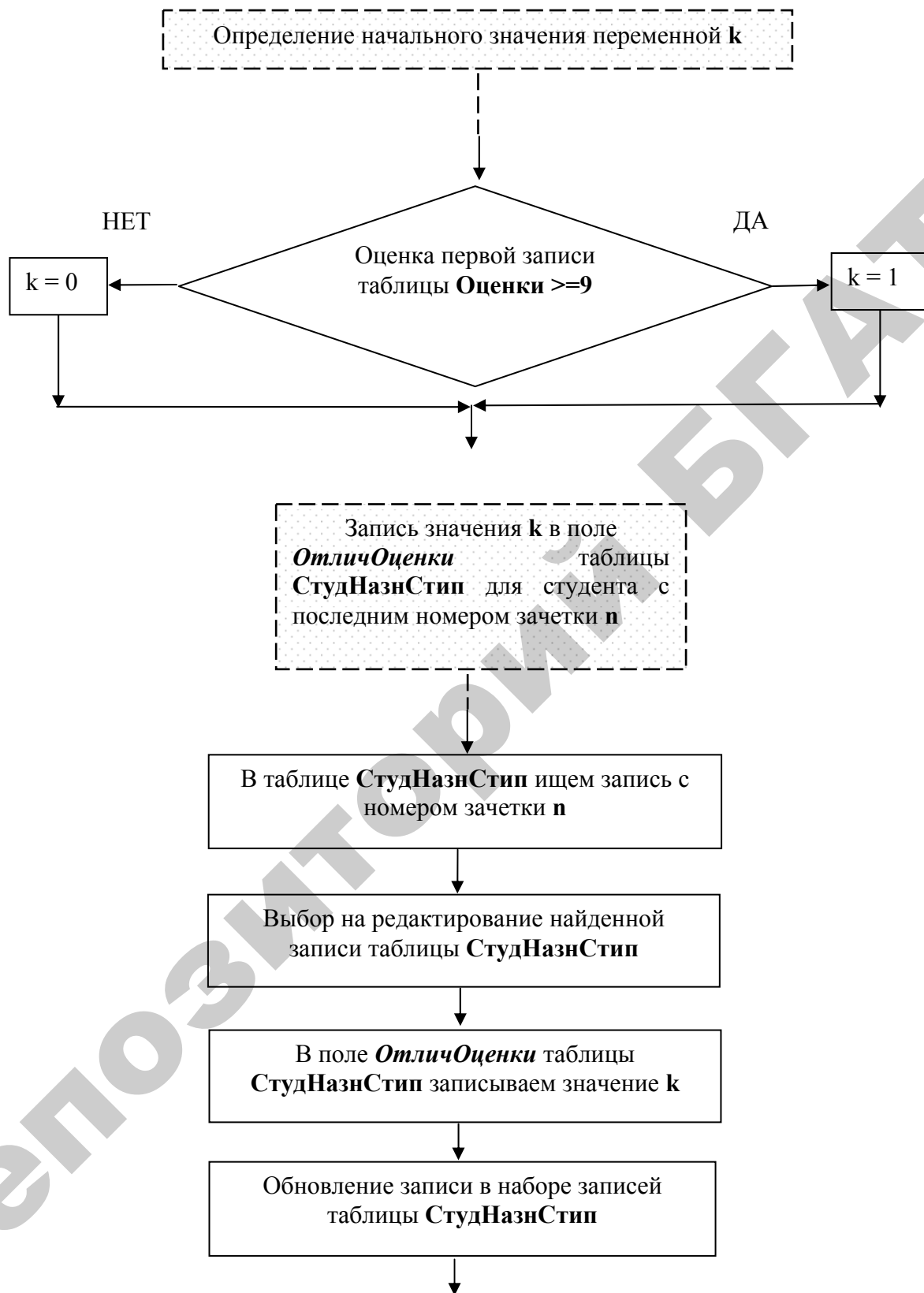


Рис. 6.10. Детализация блоков алгоритма задания 3

Текст программы задания 3 представлен на рисунке 6.11.

```

Private Sub Кнопка2_Click()
Dim db As Database, rst As Recordset, rst1 As Recordset, n, k As Integer
Set db = DBEngine(0) (0)
Set rst = db.OpenRecordset("Оценки", dbOpenTable)
Set rst1 = db.OpenRecordset("СтудНазнСтип ", dbOpenTable)
rst.Index = "ИндОценки"
rst.MoveFirst
rst.Edit
n = rst!Студент
If rst!Оценка >= 9 Then k = 1 Else k = 0
rst.MoveNext
Do Until rst.EOF
If rst!Студент = n Then
    If rst!Оценка >= 9 Then k = k + 1
    Else
        rst1.Index = "НомЗачКн"
        rst1.Seek "=", n
        rst1.Edit
        rst1!ОтличОценки = k
        rst1.Update
        n = rst!Студент
        If rst!Оценка >= 9 Then k = 1 Else k = 0
    End If
rst.MoveNext
Loop
rst1.Index = "НомЗачКн"
rst1.Seek "=", n
rst1.Edit
rst1!ОтличОценки = k
rst1.Update
End Sub

```

Рис. 6.11. Текст программы задания 3

Индивидуальные задания

Репродуктивный уровень (оценка 4–6)

В разрабатываемом приложении выполнить задание 1 темы **Создание модулей VBA**.

Продуктивный уровень (оценка 7–9)

В разрабатываемом приложении выполнить задание 2 темы **Создание модулей VBA**.

Творческий уровень (оценка 10)

В разрабатываемом приложении выполнить задание 3 темы **Создание модулей VBA**.

Замечание. Для студентов, разрабатывающих индивидуальную базу данных, создать модуль **VBA**, вычисляющий промежуточный итог по группе данных. Подобный модуль изучен в задании 3 лабораторной работы 9.

ЛАБОРАТОРНАЯ РАБОТА 10

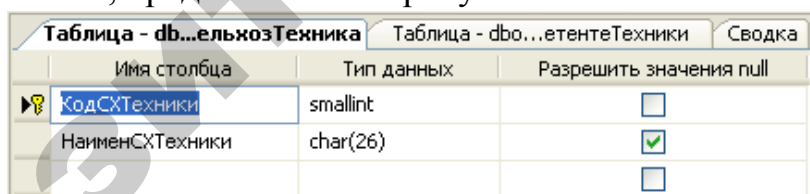
СИСТЕМЫ ОБРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ БАЗ ДАННЫХ

Цель работы: изучить принципы передачи данных по сети, системы удаленной обработки, компьютерные архитектуры *Файл/сервер* и *Клиент/сервер*, основные современные серверные СУБД; технологию проектирования и обработки многопользовательских баз данных.

База данных «СельскоеХозяйство»

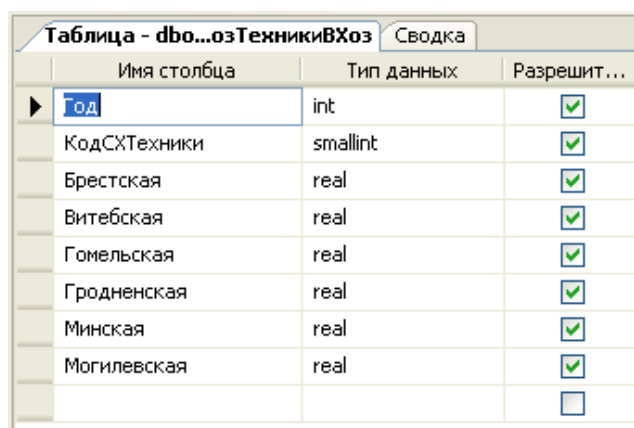
Большим спросом среди работников сельского хозяйства и студентов пользуется такая статистическая информация, как основные показатели развития сельского хозяйства республики, сравнительные статистические данные по районам и т. д. На основе этой информации на сервере сети создана база данных **СельскоеХозяйство**, отражающая статистическую информацию об основных показателях развития сельского хозяйства Республики Беларусь в последние годы.

Структуры записей таблиц базы данных **СельскоеХозяйство**, содержащих сведения о состоянии *материально-технической базы* сельского хозяйства, представлены на рисунках 6.12–6.19.



Имя столбца	Тип данных	Разрешить значения null
КодСХТехники	smallint	<input type="checkbox"/>
НаименСХТехники	char(26)	<input checked="" type="checkbox"/>

Рис. 6.12. Структура записи таблицы ТСельхозТехника



Имя столбца	Тип данных	Разрешит...
Год	int	<input checked="" type="checkbox"/>
КодСХТехники	smallint	<input checked="" type="checkbox"/>
Брестская	real	<input checked="" type="checkbox"/>
Витебская	real	<input checked="" type="checkbox"/>
Гомельская	real	<input checked="" type="checkbox"/>
Гродненская	real	<input checked="" type="checkbox"/>
Минская	real	<input checked="" type="checkbox"/>
Могилевская	real	<input checked="" type="checkbox"/>

Рис. 6.13. Структура записи таблицы ТНаличиеСельхозТехникиВХоз

Таблица - dbo...етениеТехники			Сводка
Имя столбца	Тип данных	Разрешить значения null	
▶ Год	smallint	<input checked="" type="checkbox"/>	
КодСХТехники	smallint	<input checked="" type="checkbox"/>	
Брестская	real	<input checked="" type="checkbox"/>	
Витебская	real	<input checked="" type="checkbox"/>	
Гомельская	real	<input checked="" type="checkbox"/>	
Гродненская	real	<input checked="" type="checkbox"/>	
Минская	real	<input checked="" type="checkbox"/>	
Могилевская	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.14. Структура записи таблицы **ТНаличиеСельхозТехникиВХоз**

Таблица - dbo...никовСХТысЧел			Сводка
Имя столбца	Тип данных	Разрешит...	
▶ Год	int	<input type="checkbox"/>	
Брестская	real	<input checked="" type="checkbox"/>	
Витебская	real	<input checked="" type="checkbox"/>	
Гомельская	real	<input checked="" type="checkbox"/>	
Гродненская	real	<input checked="" type="checkbox"/>	
Минская	real	<input checked="" type="checkbox"/>	
Могилевская	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.15. Структура записи таблицы **ЧислРаботниковСХТысЧел**

Структуры записей таблиц базы данных **СельскоеХозяйство**, содержащих сведения о данных по **растениеводству**, представлены на рисунках 6.16–6.19.

Таблица - dbo.РСХКультуры			Сводка
Имя столбца	Тип данных	Разрешить значения null	
▶ КодСХКультуры	smallint	<input type="checkbox"/>	
НаимСХКультуры	char(25)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.16. Структура записи таблицы **РСХКультуры**

Таблица - dbo.РСХКультуры			Сводка
КодСХКультуры	НаимСХКультуры		
1	Пшеница		
2	Ячень		
3	Рожь		
4	Тритикале		
5	Гречиха		
6	Кукуруза		
7	Овёс		
8	Картофель		
9	Лён-долгунец		
10	Сахарная свекла		
11	Рапс		
12	Капуста		
13	Зерно		
14	Зерновые и зернобобовые		
▶*	NULL	NULL	

Рис. 6.17. Таблица **РСХКультуры** в режиме просмотра

Таблица - dbo...ПлВСХОргТысГа			Сводка
Имя столбца	Тип данных	Разрешить значения null	
Год	int	<input checked="" type="checkbox"/>	
КодКультуры	smallint	<input checked="" type="checkbox"/>	
Брестская	real	<input checked="" type="checkbox"/>	
Витебская	real	<input checked="" type="checkbox"/>	
Гомельская	real	<input checked="" type="checkbox"/>	
Гродненская	real	<input checked="" type="checkbox"/>	
Минская	real	<input checked="" type="checkbox"/>	
Могилёвская	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.18. Структура записи таблицы РПосевнаяПлВСХОргТысГа

Таблица - dbo...ьтурВСХОргЦГа			Сводка
Имя столбца	Тип данных	Разрешит...	
Год	int	<input checked="" type="checkbox"/>	
КодКультуры	smallint	<input checked="" type="checkbox"/>	
Брестская	real	<input checked="" type="checkbox"/>	
Витебская	real	<input checked="" type="checkbox"/>	
Гомельская	real	<input checked="" type="checkbox"/>	
Гродненская	real	<input checked="" type="checkbox"/>	
Минская	real	<input checked="" type="checkbox"/>	
Могилёвская	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.19. Структура записи таблицы РУрожайКультурВСХОргЦГа

Структуры записей таблиц базы данных **СельскоеХозяйство**, содержащих сведения о данных по производству и потреблению продуктов питания представлены на рисунках 6.20–6.22.

Таблица - dbo...ПитанияВГодКг			Сводка
Имя столбца	Тип данных	Разрешит...	
Год	int	<input type="checkbox"/>	
МясоИМясопродукты	real	<input checked="" type="checkbox"/>	
МолокоИМолокопрод...	real	<input checked="" type="checkbox"/>	
ХлебныеПродукты	real	<input checked="" type="checkbox"/>	
Картофель	real	<input checked="" type="checkbox"/>	
Овощи	real	<input checked="" type="checkbox"/>	
ПлодыИЯгоды	real	<input checked="" type="checkbox"/>	
ЯйцаШтук	real	<input checked="" type="checkbox"/>	
Сахар	real	<input checked="" type="checkbox"/>	
МаслоРастительное	real	<input checked="" type="checkbox"/>	
Рыба	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.20. Структура записи таблицы ПотреблПродПитанияВГодКг

Таблица - db...аДушуНаселКг		Таблица - dbo...ПитанияВГодКг	
Имя столбца	Тип данных	Разрешит...	
▶ Год	int	<input checked="" type="checkbox"/>	
Зерно	real	<input checked="" type="checkbox"/>	
Картофель	real	<input checked="" type="checkbox"/>	
Овощи	real	<input checked="" type="checkbox"/>	
СахарнаяСвекла	real	<input checked="" type="checkbox"/>	
Мясо	real	<input checked="" type="checkbox"/>	
Молоко	real	<input checked="" type="checkbox"/>	
ЯйцаШтук	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.21. Структура записи таблицы **ПроизПродНаДушуНаселКг**

Таблица - db...итНормаВГодКг		Сводка	
Имя столбца	Тип данных	Разрешить значения null	
▶ МясоИМясопродукты	real	<input checked="" type="checkbox"/>	
МолокоИМолокопрод...	real	<input checked="" type="checkbox"/>	
ХлебныеПродукты	real	<input checked="" type="checkbox"/>	
Картофель	real	<input checked="" type="checkbox"/>	
Овощи	real	<input checked="" type="checkbox"/>	
ПлодыИЯгоды	real	<input checked="" type="checkbox"/>	
ЯйцаШтук	real	<input checked="" type="checkbox"/>	
Сахар	real	<input checked="" type="checkbox"/>	
МаслоРастительное	real	<input checked="" type="checkbox"/>	
Рыба	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.22. Структура записи таблицы **ПотреблПродПитНормаВГодКг**

Структуры записей таблиц базы данных **СельскоеХозяйство**, содержащих сведения о данных по животноводству, представлены на рисунках 6.23–6.27.

Таблица - dbo...аВСХОргТысГол			
Имя столбца	Тип данных	Разрешить значения null	
▶ Год	int	<input checked="" type="checkbox"/>	
КодВидаСХЖивотн	smallint	<input checked="" type="checkbox"/>	
Брестская	real	<input checked="" type="checkbox"/>	
Витебская	real	<input checked="" type="checkbox"/>	
Гомельская	real	<input checked="" type="checkbox"/>	
Гродненская	real	<input checked="" type="checkbox"/>	
Минская	real	<input checked="" type="checkbox"/>	
Могилевская	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.23. Структура записи таблицы **ЖПоголовьеСкотаВСХОргТысГол**

Таблица - db...ой1КорСХОргКг			
Имя столбца	Тип данных	Разрешить значения null	
Год	int	<input checked="" type="checkbox"/>	
Брестская	real	<input checked="" type="checkbox"/>	
Витебская	real	<input checked="" type="checkbox"/>	
Гомельская	real	<input checked="" type="checkbox"/>	
Гродненская	real	<input checked="" type="checkbox"/>	
Минская	real	<input checked="" type="checkbox"/>	
Могилёвская	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.24. Структура записи таблицы ЖУдой1КорСХОргКг

Таблица - db...тПривесСвинГр			
Имя столбца	Тип данных	Разрешить значения null	
Год	int	<input checked="" type="checkbox"/>	
Брестская	real	<input checked="" type="checkbox"/>	
Витебская	real	<input checked="" type="checkbox"/>	
Гомельская	real	<input checked="" type="checkbox"/>	
Гродненская	real	<input checked="" type="checkbox"/>	
Минская	real	<input checked="" type="checkbox"/>	
Могилёвская	real	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.25. Структура записи таблицы ЖСреднеСутПривесСвинГр

Таблица - db...дыСХЖивотных			
Имя столбца	Тип данных	Разрешит...	
КодВидаСХЖивотных	smallint	<input type="checkbox"/>	
НаименВидаСХЖивотных	char(25)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Рис. 6.26. Структура записи таблицы ЖВидыСХЖивотных

Таблица - db...дыСХЖивотных		
	КодВидаСХЖи...	НаименВидаСХЖивотных
▶	1	Коровы
	2	Свиньи
	3	Птица
	4	Лошади
	5	Овцы
	6	МММ
*	МММ	МММ

Рис. 6.27. Таблица ЖВидыСХЖивотных

Среда SQL Server

Чтобы войти в среду SQL Server необходимо выполнить следующие действия:

1. Выбрать команды **Пуск – Программы – SQL Server – Среда SQL Server Management Studio Express**.

2. После настройки среды нажать вкладку **Соединить**. Откроются окна системы **SQL Server Management Studio Express** (рис. 6.28).

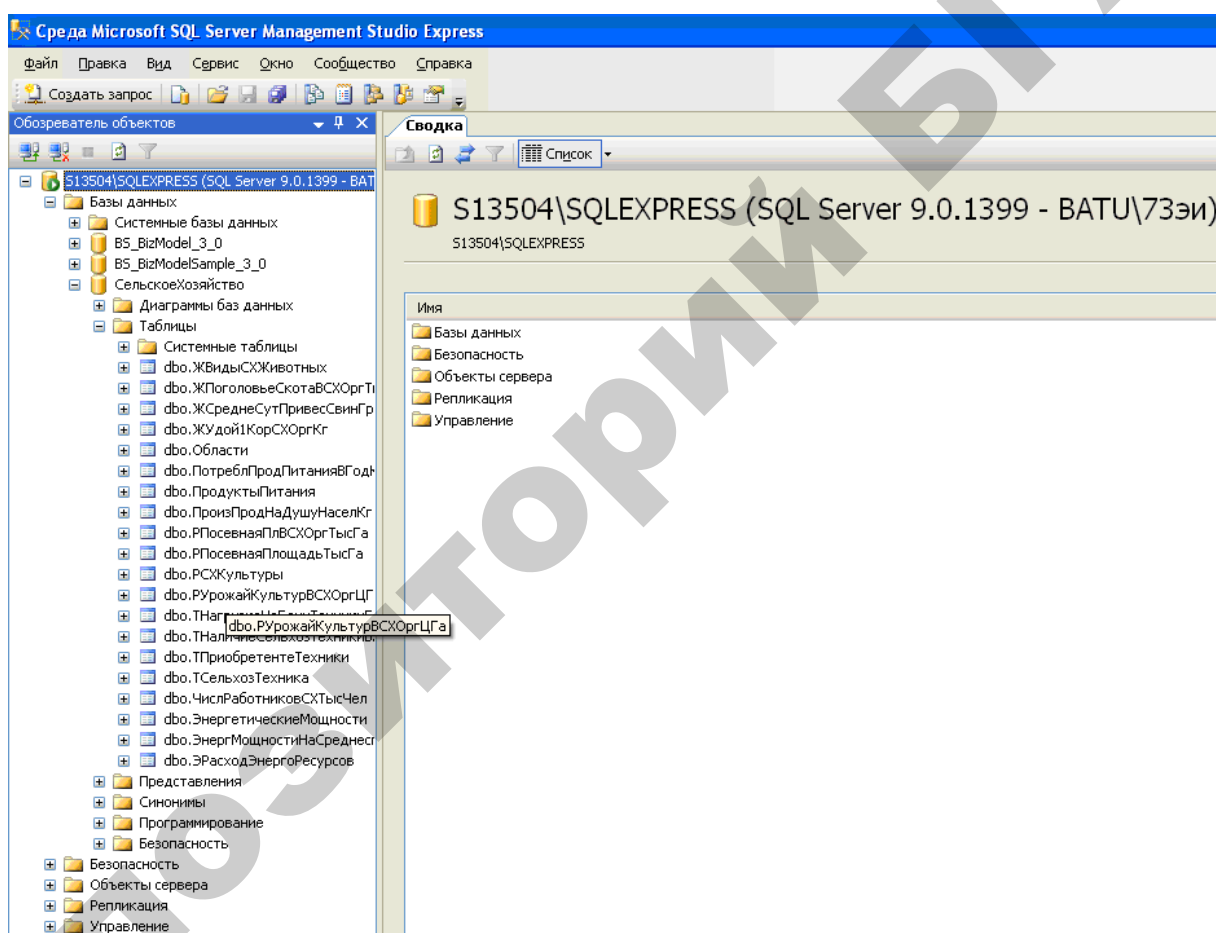




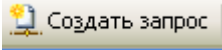
Рис. 6.28. Интерфейс среды SQL Server Management Studio Express

3. В левой части экрана возникает окно **Обозреватель объектов**, в котором можно открыть список объектов системы. Для просмотра списка баз данных открыть папку **Базы данных**. В открывшейся папке содержатся системные базы данных (в соответствующей папке) и пользовательские базы данных. Для создания объекта следует щелкнуть правой клавишей мыши по папке объектов (например **Базы данных**) и выбрать пункт **Создать**.

Для создания запроса к базе данных следует нажать кнопку  на панели инструментов при выделенном имени обрабатываемой базы данных в списке баз данных в **Обозревателе объектов**. В открывшемся окне SQLQuery.sql ввести текст запроса на языке SQL или создать запрос с помощью конструктора. Для этого щелкнуть правой клавишей мыши по свободному полю окна SQLQuery.sql и в открывшемся списке выбрать  .


Задание 1. *Вывести наличие сельскохозяйственной техники всех видов по всем областям за последний имеющийся в справочнике год отчетности.*

Выполняемые действия

1. Выделить имя базы данных **СельскоеХозяйство** в списке баз данных в Обозревателе объектов.
2. Нажать кнопку  на панели инструментов.
3. Ввести текст запроса на языке SQL.

```
Select * from TНаличиеСельхозТехникиВХоз, ТСельхозТехника  
Where Год=2008 And
```

```
TНаличиеСельхозТехникиВХоз.КодСХТехники=ТСельхозТехника.КодСХТехники
```

4. Выполнить запрос, нажав на кнопку .
5. В своей личной папке создать папку “Запросы SQL”. Сохранить созданный запрос в папке “Запросы SQL” с помощью команд

Файл – Сохранить Query1.Sql как – [Выбрать папку] – Сохранить.

Задание 2. *Вывести среднесписочную численность работников сельского хозяйства Витебской и Гомельской областей в те годы, когда в Витебской области она была больше чем в Гомельской.*

Выполняемые действия

Пользуясь указаниями **Задания1**, создать, выполнить и сохранить запрос, введя следующую инструкцию SQL.

```
Select Год, Витебская, Гомельская From ЧислРаботниковСХТысЧел  
Where Витебская>Гомельская
```

Задание 3. Вывести информацию о приобретении в Витебской области тех видов техники, которых приобреталось более 500 штук по областям за последние три года.

Текст запроса на языке SQL

```
Select * from TПриобретениеТехники, ТСельхозТехника
Where Год>=2006 And Год<=2008 And
Брестская+Витебская+Гомельская+Гродненская+Минская+Могилевская >
1550 And TПриобретениеТехники.КодСХТехники=ТСельхозТехника.КодСХТехники
```

Задание 4. Вывести информацию о расходе энергетических ресурсов в те годы, когда энергетические мощности составляли менее 20 000 тыс. л. С. Рассчитать общий объем приобретенного топлива по формуле:

**ОбъемПриобрТоплива = ПриобретеноАвтобензинаТысТонн+
ПриобретеноДизТопливаТысТонн.**

Текст запроса на языке SQL

```
Select * ,
ОбъемПриобрТоплива=ПриобретеноАвтобензинаТысТонн+
ПриобретеноДизТопливаТысТонн
from ЭРасходЭнергоРесурсов
Where ЭнергМощностиТысЛС < 20000
```

Задание 5. Вывести информацию о посевной площади картофеля во всех сельскохозяйственных организациях по районам в 2004–2008 годах в порядке убывания года отчетности.

Текст запроса на языке SQL

```
Select * from RПосевнаяПлВСХОргТысГа, РСХКультуры
Where RПосевнаяПлВСХОргТысГа.Год>=2004 and
RПосевнаяПлВСХОргТысГа.Год<=2008
And RПосевнаяПлВСХОргТысГа.КодКультуры=8
And
RПосевнаяПлВСХОргТысГа.КодКультуры=РСХКультуры.КодСХКультуры
Order by RПосевнаяПлВСХОргТысГа.Год desc
```

Задание 6. Создать запрос на вывод информации о посевных площадях зерновых и зернобобовых культур по Минской и Брестской областям Беларуси а так же об урожайности указанных культур по годам с расчетом валового сбора зерновых и зернобобовых культур в тоннах по формуле:

$$\text{ВаловойСбор} = \text{ПлощадьПосев} * \text{Урожайность} * 1000 / 10.$$

Текст запроса на языке SQL

```
Select РПосевнаяПлВСХОргТысГа.Год, МинскаяОблПосевнаяПлощадь=
РПосевнаяПлВСХОргТысГа.Минская,
МинскаяОблУрожайность=РУрожайКультурВСХОргЦГа.Минская,
МинскаяОблВалСборТонн=РПосевнаяПлВСХОргТысГа.Минская*
РУрожайКультурВСХОргЦГа.Минская*1000/10,
БрестскаяОблПосевнаяПлощадь= РПосевнаяПлВСХОргТысГа.Брестская,
БрестскаяОблУрожайность=РУрожайКультурВСХОргЦГа.Брестская,
БрестскаяОблВалСборТонн=РПосевнаяПлВСХОргТысГа.Брестская*
РУрожайКультурВСХОргЦГа.Брестская*1000/10
From РПосевнаяПлВСХОргТысГа, РУрожайКультурВСХОргЦГа
Where РПосевнаяПлВСХОргТысГа.Год=РУрожайКультурВСХОргЦГа.Год
And
РПосевнаяПлВСХОргТысГа.КодКультуры=РУрожайКультурВСХОргЦГа.КодКультуры
```

Результирующая таблица примет вид, подобный рисунку 6.29.

Год	МинскаяОблПос...	МинскаяОблПло...	МинскаяОблВалСбо...	БрестскаяОблПосев...	БрестскаяОблУрож...	БрестскаяОблВалСборТонн
1 1990	545,3	27,3	1488669	374,7	29,6	1109112
2 1995	539,6	22,2	1197912	372,9	23,7	883773
3 2000	636,8	18,9	1265242	399,9	21,6	731184
4 2004	615,7	28,2	1736274	377,6	31,1	1174336
5 2005	563,9	28,2	1590199	365,8	27,9	992681,9
6 2006	604,5	23,6	1428620	375,5	22,2	833610,1
7 2007	612,5	28,2	1727250	380,1	26,6	1011066

Рис. 6.29. Результирующая таблица запроса задания 6

Задание 7. Вывести информацию о производстве и потреблении мяса на душу населения в РБ с расчетом индекса потребления мяса на душу населения по формуле: **Индекс потребления = Потреблено / Произведено.**

Так же рассчитать уровень потребления от медицинской нормы по формуле:

Уровень потребления от мед. нормы = Потреблено / Рекомен. Норма.

Рекомендуется создать запрос с помощью конструктора запросов.

Выполняемые действия

1. Выделить имя базы данных **СельскоеХозяйство** в списке баз данных в Обозревателе объектов.

2. Нажать кнопку **Создать запрос**, перевести курсор в область текста запроса, нажать правую кнопку мыши, выбрать пункт **СОЗДАТЬ ЗАПРОС В РЕДАКТОРЕ ...**

Откроется окно конструктора запросов и в нем окно для добавления таблиц.

3. Добавить таблицы **ПроизПродНаДушуНаселКг**, **ПотреблПродПитанияВГодКг** и **ПотреблПродПитНормаВГодКг**, щелкнув клавишей мыши по их именам. Структуры таблиц появятся в верхней части окна конструктора запросов.

4. Переместить поля **Год**, **Мясо** из таблицы **ПроизПродНаДушуНаселКг** и поле **МясоИМясопродукты** из таблицы **ПотреблПродПитанияВГодКг** в макет запроса, включив элемент управления **Флажок** рядом с именем каждого перемещаемого поля. Имена полей появятся в макете запроса в графе **Столбец**.

5. Вслед за именами полей в графу **Столбец** ввести формулы для вычисления значений расчетных столбцов результирующей таблицы **МясоИМясопродукты / Мясо** и **ПотреблПродПитанияВГодКг.МясоИМясопродукты / ПотреблПродПитНормаВГодКг.МясоИМясопродукты**.

6. В столбец **Псевдоним** макета запроса ввести имена полей результирующей таблицы запроса. Для поля **Год** псевдоним не указывать, для остальных полей ввести:

Произведено на душу населения мяса,

Потребление мяса на душу населения,

Индекс потребления (вместо стандартного имени Expr1),

Уровень потребления к медиц. норме (вместо стандартного имени Expr2).

Замечание. При отсутствии псевдонима в качестве имени поля результирующей таблицы принимается имя поля исходной таблицы.

Окно **КОНСТРУКТОРА ЗАПРОСОВ** будет иметь вид, подобный приведенному на рисунке 6.30.

Столбец	Псевдоним	Таблица	Вывод
Год		ПроизПродНаДушуН...	<input checked="" type="checkbox"/>
Мясо	[Произведено на душу населения мяса]	ПроизПродНаДушуН...	<input checked="" type="checkbox"/>
МясойМясопродукты	[Потребление мяса на душу населения]	ПотреблПродПитани...	<input checked="" type="checkbox"/>
ПотреблПродПитанияВГодКг.МясойМясопродукты / ПроизПродНаДушуНаселКг.Мясо	[Индекс потребления]		<input checked="" type="checkbox"/>
▶ ПотреблПродПитанияВГодКг.МясойМясопродукты / ПотреблПродПитНормаВГодКг.МясойМясопродукты	Уровень потребления к медиц. норме		<input checked="" type="checkbox"/>

```

SELECT  ПроизПродНаДушуНаселКг.Год, ПроизПродНаДушуНаселКг.Мясо AS [Произведено на душу населения мяса],
        ПотреблПродПитанияВГодКг.МясойМясопродукты AS [Потребление мяса на душу населения],
        ПотреблПродПитанияВГодКг.МясойМясопродукты / ПроизПродНаДушуНаселКг.Мясо AS [Индекс потребления],
        ПотреблПродПитанияВГодКг.МясойМясопродукты / ПотреблПродПитНормаВГодКг.МясойМясопродукты AS [Уровень потребления к медиц. норме]
FROM    ПроизПродНаДушуНаселКг INNER JOIN
        ПотреблПродПитанияВГодКг ON ПроизПродНаДушуНаселКг.Год = ПотреблПродПитанияВГодКг.Год CROSS JOIN
        ПотреблПродПитНормаВГодКг
    
```

Рис. 6.30. Результирующая таблица запроса задания 7

Задание 8. Создать запрос на выбор информации о поголовье коров в сельскохозяйственных организациях в Брестской и Минской областях, а также об удоях молока с одной коровы с расчетом производства молока в сельскохозяйственных организациях в Брестской и Минской областях:

ПроизводствоМолока =Надой с одной коровы*Поголовье/1000.

Замечание. Воспользоваться методическими рекомендациями задания 7.

Окно Конструктора запросов приведено на рисунке 6.31.

Столбец	Псевдоним	Таблица	Вывод	Тип сортиро...	Порядок сорт...	Фильтр
Год		ЖПоголов...	<input checked="" type="checkbox"/>			
НаименВидаСХЖивотных		ЖВидыСХ...	<input checked="" type="checkbox"/>			
Брестская	[Удой с одной коровы в Брестской обл.]	ЖУдой1К...	<input checked="" type="checkbox"/>			
Брестская	[Поголовье коров в Брестской области]	ЖПоголов...	<input checked="" type="checkbox"/>			
ЖУдой1КорСХОргКг.Брестская * ЖПоголовьеСкотаВСХОргТысГол.Брестская / 100	[Объем производства молока в брестской области]		<input checked="" type="checkbox"/>			
Минская	[Удой с одной коровы в Минской области обл.]]]	ЖПоголов...	<input checked="" type="checkbox"/>			
Минская	[Поголовье коров в Минской области]	ЖПоголов...	<input checked="" type="checkbox"/>			
ЖУдой1КорСХОргКг.Минская * ЖПоголовьеСкотаВСХОргТысГол.Минская / 100	[Объем производства молока в Минской области]		<input checked="" type="checkbox"/>			
КодВидаСХЖивотн		ЖПоголов...	<input checked="" type="checkbox"/>			= 1

```

SELECT ЖПоголовьеСкотаВСХОргТысГол.Год, ЖВидыСХЖивотных.НаименВидаСХЖивотных,
ЖУдой1КорСХОргКг.Брестская AS [Удой с одной коровы в Брестской обл.],
ЖПоголовьеСкотаВСХОргТысГол.Брестская AS [Поголовье коров в Брестской области],
ЖУдой1КорСХОргКг.Брестская * ЖПоголовьеСкотаВСХОргТысГол.Брестская / 100 AS [Объем производства молока в брестской области],
ЖПоголовьеСкотаВСХОргТысГол.Минская AS [Удой с одной коровы в Минской области обл.]],
ЖПоголовьеСкотаВСХОргТысГол.Минская AS [Поголовье коров в Минской области],
ЖУдой1КорСХОргКг.Минская * ЖПоголовьеСкотаВСХОргТысГол.Минская / 100 AS [Объем производства молока в Минской области],
ЖПоголовьеСкотаВСХОргТысГол.КодВидаСХЖивотн
FROM ЖПоголовьеСкотаВСХОргТысГол INNER JOIN
ЖВидыСХЖивотных ON ЖПоголовьеСкотаВСХОргТысГол.КодВидаСХЖивотн = ЖВидыСХЖивотных.КодВидаСХЖивотных INNER JOIN
ЖУдой1КорСХОргКг ON ЖПоголовьеСкотаВСХОргТысГол.Год = ЖУдой1КорСХОргКг.Год
WHERE (ЖПоголовьеСкотаВСХОргТысГол.КодВидаСХЖивотн = 1)
    
```

Рис. 6.31. Конструктор запроса задания 8

Задание 9. Создать запрос на вывод информации о производстве свинины в сельскохозяйственных организациях Гродненской области. Вывести поголовье свиней, среднесуточный привес свиней, рассчитать Объем производства свинины по формуле:

ПроизводствоСвинины = Поголовье * Среднесуточный привес * 365/1000.

Окно конструктора запросов приведено на рисунке 6.32.

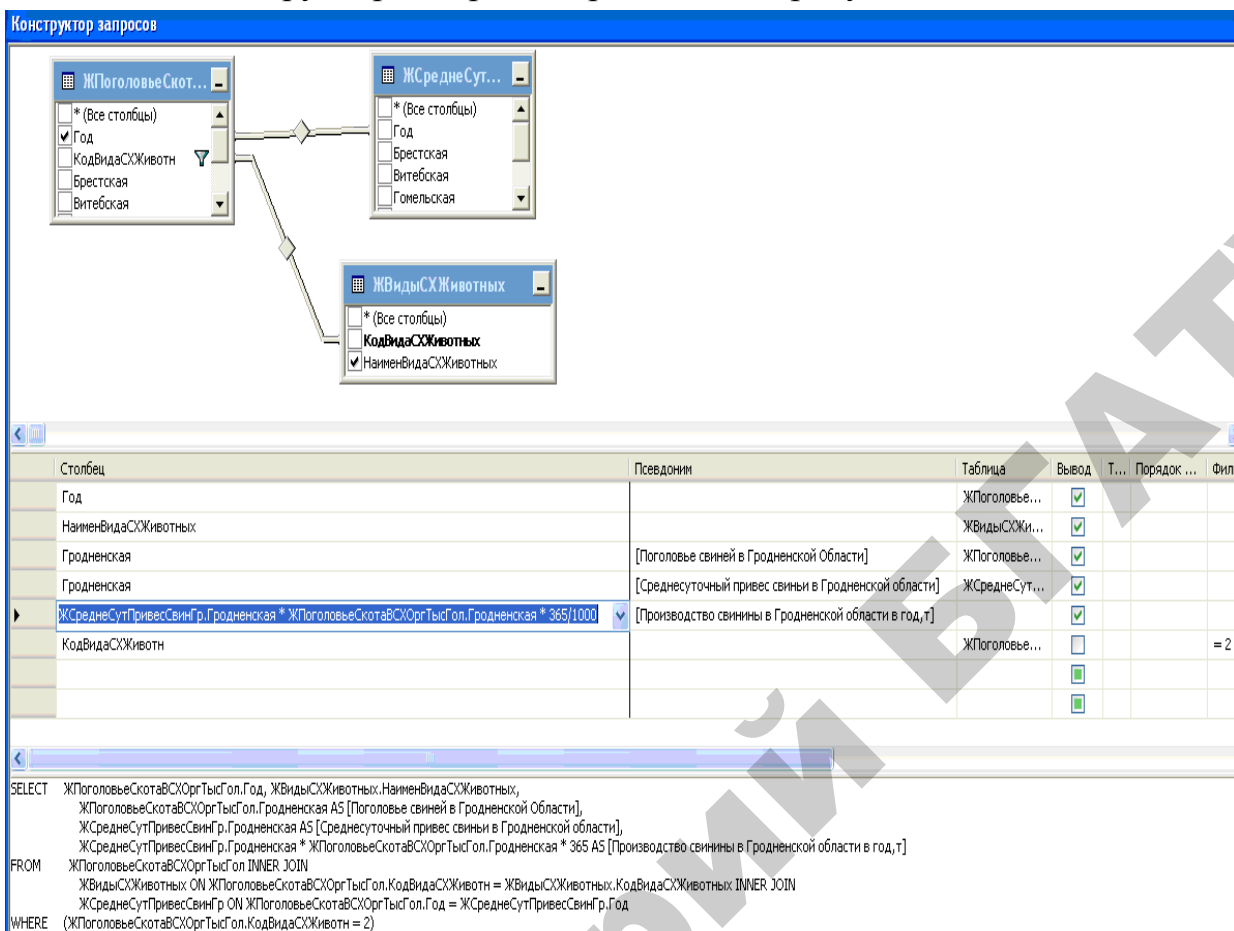


Рис. 6.32. Конструктор запроса задания 9

Задания для самостоятельного выполнения

1. Вывести наличие сельскохозяйственной техники всех видов по всем областям за последние три года отчетности, содержащейся в справочнике.

2. Вывести среднесписочную численность работников сельского хозяйства Минской, Витебской и Гродненской областей.

3. Вывести информацию о посевных площадях рапса и льна во всех сельскохозяйственных организациях по районам в 2004–2008 годах в порядке убывания года отчетности.

4. Создать запрос на вывод информации о посевных площадях зерновых и зернобобовых культур по Гродненской области, а также об урожайности указанных культур по годам с расчетом валового сбора зерновых и зернобобовых культур в тоннах по формуле:

$$\text{ВаловойСбор} = \text{ПлощадьПосев} * \text{Урожайность} * 1000 / 10.$$

5. Создать запрос на выбор информации о поголовье коров во всех категориях организаций в Могилевской и Гомельской областях, а также об удоях молока с одной коровы с расчетом производства молока в сельскохозяйственных организациях в Могилевской и Гомельской областях:

Производство Молока =Надой с одной коровы*Поголовье/1000.

Вопросы для самоконтроля

1. Каким образом можно получить доступ к данным набора записей при программировании на языке SQL?
2. Какие методы используются при перемещении в наборе записей при программировании на языке SQL?
3. Как определить, что набор записей пуст?
4. Как определить, что указатель на текущую запись вышел за пределы набора записей?
5. Чем многопользовательские СУБД отличаются от настольных СУБД?
6. Охарактеризуйте основные принципы функционирования архитектуры файл/сервер.
7. Охарактеризуйте основные принципы функционирования архитектуры клиент/сервер.
8. Приведите примеры серверных СУБД.

ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ ПО МОДУЛЮ 6

1. Классом объектов называется:
 - группа аналогичных объектов, имеющих подобные свойства, методы и события;
 - группа аналогичных объектов, имеющих подобные названия, описания и действия;
 - группа различных объектов, имеющих различные свойства, методы и события.
2. Класс форм БД включает в себя:
 - совокупность всех открытых форм БД;
 - главную форму БД;

- набор форм, выбранный пользователем.
3. Экземпляр класса – это:
- один представитель класса;
 - все представители класса;
 - нет такого понятия.
4. Класс форм БД именуется:
- Forms
 - Form
 - Form's
5. Объектами доступа к данным Jet являются:
- описание таблицы (TableDefs);
 - описание запроса (QueryDefs);
 - набор записей (RecordSet);
 - формы (Forms);
 - отчеты (Reports).
6. Для ссылки на форму с именем **Spisok** используется следующая конструкция языка VBA:
- Forms!Spisok
 - Forms.Spisok
 - Forms:Spisok
 - Forms@Spisok
7. Для ссылки на свойство **Caption** формы **Spisok** используется следующая конструкция языка VBA:
- Form!Spisok.Caption
 - Form.Spisok.Caption
 - Form!Spiso!Caption
8. Объекты **TableDef** и **QueryDef** могут представлять данные, хранимые в таблицах БД. НЕВЕРНО.
9. Физическое перемещение позволяет переходить от одной записи к другой:
- согласно их положению в наборе;
 - согласно значению активного индекса набора.
10. Логическое перемещение позволяет переходить от одной записи к другой:
- согласно их положению в наборе;
 - согласно значению активного индекса набора.

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Выполнить индивидуальные задания в соответствии с вариантом.

Создать печатный отчет о выполнении задания по следующей структуре:

1. титульный лист с указанием фамилии, группы исполнителя, темы задания;
2. постановка задачи;
3. описание выполнения задания по каждой лабораторной работе с приведением созданных объектов в режиме конструктора и в режиме выполнения;
4. анализ результатов;
5. заключение.

ВАРИАНТ 1

Расчет заработной платы сотрудникам предприятия

Создать базу данных для расчета заработной платы сотрудникам предприятия. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

Табель

<i>Год</i>	<i>Месяц</i>	<i>ТабНомСотрудника</i>	<i>КодПодразделения</i>	<i>КоличествоОтработДней</i>	<i>Дневной тариф</i>
------------	--------------	-------------------------	-------------------------	------------------------------	----------------------

Сотрудники

<i>ТабНомСотрудника</i>	<i>ФамилияСотрудника</i>	<i>КодДолжности</i>	<i>Разряд</i>
-------------------------	--------------------------	---------------------	---------------

Должности

<i>КодДолжности</i>	<i>НаименованиеДолжности</i>
---------------------	------------------------------

Подразделения

<i>КодПодразделения</i>	<i>НаименованиеПодразделения</i>
-------------------------	----------------------------------

Создание таблиц

1. Создать справочники: **Должности**, **Подразделения**.
2. Описать структуры таблиц **Сотрудники**, **Табель**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Сотрудники**, организовав поле **КодПрофессии** в виде поля со списком.
5. Заполнить таблицу **Сотрудники** с помощью созданной формы, выбирая значения поля **КодДолжности** из списка.
6. Создать форму для заполнения таблицы **Табель**, организовав поля **ТабНомСотрудника** и **КодПодразделения** в виде полей со списком.
7. Заполнить таблицу **Табель** с помощью созданной формы, выбирая поля **ТабНомСотрудника** и **КодПодразделения** из списков.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список сотрудников, разряд которых не менее 3. Выполнить сортировку по полю **Разряд**.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **Табель**, но объекты бы представлялись своими наименованиями. Выполнить сортировку по полю **ФамилияСотрудника**.
3. Составить запрос **Запрос3**, результирующая таблица которого должна содержать все поля **Запроса2**, кроме того добавить графу **НаименДолжности** (взятую из таблицы **Должности**) и графу **СуммаЗаработано** с расчетом по формуле:
$$\text{СуммаЗаработано} = \text{Дневной тариф} * \text{КоличествоОтработДней}.$$
Выполнить сортировку по полям **Год**, **Месяц**, **НаименованиеПодразделения**, **ФамилияСотрудника**.
4. Составить параметрический запрос, отображающий табель по заданному сотруднику в форме **Запрос3**
5. Составить запрос с группировкой, отображающий информацию об общей заработной сумме по каждому подразделению.
6. Создать перекрестный запрос, отображающий заработанные суммы по каждой профессии в каждом подразделении.

<i>Должность</i>	<i>Сумма заработано</i>	<i>Подразделение 1</i>	<i>Подразделение 2</i>
------------------	-------------------------	------------------------	------------------------

Запросы-действия

7. Создать таблицу **ТабельПодробно**, подобную результирующей таблице **Запрос3** с помощью запроса на создание таблицы.

8. Добавить в таблицу **ТабельПодробно** столбец **Организация**, заполнить это поле значением БГАТУ с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **ТабельПодробно** строк табеля текущего месяца со значением поля **Месяц** на единицу больше.

10. Составить запрос с параметром на удаление записей о заработке заданного сотрудника из таблицы **ТабельПодробно**.

SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **Табель1**, подобной таблице **Табель**, но добавить поля **СуммаЗаработано** и **Надбавка**.

12. С помощью инструкции SQL описать структуру таблицы **ПодразделенияЗаработано**, содержащей следующие поля: **Год**, **Месяц**, **КодПодразделения**, **СуммаЗаработаноПод**.

13. Заполнить таблицу **Табель1** тремя записями с помощью инструкций SQL (поле **СуммаЗаработано** оставить пустым).

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **Табель1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **Табель1**.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Сотрудники1**: **ФамилияСотрудника**, **Разряд**.

17. С помощью инструкции SQL вывести информацию таблицы **Сотрудники1**, добавив поле **НаименованиеДолжности** из связанной таблицы **Должности**.

18. С помощью инструкции SQL вывести информацию таблицы **Табель**, представив сотрудников фамилиями, взятыми из таблицы **Сотрудники**, а подразделения наименованиями, взятыми из таблицы **Подразделения**.

Создание форм

1. Создать форму, позволяющую вносить сведения о новой должности в справочник. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую отображать и вводить новые сведения в таблицу, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются подразделения, а в подчиненной – табель по текущему подразделению (по запросу “Запрос3”), с подсчетом итогового количества отработанных дней по подразделению.

4. Создать форму в виде сводной таблицы, содержащей информацию о расчете заработной платы сотрудникам всех подразделений с расчетом общей суммы, заработанной каждым подразделением, общей суммы, заработанной сотрудниками каждой должности и общей заработной платы всеми сотрудниками.

Создание отчетов

1. Создать отчет об имеющихся в справочнике сотрудниках, включив поля **ФамилияСотрудника**, **Разряд**, **ПовышенныйРазряд** с расчетом поля **ПовышенныйРазряд** по формуле: $ПовышенныйРазряд = Разряд + 1$ и количества сотрудников в справочнике.

2. Создать отчет с группировкой по подразделениям, отображающий информацию о расчете заработной платы сотрудникам предприятия (по **Запрос3**) с получением промежуточного итога по каждому подразделению в столбце **СуммаЗаработано**.

3. С помощью *Мастера отчетов* создать отчет с группировкой по должностям, отображающий информацию о расчете заработной платы сотрудникам предприятия с расчетом суммы, заработанной каждым сотрудником, с получением промежуточного итога: общей заработной платы по каждой должности, общей суммы, заработанной всеми сотрудниками.

Создание модулей VBA

1. С помощью программного модуля VBA рассчитать значение поля **СуммаЗаработано** таблицы **Табель1** по формуле:

СуммаЗаработано= **Дневной тариф** * **КоличествоОтработДней**.

2. Создать программный модуль VBA, рассчитывающий в таблице **Табель1** поле **Надбавка** сотрудникам, имеющим разряд не менее 4 в размере 10 процентов от заработной суммы. Для остальных сотрудников поле **Надбавка** оставить равным нулю.

3. С помощью программного модуля VBA заполнить таблицу **ПодразделенияЗаработано** данными, рассчитав общую заработанную сумму каждым подразделением и приняв ее в качестве значения поля **СуммаЗаработаноПод** в таблице **ПодразделенияЗаработано**.

ВАРИАНТ 2

Поступление материалов на предприятие

Создать базу данных для учета поступления материалов на предприятие. Количество записей в справочниках должно быть не менее 3–4, соответствующей агрегированной сущности, – не менее 10.

Замечание. **МатОтЛица**, **МОЛ** – материально-ответственные лица.

Исходные таблицы: сводная таблица – **Накладные** – отражает поступление материалов по накладным к материально-ответственным лицам.

Накладные

<i>НомДок</i>	<i>ДатаПост</i>	<i>Поставщик</i>	<i>НоменкНомМатериала</i>	<i>ТабНомМОЛ</i>	<i>Колич</i>
---------------	-----------------	------------------	---------------------------	------------------	--------------

Материалы

<i>НоменклНомМатериала</i>	<i>НаименованиеМатериала</i>	<i>ЕдиницаИзмерения</i>	<i>Цена</i>
----------------------------	------------------------------	-------------------------	-------------

МатОтЛица

<i>ТабНомМОЛ</i>	<i>ФамилияМОЛ</i>
------------------	-------------------

Поставщики

<i>НомПоставщмк</i>	<i>НаименПоставщика</i>
---------------------	-------------------------

Создание таблиц

1. Создать справочники: **Материалы**, **МатОтЛица**, **Поставщики**.
2. Описать структуру таблицы **Накладные**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Накладные**, организовав поля **Поставщик**, **НоменкНомМатериала**, **ТабНомМОЛ** в виде полей со списком.
5. Заполнить таблицу **Накладные** с помощью созданной формы, выбирая значения полей **Поставщик**, **НоменкНомМатериала**, **ТабНомМОЛ** из списков.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список материалов, цена которых более 500 000 рублей.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **Накладные**, но объекты бы представлялись своими наименованиями, выбранными из соответствующих справочников. Выполнить сортировку по полю **ДатаПост**.
3. Составить запрос **Запрос3**, результирующая таблица которого должна содержать все поля **Запроса2**, кроме того добавить поле **Стоимость** с расчетом по формуле: **Стоимость = Цена * Колич**.
Выполнить сортировку по полям **Поставщик**, **ДатаПост**.
4. Составить параметрический запрос, отображающий накладные по заданно-му материалу в форме **Запрос3**.
5. Составить запрос с группировкой, отображающий стоимость поступлений от каждого поставщика.
6. Создать перекрестный запрос, отображающий стоимость поступлений каждого материала от каждого поставщика.

Материал	Стоимость поступлений	Поставщик1	Поставщик2
----------	-----------------------	------------	------------

Запросы-действия

7. Создать таблицу **НакладныеПодробно**, подобную результирующей таблице **Запрос3** с помощью запроса на создание таблицы.

8. Добавить в таблицу **НакладныеПодробно** столбец **Организация**, заполнить это поле значением БГАТУ с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **НакладныеПодробно** накладной, идентичной имеющейся в таблице накладной с последним номером, придать новой накладной номер на единицу больший номера исходной накладной.

10. Составить запрос с параметром на удаление накладных, о поставках материалов от заданного поставщика из таблицы **НакладныеПодробно**.

SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **Накладные1**, подобной таблице **Накладные**, но добавить поле **Стоимость**.

12. С помощью инструкции SQL описать структуру таблицы **Материалы1**, подобной таблице **Материалы**, но добавить числовые поля **Наценка** и **ПоступленияКоличество**.

13. Заполнить таблицу **Накладные1** тремя записями с помощью инструкций SQL (поле **Стоимость** оставить пустым).

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **Накладные1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **Накладные1**. Добавить в таблицу **Накладные1** три строки по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Материалы**: **НаименованиеМатериала**, **Цена**.

17. С помощью инструкции SQL вывести информацию таблицы **Накладные**: **НомДок**, **ДатаПост**, **НоменкНомМатериала**, **ФамилияМОЛ** (из таблицы **МатОтЛица**), **Колич**.

18. С помощью инструкции SQL вывести информацию таблицы **Накладные₂**, представив объекты своими наименования, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новом материале в справочник. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую отображать и вводить новые сведения в таблицу **Накладные**, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются Материально-ответственные лица, а в подчиненной – накладные по материалам, поступивших на соответствующего материально-ответственного работника (по запросу **Запрос3**) с подсчетом итоговой стоимости указанных поступивших материалов.

4. Создать форму в виде сводной таблицы, содержащей информацию о поступлении материалов от всех поставщиков с расчетом общей стоимости материалов, поступивших от каждого поставщика, общей стоимости материалов, поступивших на каждого материально-ответственного работника и общей стоимости всех поступивших материалов.

4 Создание отчетов

1. Создать отчет об имеющихся в справочнике материалах, по структуре таблицы **Материалы**, добавить поле **ЦенаВТыс**, рассчитанное по формуле:

$$\text{ЦенаВТыс} = \text{Цена}/1000.$$

Рассчитать также количество материалов в справочнике.

2. Создать отчет с группировкой по поставщикам, отображающий информацию о поступлении материалов от поставщиков (по **Запрос3**) с получением промежуточного итога по каждому поставщику и общего итога по всей ведомости в столбце **Стоимость**.

3. С помощью *Мастера отчетов* создать отчет с группировкой по материально-ответственным лицам, отображающий информацию о поступлении материалов с расчетом стоимости материалов, поступивших на каждого материально-ответственного работника, и стоимости всех поступивших материалов.

Создание модулей VBA

1. Заполнить таблицу **Материалы1** пятью строками по своему усмотрению, оставив поля **Наценка** и **ПоступленияКоличество** пустыми. Создать программный модуль VBA, рассчитывающий в таблице **Материалы1** поле **Наценка** размером 2 % от цены.

2. С помощью программного модуля VBA рассчитать значение поля **Стоимость** таблицы **Накладные1** по формуле: **Стоимость**= **Цена*** **Колич.**

3. С помощью программного модуля VBA заполнить поле **Поступления-Количество** таблицы **Материалы1**, рассчитав общее количество поступлений каждого материала, исходя из данных таблицы **Накладные**.

ВАРИАНТ 3

Учет основных средств на предприятии

Создать базу данных для учета основных средств на предприятии. Количество записей в справочниках должно быть не менее 3–4, а в таблице **КартотекаОС**, соответствующей агрегированной сущности, – не менее 10.
Замечание. **ОС** – основное средство, **МатОтЛица**, **МОЛ** – материально-ответственные лица.

КартотекаОС

<i>ГруппаОС</i>	<i>ИнвНомОС</i>	<i>НаимОС</i>	<i>ТабНомМОЛ</i>	<i>ДатаВводаВЭксп</i>	<i>БалансСтоим</i>	<i>Износ</i>	<i>КодНормыАморт</i>
-----------------	-----------------	---------------	------------------	-----------------------	--------------------	--------------	----------------------

МатОтЛица

<i>ТабНомМОЛ</i>	<i>ФамилияМОЛ</i>	<i>СтажРаботы</i>
------------------	-------------------	-------------------

ГруппыОС

<i>НомГруппы</i>	<i>НаименованиеГруппы</i>
------------------	---------------------------

НормыАмортизации

<i>КодНормыАморт</i>	<i>НормаАмортизации</i>	<i>НаименНормыАморт</i>
----------------------	-------------------------	-------------------------

Создание таблиц

1. Создать справочники: **МатОтЛица**, **ГруппыОС**, **НормыАмортизации**.
2. Описать структуру таблицы **КартотекаОС**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **КартотекаОС**, организовав поля **ТабНомМОЛ**, **ГруппаОС**, **КодНормыАморт** в виде полей со списком.

5. Заполнить таблицу **КартотекаОС** с помощью созданной формы, выбирая значения полей *ТабНомМОЛ*, *ГруппаОС*, *КодНормыАморт* из списков.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список материально – ответственных лиц, имеющих стаж работы не менее 5 лет.

2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **КартотекаОС**, но объекты бы представлялись своими наименованиями, выбранными из соответствующих справочников. Выполнить сортировку по полю *ДатаВводаВЭксп.*

3. Составить запрос **Запрос3**, результирующая таблица которого должна содержать все поля **Запроса2**, кроме того добавить поле *МесИзнос* с расчетом по формуле: $МесИзнос = \frac{БалансСтоим}{(НормаАмортизации * 12)}$.

Поле *НормаАмортизации* взять из таблицы **НормыАмортизации**. Выполнить сортировку по полям *ГруппаОС*, *НаимОС*.

4. Составить параметрический запрос, отображающий основные средства по заданной группе ОС форме **Запрос3**.

5. Составить запрос с группировкой, отображающий стоимость ОС каждой группы.

6. Создать перекрестный запрос, отображающий стоимость ОС каждой группы, закрепленных за каждым материально-ответственным лицом.

<i>ГруппаОС</i>	<i>Стоимость поступлений</i>	<i>МОЛ 1</i>	<i>МОЛ 2</i>
-----------------	------------------------------	--------------	--------------

Запросы-действия

7. Создать таблицу **КартотекаОСПодробно**, подобную результирующей таблице **Запрос3** с помощью запроса на создание таблицы.

8. Добавить в таблицу **КартотекаОСПодробно** столбец *Организация*, заполнить это поле значением БГАТУ с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **КартотекаОСПодробно** строки, идентичной имеющейся в таблице строке с наибольшим номером, придать новой строке номер на единицу больший номера исходной строки.

10. Составить запрос с параметром на удаление информации об основном средстве, удовлетворяющем некоторому условию из таблицы **КартотекаОС-Подробно**.

SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **КартотекаОС1**, подобной таблице **КартотекаОС**, но добавить числовые поля *МесИзнос* и *ОстСтоимость*.

12. С помощью инструкции SQL описать структуру таблицы **ГруппыОС1**, подобной таблице **ГруппыОС**, но добавить числовые поля *СтоимостьОСГруппы* и *ИзносОСГруппы*.

13. Заполнить таблицу **КартотекаОС1** тремя записями с помощью инструкций SQL (поля *МесИзнос* и *ОстСтоимость* оставить пустыми).

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **КартотекаОС1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **КартотекаОС1**. Добавить в таблицу **КартотекаОС1** строки из таблицы **КартотекаОС** (можно скопировать).

16. С помощью инструкции SQL вывести следующую информацию из таблицы **МатОтЛица**: *ФамилияМОЛ*, *СтажРаботы*.

17. С помощью инструкции SQL вывести информацию таблицы **КартотекаОС**: *ГруппаОС*, *ИнвНомОС*, *НаимОС*, *ФамилияМОЛ* (из таблицы **МатОтЛица**), *БалансСтоим*.

18. С помощью инструкции SQL вывести информацию таблицы **Картотека-ОС**, представив объекты своими наименованиями, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новом материально-ответственном лице в справочник. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую отображать и вводить новые основные средства в таблицу **КартотекаОС**, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются материально-ответственные лица, а в подчиненной – сведения об основных средствах, которые числятся за соответствующим материально-ответственным лицом (по запросу **Запрос3**) с расчетом итоговой стоимости указанных основных средств.

4. Создать форму в виде сводной таблицы, содержащей информацию об основных средствах на предприятии с расчетом общей стоимости ОС по каждой группе, общей стоимости ОС по каждому материально-ответственному лицу и общей стоимости всех основных средств на предприятии.

Создание отчетов

1. Создать отчет об имеющихся в справочнике материально-ответственных лицах по структуре таблицы **МатОтЛица**, добавить поле **СтажРаботыМес**, рассчитываемое по формуле: **СтажРаботыМес = СтажРаботы*12**.

Рассчитать также количество материально-ответственных лиц в справочнике.

2. Создать отчет с группировкой по группам ОС, отображающий информацию об ОС (по **Запрос3**) с получением промежуточного итога по каждой группе ОС и общего итога по всей ведомости в столбце **БалансСтоим**.

3. С помощью *Мастера отчетов* создать отчет с группировкой по материально-ответственным лицам, отображающий информацию об ОС с расчетом общей стоимости ОС, которые числятся за каждым материально-ответственным лицом, и стоимости всех основных средств на предприятии.

Создание модулей VBA

1. Создать программный модуль VBA, рассчитывающий в таблице **КартотекаОС1** поле **ОстСтоимость** по формуле:

$$\text{ОстСтоимость} = \text{БалансСтоим} - \text{Износ}.$$

7. С помощью программного модуля VBA рассчитать значение поля **МесИзнос** таблицы **КартотекаОС1** по формуле:

$$\text{МесИзнос} = \text{БалансСтоим} / (\text{НормаАмортизации} * 12).$$

Поле **НормаАмортизации** взять из таблицы **НормыАмортизации**.

8. Создать программный модуль VBA, выполняющий следующие функции:

- расчет значения поля *МесИзнос* таблицы **КартотекаОС1** по формуле:

$$\text{МесИзнос} = \text{БалансСтоим} / (\text{НормаАмортизации} * 12).$$

Поле *НормаАмортизации* взять из таблицы **НормыАмортизации**;

- расчет поля *ОстСтоимость* таблицы **КартотекаОС1** по формуле:

$$\text{ОстСтоимость} = \text{БалансСтоим} - \text{Износ};$$

- заполнить таблицу **Группы1**, рассчитав значения полей *СтоимостьОС-Группы* и *ИзносОСГруппы* как общую стоимость основных средств, и общий износ основных средств каждой группы ОС, исходя из данных таблицы **КартотекаОС1**.

ВАРИАНТ 4

Создать базу данных, позволяющую вести учет подписки на периодические издания. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

Издания

<i>ИндексИздания</i>	<i>НазвИздания</i>	<i>СтоимостьЗаГод</i>	<i>КодГруппыИзданий</i>
----------------------	--------------------	-----------------------	-------------------------

Подписчики

<i>КодПодписчика</i>	<i>ФамилияПодписчика</i>	<i>Адрес</i>
----------------------	--------------------------	--------------

ГруппыИзданий

<i>КодГруппы</i>	<i>НаименованиеГруппыИзданий</i>	<i>ПроцентСкидки</i>
------------------	----------------------------------	----------------------

Подписка

<i>ГодПодписки</i>	<i>ИндексИздания</i>	<i>КодПодписчика</i>	<i>СрокПодписки(мес)</i>	<i>МестоДоставки</i>
--------------------	----------------------	----------------------	--------------------------	----------------------

Создание таблиц

1. Создать справочники: **ГруппыИзданий**, **Подписчики**.
2. Описать структуры таблиц **Издания**, **Подписка**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Издания**, организовав поле *КодГруппыИзданий* в виде поля со списком.
5. Заполнить таблицу **Издания** с помощью созданной формы, выбирая значения поля *КодГруппыИзданий* из списка.

6. Создать форму для заполнения таблицы **Подписка**, организовав поля **ИндексИздания** и **КодПодписчика** в виде полей со списком.

7. Заполнить таблицу **Подписка** с помощью созданной формы, выбирая поля **ИндексИздания** и **КодПодписчика** из списков.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список изданий, стоимость которых за год менее 50 000 тыс. руб.

2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **Подписка**, но объекты бы представлялись своими наименованиями. Выполнить сортировку по полю **ГодИздания**.

3. Составить запрос **Запрос3**, результирующая таблица которого должна содержать все поля **Запрос2**, кроме того добавить графу **Наименование-ГруппыИзданий** и графу **СтоимостьПодписки** с расчетом по формуле: $СтоимостьЗаГод/12 * СрокПодписки$. Выполнить сортировку по фамилии подписчика.

4. Составить параметрический запрос, отображающий подписку на заданное издание в форме **Запрос3**.

5. Составить запрос с группировкой, отображающий информацию о стоимости подписки каждого подписчика.

6. Создать перекрестный запрос, отображающий стоимость подписки каждого подписчика на каждую группу изданий.

<i>Подписчик</i>	<i>Общая стоимость</i>	<i>Группа 1</i>	<i>Группа 2</i>
------------------	------------------------	-----------------	-----------------

Запросы-действия

7. Создать таблицу **ПодпискаПодробно**, подобную результирующей таблице запроса п. 2 с помощью запроса на создание таблицы.

8. Добавить в таблицу **ПодпискаПодробно** поле **МестоПодписки** и заполнить это поле значением **Почтовое отделение № 5** с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **Подписка** строк подписки подписчика с кодом 1 на следующий год.

10. Составить запрос с параметром на удаление записей о подписке заданного подписчика из таблицы **ПодпискаПодробно**.

SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **Издания1**, подобной таблице **Издания**, но добавить поле **СтоимостьЗаМесяц**.

12. С помощью инструкции SQL описать структуру таблицы **ПодпискаИтог**, содержащей следующие поля: **ГодПодписки**, **ИндексИздания**, **Количество-Подписавшихся**.

13. Заполнить таблицу **Издания1** тремя записями с помощью инструкций SQL (поле **СтоимостьЗаМесяц** оставить пустым).

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **Издания1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **Издания1**.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **ГруппыИзданий**: **НаименованиеГруппыИзданий**, **ПроцентСкидки**.

17. С помощью инструкции SQL вывести информацию таблицы **Издания**, добавив поле **НаименованиеГруппыИзданий** из связанной таблицы **ГруппыИзданий**.

18. С помощью инструкции SQL вывести информацию таблицы **Подписка**, добавив поле **НаименованиеИздания** из связанной таблицы **Издания**, а поле **КодПодписчика** заменить на **ФамилияПодписчика** из таблицы **Подписчики**.

Создание форм

1. Создать форму, позволяющую вносить сведения о новых подписчиках. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую вносить новые сведения о подписке, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются подписчики, подписавшиеся на издания, а в подчиненной – список изданий, на которые подписался данный подписчик, с итоговой стоимостью подписки на год, а также раскрывающийся список по полю **ИндексИздания** для удобства добавления данных.

4. Создать форму в виде сводной таблицы, содержащей информацию о подписке на все издания по всем подписчикам с расчетом общей стоимости подписки по каждому изданию и по каждому подписчику и общей стоимости всей подписки.

Создание отчетов

1. Создать отчет об имеющихся в справочнике изданиях с подсчетом стоимости каждого издания за месяц по формуле: ***СтоимостьЗаМесяц=СтоимостьЗаГод /12*** и общей стоимости за месяц всех изданий.

2. Создать отчет с группировкой по группам изданий, отображающий информацию о подписке (по **Запрос3**) с подсчетом промежуточных итогов стоимости изданий по каждой группе изданий.

3. С помощью *Мастера отчетов* создать отчет с группировкой по подписчикам, отображающий информацию о подписке с подсчетом промежуточных итогов стоимости изданий по каждому подписчику и общей стоимости изданий по всем подписчикам.

Создание модулей VBA

1. Создать программный модуль VBA, рассчитывающий для таблицы **Издания1** значение нового столбца ***СтоимостьЗаМесяц*** по формуле: ***СтоимостьЗаГод /12***.

2. Создать программный модуль VBA, рассчитывающий для таблицы **Издания** значение нового столбца ***НоваяСтоимостьЗаГод*** по формуле:

СтоимостьЗаГод+ СтоимостьЗаГод * ПроцентСкидки (табл. **Группы Изданий**)/100.

3. С помощью программного модуля добавить в таблицу **ПодпискаИтог** информацию за текущий год подписки на основе данных таблицы **Подписка**, произведя расчет общего количества подписавшихся на каждое издание.

ВАРИАНТ 5

Создать базу данных, содержащую сведения о государствах мира. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

Страны

<i>Наименование</i>	<i>Площадь (км²)</i>	<i>Население</i>	<i>ГосЯзык</i>	<i>ДенежнаяЕдиница</i>	<i>ЧастьСвета</i>
<i>Австралия</i>	<i>7682300</i>	<i>17500000</i>	<i>2</i>	<i>1</i>	<i>4</i>
<i>Австрия</i>	<i>803855</i>	<i>7700000</i>	<i>3</i>	<i>2</i>	<i>1</i>
<i>Болгария</i>	<i>110912</i>	<i>9000000</i>	<i>4</i>	<i>3</i>	<i>1</i>
<i>Бутан</i>	<i>46500</i>	<i>700000</i>	<i>5</i>	<i>4</i>	<i>2</i>
<i>Венгрия</i>	<i>93036</i>	<i>10400000</i>	<i>6</i>	<i>5</i>	<i>1</i>
<i>Гамбия</i>	<i>1295</i>	<i>900000</i>	<i>2</i>	<i>6</i>	<i>3</i>
<i>Дания</i>	<i>43092</i>	<i>5100000</i>	<i>7</i>	<i>2</i>	<i>1</i>
<i>Италия</i>	<i>301277</i>	<i>57700000</i>	<i>8</i>	<i>2</i>	<i>1</i>

ЧастиСвета

<i>КодЧастиСвета</i>	<i>НаименованиеЧастиСвета</i>	<i>Полушарие</i>	<i>ПлощадьСушиКвКм</i>
<i>1</i>	<i>Европа</i>	<i>Восточное</i>	
<i>2</i>	<i>Азия</i>	<i>Восточное</i>	
<i>3</i>	<i>Северная Америка</i>	<i>Западное</i>	
<i>4</i>	<i>Австралия</i>	<i>Восточное</i>	

ДенежныеЕдиницы

<i>КодДенЕд</i>	<i>НаименованиеДенЕд</i>
<i>1</i>	<i>Доллар</i>
<i>2</i>	<i>Евро</i>
<i>3</i>	<i>Лев</i>
<i>4</i>	<i>Нгултрум</i>
<i>5</i>	<i>Форинт</i>
<i>6</i>	<i>Даласи</i>

Языки

<i>КодЯзыка</i>	<i>НаименованиеЯзыка</i>	<i>ЯзыковаяГруппа</i>
<i>1</i>	<i>Русский</i>	<i>5</i>
<i>2</i>	<i>Английский</i>	<i>1</i>
<i>3</i>	<i>Немецкий</i>	<i>2</i>
<i>4</i>	<i>Болгарский</i>	<i>5</i>
<i>5</i>	<i>Дзонг-кэ</i>	<i>2</i>
<i>6</i>	<i>Венгерский</i>	<i>2</i>
<i>7</i>	<i>Датский</i>	<i>1</i>
<i>8</i>	<i>Итальянский</i>	<i>3</i>

Языковые Группы

<i>КодЯзГруппы</i>	<i>НаименованиеЯзГруппы</i>
1	<i>Романская</i>
2	<i>Германская</i>
3	<i>Кельтская</i>
4	<i>Балтийская</i>
5	<i>Славянская</i>
6	<i>Иранская</i>
7	<i>Индийская</i>

Создание таблиц

1. Создать справочники: **ЧастиСвета**, **ДенежныеЕдиницы**, **Языковы-еГруппы**.

2. Описать структуру таблицы **Языки**.

3. Описать структуру таблицы **Страны**.

4. Создать связи между таблицами.

5. Создать форму для заполнения таблицы **Языки**, организовав поле **Языко-ваяГруппа** в виде поля со списком.

6. Заполнить таблицу **Языки** с помощью созданной формы, выбирая значения поля **ЯзыковаяГруппа** из списка.

7. Создать форму для заполнения таблицы **Страны**, организовав поля **Гос-Язык**, **ДенежнаяЕдиница**, **ЧастьСвета** в виде полей со списком.

8. Заполнить таблицу **Страны** с помощью созданной формы, выбирая значения полей **ГосЯзык**, **ДенежнаяЕдиница**, **ЧастьСвета** из списков.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список континентов Восточного полушария.

2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **Страны**, но объекты должны быть представлены своими наименованиями. Выполнить сортировку по возрастанию численности населения.

3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запрос2**, кроме того добавить столбец **Наименование-ЯзыковойГруппы**. Поле **Площадь** представить в 1000 км². Выполнить сортировку в алфавитном порядке наименований стран.

4. Составить параметрический запрос, отображающий список стран в форме “Запрос3”, имеющий заданный государственный язык.
5. Составить запрос с группировкой, отображающий информацию об общей площади стран на каждом континенте.
6. Создать перекрестный запрос, отображающий количество стран, говорящих на языке каждой группы для каждого континента.

<i>ЧастьСвета</i>	<i>Количество стран</i>	<i>Группа 1</i>	<i>Группа 2</i>
-------------------	-------------------------	-----------------	-----------------

Запросы-действия

7. Создать таблицу **СтраныПодробно**, подобную результирующей таблице **Запроса2** с помощью запроса на создание таблицы.
8. Добавить в таблицу **ДенежныеЕдиницы** столбец **Примечание** и заполнить это поле значением **Деньги** с помощью запроса на обновление.
9. Создать запрос на добавление еще одной языковой группы с кодом, следующим за последним имеющимся кодом.
10. Составить запрос с параметром на удаление информации о заданной стране из таблицы **СтраныПодробно**.

2.3 SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **ЧастиСвета1**, подобной таблице **ЧастиСвета**.
12. С помощью инструкции SQL описать структуру таблицы **Языки1**, подобную таблице **Языки**, но добавив поле **КоличествоСтран**.
13. Заполнить таблицу **ЧастиСвета1** тремя записями с помощью инструкций SQL.
14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **ЧастиСвета1** с помощью инструкции SQL.
15. Удалить записи, соответствующие определенному условию отбора из таблицы **ЧастиСвета1**.
16. С помощью инструкции SQL вывести следующую информацию из таблицы **Страны**: **Наименование**, **Площадь**, **Население**.
17. С помощью инструкции SQL вывести информацию таблицы **Языки**, добавив поле **НаименованиеЯзГруппы** из связанной таблицы **ЯзыковыеГруппы**.
18. С помощью инструкции SQL вывести информацию таблицы **Страны**, представив поля **ГосЯзык**, **ДенежнаяЕдиница**, **ЧастиСвета1** наименованиями, взятыми из таблиц соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новых языковых группах. На форме предусмотреть надпись с наименованием формы.
2. На форму, позволяющую вносить сведения о новых странах, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.
3. Создать форму с подчиненной, где в главной форме отображаются языки, а в подчиненной – список стран, имеющих государственным данный язык, с итоговым количеством народонаселения этих стран.
4. Создать форму в виде сводной таблицы, содержащей информацию обо всех странах с расчетом общей численности населения стран на каждом континенте, и общей численности населения стран, говорящих на каждом из языков, и суммарной численности населения всех стран.

Создание отчетов

1. Создать отчет об имеющихся в справочнике частях света с подсчетом площади в гектарах каждой части света по формуле:

$$\text{ПлощадьСушиГа} = \text{ПлощадьСушиКвКм} * 100$$

и общей площади суши всех частей света в гектарах.

2. Создать отчет с группировкой по частям света, отображающий информацию о странах (по структуре **Запрос3**) с подсчетом итогового количества населения по каждой части света.
3. С помощью *Мастера отчетов* создать отчет с группировкой по наименованию денежных единиц, отображающий информацию о странах с подсчетом промежуточных итогов по общей площади стран, пользующихся каждой из денежных единиц, и суммарной площади всех стран.

Создание модулей VBA

1. Создать программный модуль VBA, рассчитывающий для таблицы **ЧастиСвета1** значение нового поля **ПлощадьСушиГа** по формуле: **ПлощадьСушиГа=ПлощадьСушиКвКм*100**. Предварительно добавить новое поле в структуру таблицы **ЧастиСвета1**.

2. Создать программный модуль VBA, представляющий поле **Площадь** стран Западного полушария в тыс. км² и заполняющий новое поле **ПлощадьЕдИз** значениями км² и тыс. км².

3. С помощью программного модуля заполнить таблицу **Языки1** данными, рассчитав количество стран, имеющих в качестве государственного языка каждый из языков на основе данных таблицы **Страны**.

ВАРИАНТ 6

Создать базу данных **АлмазыМира**, содержащую информацию о наиболее известных алмазах мира. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

Алмазы

<i>Наименование</i>	<i>СтранаПроисхождения</i>	<i>КогдаНайден</i>	<i>МассаВКаратах</i>
<i>Куллинан</i>	<i>1</i>	<i>1905</i>	<i>3106,0</i>
<i>Эксельсиор</i>	<i>1</i>	<i>1893</i>	<i>971,5</i>
<i>Звезда Сьерра-Леоне</i>	<i>2</i>	<i>1972</i>	<i>968,9</i>
<i>Великий Могол</i>	<i>3</i>	<i>XVII в.</i>	<i>787,0</i>
<i>РекаУоие</i>	<i>2</i>	<i>1945</i>	<i>770,0</i>
<i>Президент Варгас</i>	<i>4</i>	<i>1938</i>	<i>726,6</i>
<i>Джонкер</i>	<i>1</i>	<i>1934</i>	<i>726,0</i>

Страны

<i>КодСтраны</i>	<i>НаименованиеСтраны</i>	<i>ЧастьСвета</i>
<i>1</i>	<i>Южная Африка</i>	<i>1</i>
<i>2</i>	<i>Западная Африка</i>	<i>1</i>
<i>3</i>	<i>Индия</i>	<i>2</i>
<i>4</i>	<i>Бразилия</i>	<i>3</i>

Континенты

<i>КодЧастиСвета</i>	<i>НаименованиеЧастиСвета</i>	<i>Полушарие</i>	<i>ПлощадьСушиКвКм</i>
<i>1</i>	<i>Африка</i>	<i>Восточное</i>	
<i>2</i>	<i>Азия</i>	<i>Восточное</i>	
<i>3</i>	<i>Америка</i>	<i>Западное</i>	
<i>4</i>	<i>Европа</i>	<i>Восточное</i>	

Создание таблиц

1. Создать справочник: **ЧастиСвета**.
2. Описать структуры таблиц **Страны**, **Алмазы**.
3. Создать связи между таблицами.

4. Создать форму для заполнения таблицы **Страны**, организовав поле **ЧастьСвета** в виде поля со списком.

5. Заполнить таблицу **Языки** с помощью созданной формы, выбирая значения поля **ЧастьСвета** из списка.

6. Создать форму для заполнения таблицы **Алмазы**, организовав поле **СтранаПроисхождения** в виде поля со списком.

7. Заполнить таблицу **Алмазы** с помощью созданной формы, выбирая значения поля **СтранаПроисхождения** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список стран заданного континента.

2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **Алмазы**, но объекты должны быть представлены своими наименованиями. Выполнить сортировку по убыванию массы.

3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запроса2**, кроме того, добавить столбец **НаименованиеЧастиСвета**, а так же столбец **МассаВГрамах**, рассчитанный по формуле: $МассаВКаратах * 0,2$. Выполнить сортировку в алфавитном порядке наименований алмазов.

4. Составить параметрический запрос, подобный **Запросу3**, но отображающий список алмазов, найденных на заданном континенте.

5. Составить запрос с группировкой, отображающий общую массу алмазов в каратах на каждом континенте.

6. Создать перекрестный запрос, отображающий количество алмазов, найденных в каждой из стран по периодам.

<i>Страна</i>	<i>Количество алмазов</i>	<i>Период1</i>	<i>Период2</i>
---------------	---------------------------	----------------	----------------

Запросы на выборку

7. Создать таблицу **АлмазыПодробно**, подобную результирующей таблице запроса п. 2 с помощью запроса на создание таблицы.

8. Добавить в таблицу **ЧастиСвета** столбец **Примечание** и заполнить это поле значением **Алмазы на континенте** с помощью запроса на обновление.

9. Создать запрос на добавление еще одной страны с кодом, следующим за последним имеющимся кодом.

10. Составить запрос с параметром на удаление введенной в предыдущем задании страны.

SQL запросы

11. С помощью инструкций SQL описать структуру таблицы **ЧастиСвета1**, подобную таблице **ЧастиСвета1**, и структуру таблицы **Алмазы1**, подобную таблице **Алмазы**.

12. С помощью инструкции SQL описать структуру таблицы **Страны1**, подобную таблице **Страны**, но добавив поле *КоличествоАлмазов*.

13. Заполнить таблицу **ЧастиСвета1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **ЧастиСвета1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **ЧастиСвета1**.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Алмазы**: *Наименование, КогдаНайден, МассаВКаратах*.

17. С помощью инструкции SQL вывести информацию таблицы **Страны**, добавив поле *НаименованиеЧастиСвета* из связанной таблицы **ЧастиСвета**.

18. С помощью инструкции SQL вывести информацию таблицы **Алмазы**, представив поле *СтранаПроисхождения* наименованием, взятым из таблицы **Страны**, и добавив поле *НаименованиеЧастиСвета* из таблицы **ЧастиСвета**.

Создание форм

1. Создать форму, позволяющую вносить сведения о новых частях света. На форме предусмотреть надпись с наименованием формы.

2. На форме, позволяющей вносить сведения о новых алмазах, создать кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются части света, а в подчиненной – список алмазов, найденных в странах данной части света с указанием страны, получить итоговый вес алмазов каждого списка.

4. Создать форму в виде сводной таблицы, содержащей информацию обо всех алмазах с расчетом общей массы всех алмазов, обнаруженных в каждой из стран, и суммарной массы всех алмазов.

Создание отчетов

5. Создать отчет об имеющихся в справочнике частях света с подсчетом площади каждой части света в тыс. км² по формуле: **ПлощадьСушиТысКвКм=ПлощадьСушиКвКм/1000** и общей площади суши всех частей света в тыс. км².

6. Создать отчет с группировкой по частям света, отображающий информацию об алмазах (по **Запрос3**) с подсчетом итоговой массы всех алмазов по каждой части света и общей массы всех зарегистрированных алмазов.

7. С помощью *Мастера отчетов* создать отчет с группировкой по странам, отображающий информацию об алмазах с подсчетом количества алмазов, обнаруженных в каждой стране, и общего количества алмазов.

Создание модулей VBA

1. Создать программный модуль, рассчитывающий массу каждого алмаза в граммах по формуле: **МассаВГрамах = МассаВКаратах*0,2** и размещающий рассчитанную величину в предварительно созданный столбец таблицы **Алмазы**.

2. С помощью программного модуля выбрать из таблицы **Алмазы** информацию об алмазах, найденных в странах Восточного полушария и записать ее в таблицу **Алмазы1**.

3. С помощью программного модуля заполнить таблицу **Страны1** данными, рассчитав количество алмазов, найденных в каждой стране, на основе таблицы **Алмазы**.

ВАРИАНТ 7

Создать базу данных, позволяющую вести учет уплаты налогов предприятиями. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

УплатаНалогов

Год	Кварталл	КодПредприятия	КодНалога	СуммаДляНалога
-----	----------	----------------	-----------	----------------

Предприятия

КодПредприятия	НаименованиеПредприятия	АдресПредприятия	РайонГорода
----------------	-------------------------	------------------	-------------

Налоги

КодНалога	НаименованиеНалога	Процент
-----------	--------------------	---------

Районы

КодРайона	НаименованиеРайона	Коэффициент
-----------	--------------------	-------------

Создание таблиц

1. Создать справочники: **Налоги**, **Районы**.
2. Описать структуры таблиц **Предприятия**, **УплатаНалогов**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Предприятия**, организовав поле **РайонГорода** в виде поля со списком.
5. Заполнить таблицу **Предприятия** с помощью созданной формы, выбирая значения полей **РайонГорода** и **КодГруппыИзданий** из списка.
6. Создать форму для заполнения таблицы **УплатаНалогов**, организовав поля **КодПредприятия**, **КодНалога** в виде полей со списком.
7. Заполнить таблицу **УплатаНалогов** с помощью созданной формы, выбирая поля **КодПредприятия**, **КодНалога** из списков.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список налогов из общего перечня, процент которых менее 10.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **УплатаНалогов**, но объекты бы представлялись своими наименованиями. Выполнить сортировку по наименованию налога.
3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запроса2**, кроме того добавить поле **СуммаНалога**, рас-

считываемое по формуле: *СуммаДляНалога* Процент* Коэффициент*, а также поле *НаименованиеРайона*. Выполнить сортировку по наименованию предприятия.

4. Составить параметрический запрос, отображающий информацию по заданному году и кварталу.

5. Составить запрос с группировкой, отображающий информацию об общей сумме по каждому налогу.

6. Создать перекрестный запрос, отображающий общую сумму налогов за каждый год по каждому району.

<i>Год</i>	<i>ОбщаяСуммаНалогов</i>	<i>Район 1</i>	<i>Район 2</i>
------------	--------------------------	----------------	----------------

Запросы-действия

7. Создать таблицу **УплатаНалоговПодробно**, подобную результирующей таблице **Запрос2** с помощью запроса на создание таблицы.

8. Добавить в таблицу **УплатаНалоговПодробно** поле *Район* и заполнить это поле значением **Центральный** с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **УплатаНалогов** строк по предприятию с кодом 1, касающихся следующего года.

10. Составить запрос с параметром на удаление записей по заданному предприятию из таблицы **УплатаНалоговПодробно**.

SQL запросы

11. С помощью инструкций SQL описать структуру таблицы **УплатаНалогов1**, подобной таблице **УплатаНалогов**.

12. С помощью инструкции SQL описать структуру таблицы **Налоги1**, подобную таблице **Налоги**, но добавив поле *ОбщаяСумма*.

13. Заполнить таблицу **УплатаНалогов1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **УплатаНалогов1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **УплатаНалогов1**. Добавить в таблицу **УплатаНалогов1** две записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Районы**: *КодРайона*, *Наименование*.

17. С помощью инструкции SQL вывести информацию таблицы **Предприятия**, добавив поле *НаименованиеРайона* из связанной таблицы **Районы**.

18. С помощью инструкции SQL вывести информацию таблицы **УплатаНалогов**, представив предприятия и налоги не кодами (как в таблице), а наименованиями, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новом налоге. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую вносить сведения об уплате налогов, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются предприятия, а в подчиненной – список налогов, уплаченных каждым предприятием, с подсчетом итоговой суммы налогов, уплаченных каждым предприятием.

4. Создать форму в виде сводной таблицы, содержащей информацию обо всех уплаченных налогах всеми предприятиями с расчетом общей суммы налогов, уплаченных каждым предприятием, общей суммы налога каждого вида, уплаченной всеми предприятиями, и общей суммы всех налогов.

Создание отчетов

1. Создать отчет об имеющихся в справочнике налогах с подсчетом доли каждого налога по формуле: $Доля = \frac{Процент}{100}$ и количества налогов в справочнике.

2. Создать отчет с группировкой по предприятиям, отображающий информацию об уплате налогов (по **Запрос3**) с подсчетом итоговой суммы налогов, уплаченных каждым предприятием.

3. С помощью *Мастера отчетов* создать отчет с группировкой по районам города, отображающий информацию об уплате налогов с расчетом суммы налогов, уплаченных предприятиями каждого района, и общей суммы всех уплаченных налогов.

Создание модулей VBA

1. С помощью модуля VBA уменьшить во всех записях таблицы **УплатаНалогов1** поле **СуммаДляНалога** на один процент.

2. Создать программный модуль VBA, рассчитывающий для таблицы **УплатаНалогов1** новую графу **СуммаНалога** по формуле: **СуммаДляНалога*Коэффициент**.

3. С помощью программного модуля VBA заполнить таблицу **НалогИ1** данными, рассчитав общую уплаченную сумму для каждого вида налога, используя информацию таблицы **УплатаНалогов**.

ВАРИАНТ 8

Создать базу данных, позволяющую вести учет выработки продукции на конвейере за сутки. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

ВыработкаПродукции

<i>КодЦеха</i>	<i>НомерУчастка</i>	<i>ТабНомРабочего</i>	<i>КодДетали</i>	<i>КоличествоВыработки</i>
----------------	---------------------	-----------------------	------------------	----------------------------

Продукция

<i>КодДетали</i>	<i>НаименованиеДетали</i>	<i>РасценкаЗаШтуку</i>	<i>НормаВремени</i>
	<i>Вал</i>		
	<i>Втулка</i>		
	<i>Шатун</i>		

Рабочие

<i>ТабНомер</i>	<i>ФамилияРабочего</i>	<i>КоэффициентЗаРазряд</i>
-----------------	------------------------	----------------------------

Цеха

<i>КодЦеха</i>	<i>НаименованиеЦеха</i>
----------------	-------------------------

Создание таблиц

1. Создать справочник: **Продукция, Рабочие, Цеха**.

2. Описать структуру таблицы **ВыработкаПродукции**.

3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **ВыработкаПродукции**, организовав поля *КодЦеха*, *ТабНомРабочего*, *КодДетали* в виде полей со списком.
5. Заполнить таблицу **ВыработкаПродукции** с помощью созданной формы, выбирая значения полей *КодЦеха*, *ТабНомРабочего*, *КодДетали* из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список рабочих, начиная с четвертого разряда.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблицы **ВыработкаПродукции**, но объекты должны быть представлены своими наименованиями. Выполнить сортировку по наименованию цеха.
3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запрос2**, кроме того добавить поле *Сумма*, рассчитываемое по формуле: *РасценкаЗаШтуку * КоличествоВыработки* и поле *Фактические-Часы*, рассчитываемое по формуле: *НормаВремени * КоличествоВыработки*. Выполнить сортировку по фамилии рабочего.
4. Составить параметрический запрос по форме **Запрос3**, отображающий информацию по заданному цеху.
5. Составить запрос с группировкой, отображающий информацию об общей сумме и суммарным фактическим часам выработки по каждой детали.
6. Создать перекрестный запрос, отображающий общую сумму выработки по каждому цеху по каждой детали.

<i>Цех</i>	<i>Общая сумма</i>	<i>Деталь1</i>	<i>Деталь 2</i>
------------	--------------------	----------------	-----------------

Запросы-действия

7. Создать таблицу **ВыработкаПродукцииПодробно**, подобную результирующей таблице запроса п. 2 с помощью запроса на создание таблицы.

8. Добавить в таблицу **ВыработкаПродукцииПодробно** поле *Завод*, заполнить это поле значением **Интеграл** с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **ВыработкаПродукции** строк по рабочему с табельным номером 1.

10. Составить запрос с параметром на удаление записей по заданному цеху из таблицы **ВыработкаПродукцииПодробно**.

11. Создать программный модуль VBA, рассчитывающий для таблицы **ВыработкаПродукции** новый столбец *СтоимостьВыработки* по формуле:
*РасценкаЗаШтуку * КоличествоВыработки * КоэффициентЗаРазряд*.

SQL запросы

12. С помощью инструкций SQL описать структуру таблицы **ВыработкаПродукции1**, подобную таблице **ВыработкаПродукции**.

13. С помощью инструкции SQL описать структуру таблицы **Продукция1**, подобную таблице **Продукция**, но добавив поле *КоличествоВыпущено*.

14. Заполнить таблицу **ВыработкаПродукции1** тремя записями с помощью инструкций SQL.

15. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **ВыработкаПродукции1** с помощью инструкции SQL.

16. Удалить записи, соответствующие определенному условию отбора из таблицы **ВыработкаПродукции1**. Добавить в таблицу **УплатаНалогов1** две записи по своему усмотрению.

17. С помощью инструкции SQL вывести следующую информацию из таблицы **Рабочие**: *ФамилияРабочего*, *КоэффициентЗаРазряд*.

18. С помощью инструкции SQL вывести информацию таблицы **ВыработкаПродукции**, добавив поле *ФамилияРабочего* из связанной таблицы **Рабочие**.

19. С помощью инструкции SQL вывести информацию таблицы **ВыработкаПродукции**, представив цеха, рабочих и детали не кодами (как в таблице), а наименованиями, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новом рабочем. На форме предусмотреть надпись с наименованием формы.

2. На форму, созданную ранее по таблице **ВыработкаПродукции**, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются цеха, а в подчиненной – список произведенных деталей каждым цехом, с подсчетом итоговых сумм в колонках **Сумма** и **ФактическиеЧасы** по каждому цеху.

4. Создать форму в виде сводной таблицы, содержащей информацию о выработке продукции на конвейере за сутки всеми рабочими по всем деталям с расчетом общей стоимости деталей, изготовленных каждым рабочим, общей стоимости деталей каждого типа, изготовленных всеми рабочими, общей стоимости всех изготовленных деталей.

Создание отчетов

1. Создать отчет об имеющихся в справочнике деталях с подсчетом поля **РасценкаЗаШтукуТыс** для каждой детали по формуле: $\text{РасценкаЗаШтукуТыс} = \text{РасценкаЗаШтуку} / 1000$ и количества деталей в справочнике.

2. Создать отчет с группировкой по цехам, отображающий информацию о выпуске продукции по структуре **Запрос3** с подсчетом промежуточных итогов в столбцах **Сумма** и **ФактическиеЧасы** по каждому цеху.

3. С помощью *Мастера отчетов* создать отчет с группировкой по видам деталей, отображающий информацию о выпуске деталей с расчетом общей стоимости деталей каждого вида, общей стоимости всех изготовленных деталей.

Создание модулей VBA

1. С помощью модуля VBA во всех записях таблицы **Рабочие** увеличить значение поля **КоэффициентЗаРазряд** на 0,1.

2. Создать программный модуль VBA, рассчитывающий для таблицы **ВыработкаПродукции** новый столбец **СтоимостьВыработки** по формуле: $\text{РасценкаЗаШтуку} * \text{КоличествоВыработки} * \text{КоэффициентЗаРазряд}$.

3. С помощью программного модуля VBA заполнить таблицу **Продукция1** данными, рассчитав общее количество деталей каждого вида, изготовленных всеми рабочими (поместить в поле **КоличествоВыпущено**), использовать информацию таблицы **ВыработкаПродукции**.

ВАРИАНТ 9

Для анализа потребления основных продуктов питания на душу населения в год, кг, по странам СНГ (Беларусь, Армения, Азербайджан, Киргизия, Таджикистан, Казахстан, Россия, Украина) за последние 3 года необходимо собрать следующие данные: по мясу и мясопродуктам, молоку и молочным продуктам, растительному маслу, картофелю, овоще-бахчевым и плодам, ягодам, хлебопродуктам. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

Потребление

<i>Год</i>	<i>Страна</i>	<i>ГруппаПродуктов</i>	<i>ПотреблениеКоличествоКг</i>
------------	---------------	------------------------	--------------------------------

ГруппыПродуктов

<i>КодГрПродуктов</i>	<i>НаименованиеГрПродуктов</i>	<i>Примечание</i>
-----------------------	--------------------------------	-------------------

Страны

<i>КодСтраны</i>	<i>НаименованиеСтраны</i>	<i>ЧастьСвета</i>	<i>ПлощадьСушиКвКм</i>
------------------	---------------------------	-------------------	------------------------

ЧастиСвета

<i>КодЧастиСвета</i>	<i>НаименованиеЧастиСвета</i>
----------------------	-------------------------------

Создание таблиц

1. Создать справочник: **ГруппыПродуктов**, **ЧастиСвета**.
2. Описать структуры таблиц **Страны**, **Потребление**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Страны**, организовав поле **ЧастьСвета** в виде поля со списком.
5. Заполнить таблицу **Страны** с помощью созданной формы, выбирая значения поля **ЧастьСвета** из списка.
6. Создать форму для заполнения таблицы **Потребление**, организовав поля **Страна**, **ГруппаПродуктов** в виде полей со списком.
7. Заполнить таблицу **Потребление** с помощью созданной формы, выбирая значения полей **Страна**, **ГруппаПродуктов** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список стран, расположенных в Европе.

2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблицы **Потребление**, но объекты должны быть представлены своими наименованиями. Выполнить сортировку по полю **Год**.

3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запрос2**, кроме того добавить поле **Наименование-ЧастиСвета** и поле **ПотреблениеКоличествоТ**, рассчитываемое по формуле: **Потребление-КоличествоКг/1000**. Выполнить сортировку по наименованию страны.

4. Составить запрос с группировкой, отображающий информацию об общем количестве потребляемых продуктов в каждой стране.

5. Составить параметрический запрос, отображающий информацию о заданной группе продуктов по структуре **Запрос3**.

6. Создать перекрестный запрос, отображающий общее количество потребляемых продуктов каждой группы в каждой стране.

<i>Страна</i>	<i>ОбщееКоличество</i>	<i>ГруппаПродуктов1</i>	<i>ГруппаПродуктов 2</i>
---------------	------------------------	-------------------------	--------------------------

Запросы на выборку

7. Создать таблицу **ПотреблениеПодробно**, подобную результирующей таблице запроса п. 2 с помощью запроса на создание таблицы **Потребление**.

8. Добавить в таблицу **ПеречисленияПодробно** поле **Примечание**, заполнить это поле значением **Анализ** с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **Потребление** строк по следующему календарному году.

10. Составить запрос с параметром на удаление записей по заданной стране из таблицы **ПотреблениеПодробно**.

SQL запросы

11. С помощью инструкций SQL описать структуру таблицы **Потребление1** подобную таблице **Потребление**, но добавить поле **ПотреблениеКоличество-ТысКг**.

12. С помощью инструкции SQL описать структуру таблицы **Группы-Продуктов1**, содержащей поля: **КодГрПродуктов**, **НаименованиеГрПродуктов**, **ВсегоПотребление**.

13. Заполнить таблицу **Потребление1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **Потребление1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **Потребление1**. Добавить в таблицу **Потребление1** три записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **ГруппыПродуктов**: **КодГрПродуктов**, **НаименованиеГрПродуктов**.

17. С помощью инструкции SQL вывести информацию таблицы **Страны**, добавив поле **НаименованиеЧастиСвета** из связанной таблицы **ЧастиСвета**.

18. С помощью инструкции SQL вывести информацию таблицы **Потребление**, представив страны и группы продуктов не кодами (как в таблице), а наименованиями, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новых группах продуктов в справочник. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую отражать новые сведения о потреблении продукции нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются группы продуктов, а в подчиненной – информация о потреблении групп продуктов по странам, с подсчетом итога по каждой стране в столбце **ПотреблениеКоличествоКг**.

4. Создать форму в виде сводной таблицы, содержащей информацию о потреблении основных продуктов питания на душу населения в год по странам СНГ, с расчетом общего количества потребляемой продукции каждой группы всеми анализируемыми странами и общего количества потребляемых товаров в каждой стране, а также общего количества всех потребляемых товаров.

Создание отчетов

1. Создать отчет об имеющихся в справочнике частях света с подсчетом площади суши каждой части света в тысячах квадратных километров по формуле: $\text{ПлощадьСушиТысКвКм} = \text{ПлощадьСушиКвКм} / 1000$ и общей площади суши всех частей света в тыс. км².

2. Создать отчет с группировкой по группам продуктов, отображающий информацию о потреблении основных групп продуктов питания (по **Запрос3**) с получением промежуточного итога по каждой группе продуктов в столбце **ПотреблениеКоличествоКг**.

3. С помощью *Мастера отчетов* создать отчет с группировкой по частям света, отображающий информацию о потреблении основных групп продуктов питания с расчетом общего количества потребляемых продуктов странами каждой части света, общего количества потребления каждой группы продуктов всеми рассматриваемыми странами и общего количества всех потребляемых продуктов.

Создание модулей VBA

1. С помощью программного модуля VBA рассчитать новый столбец **ПотреблениеКоличествоТысКг** таблицы **Потребление1** по формуле:

$$\text{ПотреблениеКоличествоТысКг} = \text{ПотреблениеКоличествоКг} / 1000.$$

2. Создать программный модуль VBA, представляющий поле **Потребление-КоличествоКг** таблицы **Потребление1** в тыс. кг. только для стран Европы и заполняющий новое поле **ПотреблениеЕдИз** указанной таблицы значениями кг и тыс.кг.

3. С помощью программного модуля VBA заполнить таблицу **ГруппыПродуктов1** данными, рассчитав общее количество потребляемых продуктов каждой группы (поместить в поле **ВсегоПотребление**), использовать информацию таблицы **Потребление**.

ВАРИАНТ 10

Создать базу данных для учета количества вакантных мест в учреждениях и организация города по районам. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10. Площадь района привести в м².

Вакансии

<i>Год</i>	<i>Месяц</i>	<i>Район</i>	<i>Организация</i>	<i>Должность</i>	<i>МинОклад</i>	<i>МаксОклад</i>
------------	--------------	--------------	--------------------	------------------	-----------------	------------------

Районы

<i>КодРайона</i>	<i>НаименованиеРайона</i>	<i>ПредседательРайисп</i>	<i>Коэффициент</i>	<i>Площадь</i>
------------------	---------------------------	---------------------------	--------------------	----------------

Организации

<i>КодОрганизации</i>	<i>НаименованиеОрганизации</i>	<i>КоличествоСотрудников</i>
-----------------------	--------------------------------	------------------------------

Должности

<i>КодДолжности</i>	<i>НаименованиеДолжности</i>
---------------------	------------------------------

Создание таблиц

1. Создать справочники: **Районы**, **Организации**, **Должности**.
2. Описать структуру таблицы **Вакансии**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Вакансии**, организовав поля **Район**, **Организация**, **Должность** в виде полей со списком.
5. Заполнить таблицу **Вакансии** с помощью созданной формы, выбирая значения полей **Район**, **Организация**, **Должность** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список мелких организаций, количество сотрудников которых менее 20.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблицы **Вакансии**, но объекты должны

быть представлены своими наименованиями. Выполнить сортировку по полям **Год, Месяц, Должность**.

3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запрос2**, кроме того добавить поле **РазницаОкладов**, рассчитываемое по формуле: **МаксОклад – МинОклад**. Выполнить сортировку по полям **Год, Месяц, Район**.

4. Составить запрос с группировкой, отображающий информацию об общем количестве вакантных должностей по районам.

5. Составить параметрический запрос, отображающий информацию о вакансиях по заданной должности (по структуре **Запроса3**).

6. Создать перекрестный запрос, отображающий общее количество вакантных должностей по каждой организации по каждому району

Район	ОбщееКоличествоВакансий	Организация 1	Организация 2
--------------	--------------------------------	----------------------	----------------------

Запросы-действия

7. Создать таблицу **ВакансииПодробно**, подобную результирующей таблице запроса п. 2 с помощью запроса на создание таблицы.

8. Добавить в таблицу **ВакансииПодробно** поле **Город**, заполнить это поле значением Минск с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **ВакансииПодробно** строк, соответствующих последнему заполненному месяцу со следующим значением поля **Месяц**.

10. Составить запрос с параметром на удаление записей по заданному району из таблицы **ВакансииПодробно**.

SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **Вакансии1**, подобную таблице **Вакансии**, но добавить числовое поле **РазницаОкладов**.

12. С помощью инструкции SQL описать структуру таблицы **Районы1**, подобную таблице **Районы**, но добавить поле **ВсегоВакансий**.

13. Заполнить таблицу **Вакансии1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **Вакансии1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **Вакансии1**. Добавить в таблицу **Вакансии1** три записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Районы**: *НаименованиеРайона, Коэффициент, ПлощадьКвКм*.

17. С помощью инструкции SQL вывести следующие поля таблицы **Вакансии**: *Год, Месяц, Организация, Должность, МинОклад, МахОклад*, кроме того добавить поле *НаименованиеДолжности* из связанной таблицы **Должности**.

18. С помощью инструкции SQL вывести информацию таблицы **Вакансии**, представив районы, организации и должности не кодами (как в таблице), а наименованиями, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новой организации в справочник. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую отображать и вводить новые сведения о вакансиях нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются должности, а в подчиненной – сведения о вакансиях во всех организациях, с расчетом общего количества вакансий по каждой должности.

4. Создать форму в виде сводной таблицы, содержащей информацию о количестве вакантных рабочих мест в учреждениях и организациях города по районам, с расчетом общего количества вакантных рабочих мест в каждом районе, общего количества вакантных рабочих мест по каждой должности и общего количества рабочих мест всего.

Создание отчетов

1. Создать отчет об имеющихся в справочнике районах города с подсчетом площади в тыс. км² каждого района по формуле: *ПлощадьТысКвКм = Площадь/1000* и общей площади всех районов в тыс. км².

2. Создать отчет с группировкой по должностям, отображающий информацию о вакантных должностях (по структуре **Запрос3**) с получением промежуточного итога по каждой должности в столбцах **МинОклад, МаксОклад**.

3. С помощью *Мастера отчетов* создать отчет с группировкой по районам, отображающий информацию о вакантных рабочих местах с получением промежуточного итога: количества вакантных рабочих мест по каждому району и общего количества всех вакантных рабочих мест в городе.

Создание модулей VBA

3. С помощью программного модуля VBA рассчитать значение поля **РазницаОкладов** таблицы **Вакансии1** по формуле: **МаксОклад – МинОклад**.

4. Создать программный модуль VBA, пересчитывающий для таблицы **Вакансии** значения полей **МинОклад** и **МаксОклад** формулам:

МинОклад* Коэффициент (из справочника **Районы**);

МаксОклад* Коэффициент (из справочника **Районы**).

5. С помощью программного модуля VBA заполнить таблицу **Районы1** данными, рассчитав количество вакантных рабочих мест по каждому району (поместить в поле **ВсегоВакансий**), на основе информации таблицы **Вакансии**.

ВАРИАНТ 11

Бригады строительного управления работают на строительстве некоторых объектов. Создать базу данных для учета рабочего времени по месяцам. Количество записей в справочниках должно быть не менее 3-4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

Табель

Год	Месяц	Объект	Бригада	Рабочий	КоличествоОтрабДней	ДневнойТариф
-----	-------	--------	---------	---------	---------------------	--------------

Бригады

КодБригады	НаименованиеБригады	Бригадир
------------	---------------------	----------

Рабочие

ТабНомерРабочего	ФамилияРабочего	ГодРождения
------------------	-----------------	-------------

Объекты

КодОбъекта	НаименованиеОбъекта	Коэффициент
------------	---------------------	-------------

Создание таблиц

1. Создать справочники: **Бригады, Рабочие, Объекты**.
2. Описать структуру таблицы **Табель**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Табель**, организовав поля **Объект, Бригада, Рабочий** в виде полей со списком.
5. Заполнить таблицу **Табель** с помощью созданной формы, выбирая значения полей **Объект, Бригада, Рабочий** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список рабочих моложе 20 лет.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблицы **Табель**, но объекты должны быть представлены своими наименованиями. Выполнить сортировку по полям **Год, Месяц, Рабочий**.
3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запроса2**, кроме того добавить поле **Заработано**, рассчитываемое по формуле: $ДневнойТариф * КоличествоОтрабДней * Коэффициент$. Выполнить сортировку по полям **Год, Месяц, Объект**.
4. Составить запрос с группировкой, отображающий информацию об общем количестве дней, отработанных на каждом объекте.
5. Составить параметрический запрос, отображающий информацию о работе заданного рабочего по структуре запроса **Запрос 3**.
6. Создать перекрестный запрос, отображающий Общую заработанную сумму каждой бригадой по каждому объекту.

<i>Объект</i>	<i>ОбщаяЗаработаннаяСумма</i>	<i>Бригада 1</i>	<i>Бригада 2</i>
---------------	-------------------------------	------------------	------------------

Запросы-действия

7. Создать таблицу **ТабельПодробно** с помощью запроса на создание таблицы, подобную результирующей таблице **Запрос3**.

8. Добавить в таблицу **ТабельПодробно** поле **Строительное-Управление**, заполнить это поле значением СУ-5 с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **ТабельПодробно** строк, соответствующих последнему заполненному месяцу со следующим значением поля **Месяц**.

10. Составить запрос с параметром на удаление записей по заданному объекту из таблицы **ТабельПодробно**.

SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **Табель1**, подобную таблице **Табель**, но добавить поле **СуммаЗаработано**.

12. С помощью инструкции SQL описать структуру таблицы **Объекты1**, подобную таблице **Объекты**, но добавить поле **ВсегоЗаработано**.

13. Заполнить таблицу **Табель1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **Табель1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **Табель1**. Добавить в таблицу **Табель1** три записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Объекты**: **НаименованиеОбъекта**, **Коэффициент**.

17. С помощью инструкции SQL вывести следующие поля таблицы **Табель**: **Год**, **Месяц**, **Рабочий**, **КоличествоОтрабДней**, **ДневнойТариф**, кроме того добавить поле **ФамиляРабочего** из связанной таблицы **Рабочие**.

18. С помощью инструкции SQL вывести информацию таблицы **Табель**, представив объекты, бригады и рабочих не кодами (как в таблице), а наименованиями, взятыми из соответствующих справочников

Создание форм

5. Создать форму, позволяющую вносить сведения о новом объекте строительства в справочник. На форме предусмотреть надпись с наименованием формы.

6. На форму, позволяющую отражать новые строки табеля нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

7. Создать форму с подчиненной, где в главной форме отображаются объекты, а в подчиненной – список рабочих, строивших объект, с указанием количества отработанных дней, дневного тарифа и расчетом суммы **Заработано** по формуле: $\text{Заработано} = \text{ДневнойТариф} * \text{КоличествоОтрабДней}$. Подсчитать итог по каждому объекту в колонке **Заработано**.

8. Создать форму в виде сводной таблицы, содержащей информацию для учета рабочего времени рабочих, работающих в бригадах на строительстве объектов, с расчетом общей суммы заработанных денег всеми бригадами на каждом объекте, общей суммы заработанных денег по каждой бригаде на всех объектах и итоговой суммы заработанных денег всего.

Создание отчетов

1. Создать отчет об имеющихся в справочнике рабочих с подсчетом возраста каждого рабочего по формуле: $\text{Возраст} = (\text{Текущий год}) - \text{ГодРождения}$ и среднего возраста рабочих.

2. Создать отчет с группировкой по объектам, отображающий табель (по структуре **Запроса3**) с получением промежуточного итога по каждому объекту в столбце **Заработано**.

3. С помощью *Мастера отчетов* создать отчет с группировкой по рабочим, отображающий информацию для учета рабочего времени рабочих, работающих в бригадах на строительстве объектов, с расчетом общей суммы, заработной каждым рабочим во всех бригадах (промежуточный итог), общей заработной суммы всеми рабочими.

Создание модулей VBA

1. Создать программный модуль VBA, рассчитывающий для таблицы **Табель1** новый столбец **СуммаЗаработано** по формуле:

$\text{ДневнойТариф} * \text{КоличествоОтрабДней}$.

2. Создать программный модуль VBA, рассчитывающий для таблицы **Табель1** новый столбец *СуммаЗаработано* по формуле:

ДневнойТариф КоличествоОтрабДней*Коэффициент* (из справочника объектов).

3. С помощью программного модуля VBA заполнить таблицу **Объекты1** данными, рассчитав общую заработанную сумму на каждом объекте (поместить в поле *ВсегоЗаработано*), на основе информации таблицы **Табель**.

ВАРИАНТ 12

Информация о проживании студентов в общежитиях университета

Создать базу данных, позволяющую вести учет проживания студентов в общежитиях. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

ЗаселениеОбщежитие

<i>КодОбщежития</i>	<i>КодСтудента</i>	<i>НомерКомнаты</i>	<i>ДатаЗаселения</i>	<i>ДатаВыселения</i>
---------------------	--------------------	---------------------	----------------------	----------------------

Студенты

<i>КодСтудента</i>	<i>ФИОСтудента</i>	<i>ГодРождения</i>	<i>КодФакультета</i>
--------------------	--------------------	--------------------	----------------------

Факультеты

<i>КодФакультета</i>	<i>НазваниеФакультета</i>
----------------------	---------------------------

Общежития

<i>КодОбщежития</i>	<i>Адрес</i>	<i>КоличествоМест</i>
---------------------	--------------	-----------------------

Создание таблиц

1. Создать справочники: **Факультеты**, **Общежития**.
2. Описать структуру таблиц **ЗаселениеОбщежитие**, **Студенты**.
3. Создать связи между таблицами.

4. Создать форму для заполнения таблицы **Студенты**, организовав поле **КодФакультета** в виде поля со списком.

5. Заполнить таблицу **Студенты** с помощью созданной формы, выбирая значение поля **КодФакультета** из списка.

6. Создать форму для заполнения таблицы **ЗаселениеОбщежитие**, организовав поля **КодОбщежития**, **КодСтудента** в виде полей со списком.

7. Заполнить таблицу **ЗаселениеОбщежитие** с помощью созданной формы, выбирая значения полей **КодОбщежития**, **КодСтудента** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, отображающий список несовершеннолетних студентов. Данные вывести в алфавитном порядке ФИО студентов.

2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **ЗаселениеОбщежитие**, но студента представить фамилией и добавить адрес общежития. Строки вывести в алфавитном порядке кода общежития.

3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запрос2**, кроме того добавить столбец **НазваниеФакультета** и создать расчетный столбец с именем **Проживает**, в котором рассчитать, какое время (количество дней) студент проживал в общежитии.

4. Составить параметрический запрос, отображающий информацию о студентах, проживающих в заданном общежитии.

5. Составить запрос с группировкой, отображающий информацию о количестве студентов, проживающих в каждом общежитии.

6. Создать перекрестный запрос, отображающий количество студентов каждого факультета, проживающих в каждом общежитии.

Факультет	ОбщееКоличество	Общежитие1	...	ОбщежитиеN
------------------	------------------------	-------------------	------------	-------------------

Запросы-действия

7. Создать таблицу **ЗаселениеОбщежитиеВремя**, подобную результирующей таблице **Запрос2** с помощью запроса на создание таблицы.

8. Добавить в таблицу **ЗаселениеОбщежитиеВремя** столбец **Состояние-Комнаты** и заполнить это поле значением **Хорошее** с помощью запроса на обновление.

9. Создать запрос на добавление в таблицу **Студенты** новой строки с кодом, следующим за последним имеющимся кодом.

10. Составить запрос с параметром на удаление записей по заданному студенту из таблицы **ЗаселениеОбщежитиеВремя**.

2.3 SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **Факультеты1**, подобную таблице **Факультеты**, но добавить числовое поле **Количество-СтудОбщ**.

12. С помощью инструкции SQL описать структуру таблицы **ЗаселениеОбщежитие1**, подобной таблице **ЗаселениеОбщежитие**, но добавить числовые поля **ВремяПроживания**, **КоличествоМест**.

13. Заполнить таблицу **ЗаселениеОбщежитие1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **ЗаселениеОбщежитие1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **ЗаселениеОбщежитие1**. Добавить в таблицу **ЗаселениеОбщежитие1** три записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Общежития**: **КодОбщежития**, **КоличествоМест**.

17. С помощью инструкции SQL вывести информацию таблицы **Студенты**, кроме того добавить поле **НаименованиеФакультета** из связанной таблицы **Факультеты**.

18. С помощью инструкции SQL вывести информацию таблицы **Заселение-Общежитие**, представив общежития адресами, а студентов фамилиями, взяв значения указанных полей из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новом общежитии. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую вносить новые сведения о заселении студентов, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются факультеты, а в подчиненной – список студентов, заселенных в общежитие с расчетом их количества по каждому факультету.

4. Создать форму в виде сводной таблицы, содержащей информацию о заселении студентов в общежития с расчетом общего количества студентов, заселенных в общежитие по каждому факультету, общего количества студентов всех факультетов, заселенных в каждое общежитие, и общего количества всех студентов, заселенных в общежития.

Создание отчетов

1. Создать отчет об общежитиях с расчетом для каждого общежития нового количества мест (увеличение на 10 %), а также общего количества всех мест в общежитиях.

2. Создать отчет с группировкой по общежитиям, отображающий информацию о количестве студентов, заселенных в каждое общежитие, и об общем количестве студентов, проживающих во всех общежитиях.

3. С помощью Мастера отчетов создать отчет с группировкой по факультетам, отображающий информацию о заселении студентов в общежития с получением промежуточного итога: количества студентов каждого факультета, заселенных в общежития, и количества всех студентов, заселенных в общежития.

Создание модулей VBA

1. С помощью программного модуля VBA рассчитать значение поля **ВремяПроживания** таблицы **ЗаселениеОбщежитие1** по формуле:

$$\text{ВремяПроживания} = \text{ДатаВыселения} - \text{ДатаЗаселения}.$$

2. Создать программный модуль VBA, рассчитывающий значение поля **ВремяПроживания** (количество дней) таблицы **ЗаселениеОбщежитие1**, по формуле: **ВремяПроживания = ДатаВыселения - ДатаЗаселения**

и заполняющий поле **КоличествоМест** значениями из таблицы **Общежития**.

3. С помощью программного модуля VBA заполнить таблицу **Факультеты1** данными, рассчитав количество студентов, заселенных в общежития по каждому факультету (поместить в поле **КоличествоСтудОбщ**), на основе информации таблицы **ЗаселениеОбщежитие**.

ВАРИАНТ 13

Расписание вылетов самолетов из аэропорта Минск-1 на сутки

Создать базу данных, позволяющую вести учет вылетов самолетов из аэропортов. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

Расписание

<i>КодСамолета</i>	<i>КодСтюарда</i>	<i>КодГорода</i>	<i>ВремяВылета</i>	<i>ВремяПрибытия</i>
--------------------	-------------------	------------------	--------------------	----------------------

Самолеты

<i>КодСамолета</i>	<i>ТипСамолета</i>	<i>ДатаРемонта</i>	<i>РасходТоплива100Км</i>
--------------------	--------------------	--------------------	---------------------------

Стюарды

<i>КодСтюарда</i>	<i>ФИО</i>
-------------------	------------

Города

<i>КодГорода</i>	<i>НазваниеГорода</i>	<i>РасстояниеКм</i>
------------------	-----------------------	---------------------

Создание таблиц

1. Создать справочники: **Самолеты**, **Стюарды**, **Города**.
2. Описать структуру таблицы **Расписание**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **Расписание**, организовав поля **Код-Самолета**, **КодСтюарда**, **КодГорода** в виде полей со списком.
5. Заполнить таблицу **Расписание** с помощью созданной формы, выбирая значения полей **КодСамолета**, **КодСтюарда**, **КодГорода** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, отображающий список городов, находящихся от Минска на расстоянии более 350 км.

2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **Расписание**, но объекты должны быть представлены своими наименованиями. Выполнить сортировку по полю **ТипСамолета**.

3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запрос2**, кроме того создать расчетный столбец с именем **ВремяВПути**, значения в котором рассчитать по формуле: $\text{FormatDateTime}([\text{ВремяПрибытия}] - [\text{ВремяВылета}])$. Выполнить сортировку по названию города назначения.

4. Составить параметрический запрос, отображающий список рейсов в заданный город.

5. Составить запрос с группировкой, отображающий информацию о суммарном расстоянии, преодоленном каждым самолетом за сутки.

6. Создать перекрестный запрос, подсчитывающий количество рейсов, совершенных каждым стюардом в каждый из городов.

ФИОСтюарда	ЧислоРейсов	Город1	...	ГородN
------------	-------------	--------	-----	--------

Запросы-действия

7. Создать таблицу **РасписаниеВремяВПути**, подобную результирующей таблице **Запрос3** с помощью запроса на создание таблицы.

8. Добавить в таблицу **Стюарды** столбец **Пропуска** и заполнить это поле значением **г.Минск** с помощью запроса на обновление.

9. Создать запрос на добавление еще одного города в таблицу **Города** с кодом, следующим за последним имеющимся кодом.

10. Составить запрос с параметром на удаление информации о заданном типе самолета из таблицы **РасписаниеВремяВПути**.

SQL запросы

11. С помощью инструкции SQL описать структуру таблицы **Расписание1**, подобную таблице **Расписание**, но добавить числовые поля **ВремяВПути** и **ДальностьПолета**.

12. С помощью инструкции SQL описать структуру таблицы **Города1**, подобную таблице **Города**, но добавить числовое поле **КоличествоРейсов**.

13. Заполнить таблицу **Расписание1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **Расписание1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **Расписание1**. Добавить в таблицу **Расписание1** три записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Самолеты**: **ТипСамолета**, **ДатаРемонта**.

17. С помощью инструкции SQL вывести следующую информацию о вылетах самолетов по разным направлениям: **КодГорода**, **НазваниеГорода**, **Время-Вылета**, **ВремяПрибытия**. Данные взять из таблицы **Расписание**, а поле **НазваниеГорода** взять из связанной таблицы **Города**.

18. С помощью инструкции SQL вывести информацию таблицы **Расписание**, представив самолеты своим типами, стюардов – фамилиями, города – наименованиями, взяв значения указанных полей из связанных справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новых пунктах назначения, в которые совершаются рейсы. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую вносить новые сведения о рейсах самолетов, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются самолеты, а в подчиненной – список пунктов назначения, в которые они совершали рейсы, с расчетом количества самолетов, совершивших рейсы в каждый пункт назначения.

4. Создать форму в виде сводной таблицы, содержащей информацию о рейсах самолетов в пункты назначения с расчетом общего количества рейсов в каждый из пунктов назначения, общего количества рейсов каждого типа самолета и общего количества всех рейсов в расписании.

Создание отчетов

1. Создать отчет о самолетах с расчетом для каждого самолета расхода топлива на 1000 км пути по формуле::

$$\text{РасходТоплива1000км} = \text{РасходТоплива100Км} * 10,$$

а также с расчетом общего количества самолетов в справочнике.

2. Создать отчет о рейсах с группировкой по самолетам. Вычислить количество израсходованного топлива в каждом рейсе по формуле: $\text{Расстояние} * \text{РасходТоплива100Км} / 100$. Получить промежуточный итог: количество израсходованного топлива по каждому самолету, а также общий итог – общее количество израсходованного топлива.

3. С помощью Мастера отчетов создать отчет с группировкой по стюардам, отображающий информацию о рейсах самолетов с получением промежуточного итога: количества рейсов, совершенным каждым стюардом, и общего итога – количества всех рейсов в расписании.

Создание модулей VBA

1. С помощью программного модуля VBA рассчитать значение поля **ВремяВПути** таблицы **Расписание1**, используя функцию преобразования типов:

$$\text{ВремяВПути} = \text{FormatDateTime}([\text{ВремяПрибытия}] - [\text{ВремяВылета}]).$$

2. Создать программный модуль VBA, рассчитывающий значение поля **ВремяВПути** таблицы **Расписание1** по формуле:

$$\text{ВремяВПути} = \text{FormatDateTime}([\text{ВремяПрибытия}] - [\text{ВремяВылета}])$$

и заполняющий поле **ДальностьПолета** значениями поля **РасстояниеКм** таблицы **Города**.

3. Создать модуль на языке VBA, подсчитывающий количество рейсов в каждый пункт назначения. Результат вывести в новый столбец **КоличествоРейсов** таблицы **Города1**.

ВАРИАНТ 14

Создать базу данных, позволяющую автоматизировать учет выполненных услуг в парикмахерской. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

ВыполнениеУслуг

<i>КодУслуги</i>	<i>КодКлиента</i>	<i>КодПарикмахера</i>	<i>ДатаИсполнения</i>	<i>ПроцентСкидки</i>
------------------	-------------------	-----------------------	-----------------------	----------------------

Услуги

<i>КодУслуги</i>	<i>НазвУслуги</i>	<i>Стоимость</i>
------------------	-------------------	------------------

Парикмахеры

<i>КодПарикмахера</i>	<i>ФамилияПарикмахера</i>	<i>Разряд</i>
-----------------------	---------------------------	---------------

Клиенты

<i>КодКлиента</i>	<i>ФамилияКлиента</i>	<i>Коэффициент</i>
-------------------	-----------------------	--------------------

Создание таблиц

1. Создать справочники: **Услуги**, **Парикмахеры**, **Клиенты**.
2. Описать структуру таблицы **ВыполнениеУслуг**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **ВыполнениеУслуг**, организовав поля **КодУслуги**, **КодКлиента**, **КодПарикмахера** в виде полей со списком.
5. Заполнить таблицу **ВыполнениеУслуг** с помощью созданной формы, выбирая значения полей **КодУслуги**, **КодКлиента**, **КодПарикмахера** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список услуг, стоимость которых менее 50 000 тыс. руб.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **ВыполнениеУслуг**, но объекты бы представлялись своими наименованиями. Выполнить сортировку по возрастанию даты исполнения.
3. Составить запрос, результирующая таблица которого содержала бы все поля **Запроса2**, кроме того добавить столбец **СтоимостьУслуги** с расчетом по формуле: $Стоимость - стоимость * ПроцентСкидки / 100$. Выполнить сортировку по фамилии парикмахера.

4. Составить параметрический запрос, отображающий список выполненных работ заданным парикмахером.

5. Составить запрос с группировкой, отображающий информацию о стоимости выполненных услуг на каждую дату.

6. Создать перекрестный запрос, отображающий стоимость выполненных работ каждым парикмахером по каждой услуге.

<i>Парикмахер</i>	<i>Общая стоимость</i>	<i>Услуга 1</i>	<i>Услуга 2</i>
-------------------	------------------------	-----------------	-----------------

Запросы-действия

7. Создать таблицу **ВыполненныеУслугиПодробно**, подобную результирующей таблице **Запрос2** с помощью запроса на создание таблицы.

8. Добавить в таблицу **ВыполненныеУслугиПодробно** столбец **Парикмахер-ская** и заполнить это поле значением **Александрина** с помощью запроса на обновление.

9. Создать запрос на добавление строк выполненных услуг парикмахером с кодом 1, подобных имеющимся в таблице, но поле **Дата** увеличить на один день в каждой добавляемой записи.

10. Составить запрос с параметром на удаление записей об услугах заданного парикмахера из таблицы **ВыполненныеУслугиПодробно**.

SQL запросы

11. С помощью инструкций SQL описать структуру таблицы **ВыполнениеУслуг1**, подобную таблице **ВыполнениеУслуг**.

12. С помощью инструкции SQL описать структуру таблицы **Услуги1**, подобную таблице **Услуги**, но добавив поле **ВыполненноеКоличество**.

13. Заполнить таблицу **ВыполнениеУслуг1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **ВыполнениеУслуг1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **ВыполнениеУслуг1**. Добавить в таблицу **ВыполнениеУслуг1** две записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Парикмахеры**: *ФамилияПарикмахера, Разряд*.

17. С помощью инструкции SQL вывести информацию таблицы **ВыполнениеУслуг**, добавив поле *НаименованиеУслуги* из связанной таблицы **Услуги**.

18. С помощью инструкции SQL вывести информацию таблицы **ВыполнениеУслуг**, представив услуги, клиентов и парикмахеров не кодами (как в таблице), а наименованиями, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новых клиентах парикмахерской. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую вносить сведения о новых выполненных услугах, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются парикмахеры, а в подчиненной – список услуг, выполненных каждым парикмахером с подсчетом итоговой стоимости услуг, выполненных каждым парикмахером.

4. Создать форму в виде сводной таблицы, содержащей информацию обо всех выполненных услугах всеми парикмахерами с расчетом общей стоимости всех услуг, выполненных каждым парикмахером, общей стоимости услуг каждого вида, выполненных всеми парикмахерами, и суммарной стоимости всех услуг.

Создание отчетов

1. Создать отчет об имеющихся в справочнике услугах с подсчетом стоимости в тыс. рублей каждой услуги по формуле: *СтоимостьТысРуб=Стоимость/1000* и общей стоимости всех услуг в рублях и в тыс. руб.

2. Создать отчет с группировкой по парикмахерам, отображающий информацию об услугах (по **Запрос3**) с подсчетом итоговой стоимости услуг, выполненных каждым парикмахером, и общей стоимости всех услуг.

3. С помощью *Мастера отчетов* создать отчет с группировкой по видам услуг, отображающий информацию о выполненных услугах с подсчетом стоимости услуг каждого вида, выполненных всеми парикмахерами, и общей стоимости всех услуг.

Создание модулей VBA

1. Создать программный модуль VBA, рассчитывающий для таблицы **ВыполнениеУслуг1** новый столбец *СтоимостьУслугиСоСкид* по формуле: ***Стоимость - Стоимость*ПроцентСкидки/100***.

2. Создать программный модуль VBA, рассчитывающий для таблицы **ВыполнениеУслуг1** новый столбец *СтоимостьУслугиСоСкид* по формуле: ***Стоимость - Стоимость*ПроцентСкидки/100)* Коэффициент***.

3. С помощью программного модуля заполнить таблицу **Услуги1** данными, рассчитав количество услуг каждого вида, выполненных всеми парикмахерами исходя из таблицы **ВыполнениеУслуг**.

ВАРИАНТ 15

Создать базу данных, позволяющую автоматизировать учет выполненных услуг в юридической консультации. Количество записей в справочниках должно быть не менее 3–4, а в таблице, соответствующей агрегированной сущности, – не менее 10.

ВыполнениеУслуг

<i>КодУслуги</i>	<i>КодКлиента</i>	<i>КодКонсультанта</i>	<i>ДатаИсполнения</i>	<i>ПроцентСкидки</i>
------------------	-------------------	------------------------	-----------------------	----------------------

Услуги

<i>КодУслуги</i>	<i>НазвУслуги</i>	<i>Стоимость</i>
------------------	-------------------	------------------

Консультанты

<i>КодКонсультанта</i>	<i>ФамилияКонсультанта</i>	<i>КэффКвалификации</i>
------------------------	----------------------------	-------------------------

Клиенты

<i>КодКлиента</i>	<i>ФамилияКлиента</i>
-------------------	-----------------------

Создание таблиц

1. Создать справочники: **Услуги, Консультанты, Клиенты**.
2. Описать структуру таблицы **ВыполнениеУслуг**.
3. Создать связи между таблицами.
4. Создать форму для заполнения таблицы **ВыполнениеУслуг**, организовав поля **КодУслуги, КодКлиента, КодКонсультанта** в виде полей со списком.
5. Заполнить таблицу **ВыполнениеУслуг** с помощью созданной формы, выбирая значения полей **КодУслуги, КодКлиента, КодКонсультанта** из списка.

Создание запросов

Запросы на выбор

1. Составить запрос **Запрос1**, показывающий список услуг, стоимость которых менее 50 000.
2. Составить запрос **Запрос2**, результирующая таблица которого соответствовала бы по структуре записи таблице **ВыполнениеУслуг**, но объекты бы представлялись своими наименованиями. Выполнить сортировку по возрастанию даты исполнения.
3. Составить запрос **Запрос3**, результирующая таблица которого содержала бы все поля **Запроса2**, кроме того, добавить столбец **СтоимостьУслуги** с расчетом по формуле: **Стоимость - Стоимость*ПроцентСкидки /100**. Выполнить сортировку по фамилии консультанта.
4. Составить параметрический запрос, отображающий список выполненных работ по заданной услуге.
5. Составить запрос с группировкой, отображающий информацию о стоимости выполненных услуг на каждую дату.
6. Создать перекрестный запрос, отображающий стоимость выполненных работ каждым консультантом по каждой услуге.

<i>Консультант</i>	<i>Общая стоимость</i>	<i>Услуга 1</i>	<i>Услуга 2</i>	...
--------------------	------------------------	-----------------	-----------------	-----

Запросы-действия

7. Создать таблицу **ВыполненныеУслугиПодробно**, подобную результирующей таблице **Запрос2** с помощью запроса на создание таблицы.

8. Добавить в таблицу **ВыполненныеУслугиПодробно** столбец **ЮридическаяКонсультация** и заполнить его значением **Юридическая консультация № 5** с помощью запроса на обновление.

9. Создать запрос на добавление строк выполненных услуг заданным консультантом, подобных имеющимся в справочнике, но поле **Дата** увеличить на один день в каждой добавляемой записи.

10. Составить запрос с параметром на удаление записей по заданной услуге из таблицы **ВыполненныеУслугиПодробно**.

SQL запросы

11. С помощью инструкций SQL описать структуру таблицы **ВыполнениеУслуг1**, подобную таблице **ВыполнениеУслуг**.

12. С помощью инструкции SQL описать структуру таблицы **Клиенты1**, подобную таблице **Клиенты**, но добавив поле **КоличествоУслуг**.

13. Заполнить таблицу **ВыполнениеУслуг1** тремя записями с помощью инструкций SQL.

14. Изменить некоторое поле, соответствующее определенному условию отбора в таблице **ВыполнениеУслуг1** с помощью инструкции SQL.

15. Удалить записи, соответствующие определенному условию отбора из таблицы **ВыполнениеУслуг1**. Добавить в таблицу **ВыполнениеУслуг1** две записи по своему усмотрению.

16. С помощью инструкции SQL вывести следующую информацию из таблицы **Консультанты**: **ФамилияКонсультанта**, **КэффКвалификации**.

17. С помощью инструкции SQL вывести информацию таблицы **ВыполнениеУслуг**, добавив поле **НаименованиеУслуги** из связанной таблицы **Услуги**.

18. С помощью инструкции SQL вывести информацию таблицы **ВыполнениеУслуг**, представив услуги, клиентов и консультантов не кодами (как в таблице), а наименованиями, взятыми из соответствующих справочников.

Создание форм

1. Создать форму, позволяющую вносить сведения о новых клиентах юридической консультации. На форме предусмотреть надпись с наименованием формы.

2. На форму, позволяющую вносить сведения о выполнении новых услуг, нанести кнопку, при нажатии на которую открывается форма, созданная в предыдущем задании.

3. Создать форму с подчиненной, где в главной форме отображаются консультанты, а в подчиненной – список услуг, выполненных каждым консультантом с подсчетом итоговой стоимости услуг, выполненных каждым консультантом.

4. Создать форму в виде сводной таблицы, содержащей информацию обо всех выполненных услугах всеми консультантами с расчетом общей стоимости всех услуг, выполненных каждым консультантом, общей стоимости услуг каждого вида, выполненных всеми консультантами, и суммарной стоимости всех услуг.

Создание отчетов

1. Создать отчет об имеющихся в справочнике услугах юридической консультации с подсчетом стоимости в тыс. руб. каждой услуги по формуле: ***СтоимостьТысРуб=Стоимость/1000*** и общей стоимости всех услуг в рублях и в тыс. руб.

2. Создать отчет с группировкой по консультантам, отображающий информацию об услугах (по **Запрос3**) с подсчетом итоговой стоимости услуг, выполненных каждым консультантом, и общей стоимости всех услуг.

3. С помощью *Мастера отчетов* создать отчет с группировкой по клиентам, отображающий информацию о выполненных услугах с подсчетом стоимости услуг, оказанных каждому клиенту, и общей стоимости всех услуг.

Создание модулей VBA

1. Создать программный модуль VBA, рассчитывающий для таблицы **ВыполнениеУслуг1** новый столбец ***СтоимостьУслугиСоСкид*** по формуле: ***Стоимость - Стоимость*ПроцентСкидки /100***.

2. Создать программный модуль VBA, рассчитывающий для таблицы **ВыполнениеУслуг1** новый столбец ***СтоимостьУслугиСоСкид*** по формуле: ***(Стоимость - Стоимость*ПроцентСкидки /100)* КоэффКвалификации***.

3. С помощью программного модуля заполнить таблицу **Клиенты1** данными, рассчитав количество услуг, оказанных каждому клиенту, исходя из таблицы **ВыполнениеУслуг**.

ГЛОССАРИЙ

Атрибут – элемент данных в кортеже.

База данных – набор связанных данных.

База Данных (БД) – структурированный организованный набор данных, описывающих характеристики каких-либо физических или виртуальных систем.

База Данных (БД) – совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

Внешний ключ (foreign key) – поле таблицы, предназначенное для хранения значения первичного ключа другой таблицы с целью организации связи между этими таблицами.

Домен – набор допустимых значений одного или нескольких атрибутов.

Запрос (query) – специальным образом описанное требование, определяющее состав производимых над базой данных операций по выборке или модификации хранимых данных.

Индекс (index) – средство ускорения операции поиска записей в таблице, а также выполнения других операций, использующих поиск: извлечение, модификация, сортировка и т. д.

Индексный файл (indexfile) – файл, в котором хранится информация индекса.

Информационная система (information system) – система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и обслуживающий персонал.

Информационная система типа клиент-сервер (client-server information system) – система, в которой программы СУБД функционально разделены на две части, называемые сервером и клиентом.

Каталог данных (data directory) хранит информацию о месте и способе хранения данных.

Клиент (client) определенного ресурса в компьютерной сети – компьютер (программа), использующий этот ресурс.

Клиентская программа (front-end program) – программа, отвечающая за интерфейс с пользователем, для чего она преобразует его запросы в команды запросов к серверной части (back-end), а при получении результатов выполняет обратное преобразование и отображение информации для пользователя.

Компьютер-клиент (computer-client) – ЭВМ сети, обращающаяся за ресурсами к компьютерам-серверам.

Компьютер-сервер (computer-server) – ЭВМ сети, предоставляющая свои ресурсы другим компьютерам сети.

Концептуальное проектирование – сбор, анализ и редактирование требований к данным.

Логическое проектирование – преобразование требований к данным в структуры данных.

Логическая целостность (logical integrity) – отсутствие логических ошибок в базе данных, к которым относятся нарушение структуры базы данных или ее объектов, удаление или изменение связей между объектами и т. д.

Локальная вычислительная сеть (local area network) – сеть, в которой компьютеры расположены на расстоянии до нескольких километров и обычно соединены при помощи скоростных линий связи.

Макрос (macro) – последовательность макрокоманд встроенного языка программирования СУБД, автоматизирующая выполнение последовательности действий пользователя.

Модель данных – интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.

Модель данных есть формальная теория представления и обработки данных в системе управления базами данных (СУБД), которая включает, по меньшей мере, три аспекта:

- аспект структуры – методы описания типов и логических структур данных;
- аспект манипуляции – методы манипулирования данными;
- аспект целостности – методы описания и поддержки целостности базы данных.

Модель представления данных (data model) – логическая структура данных, хранимых в базе данных.

Нормальная форма – требование, предъявляемое к отношениям в теории реляционных баз данных для устранения из базы избыточности, которая потенциально может привести к логически ошибочным результатам выборки или изменения данных.

Объектно-ориентированная модель данных (object-oriented data model) – модель данных, которая позволяет между записями базы данных и функциями их обработки устанавливать взаимосвязи с помощью механизмов, подобных соответствующим средствам в объектно-ориентированных языках программирования.

Ограничения целостности (integrity constraints) – условия, которым должны удовлетворять хранимые в базе данные.

Отношение (relation) – множество, представляемое двумерной таблицей, состоящей из строк и столбцов данных.

Отношение – N -арным отношением R , или отношением R степени n , называют подмножество декартового произведения множеств D_1, D_2, \dots, D_n , не обязательно различных. Исходные множества D_1, D_2, \dots, D_n называют в модели доменами (в СУБД используется понятие тип данных).

Отношение имеет простую графическую интерпретацию, оно может быть представлено в виде таблицы, столбцы (поля, атрибуты) которой соответствуют вхождением доменов в отношение, а строки (записи) – наборам из n значений, взятых из исходных доменов.

Отчет (report) – объект базы данных, основное назначение которого – описание и вывод на печать документов на основе данных базы.

Отчет (report) – объект базы данных, основное назначение которого – описание и вывод на печать документов на основе данных базы.

Первичный ключ (primary key) – минимальное множество атрибутов, являющееся подмножеством заголовка данного отношения, составное значение которых уникально определяет кортеж отношения.

Поле – некая характеристика моделируемого объекта.

Постреляционная модель данных (post-relational data model) – расширенная реляционная модель, снимающая ограничение неделимости данных, хранящихся в записях таблиц.

Представления – виртуальная (логическая) таблица, результат запроса из базы данных.

Приложение базы данных (database application) – программа или комплекс программ, использующих базу данных и обеспечивающих автоматизацию обработки информации из некоторой предметной области.

Приложение Windows (Windows application) – программа, выполняемая под управлением Windows.

Реляционная алгебра – формальная система манипулирования отношениями в реляционной модели данных.

Реляционная модель данных – логическая модель данных, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в реляционных базах данных.

Реляционная модель данных (relational data model) – модель данных, хранящихся в базе, описывающая взаимосвязи элементов данных в виде отношения (таблицы).

Сервер базы данных (database server) – программа, выполняющая функции управления и защиты базы данных. В случаях, когда вызов функций сервера выполняется на языке SQL, его называют SQL-сервером.

Сервер (server) определенного ресурса в компьютерной сети – компьютер (программа), управляющий этим ресурсом.

Сетевая СУБД (network database managementsystem) – система управления базами данных с произвольной моделью данных, ориентированная на использование в сети.

Сеть (network) – совокупность компьютеров, объединенных средствами передачи данных.

Система управления базами данных (СУБД) – программное обеспечение, управляющее доступом к БД.

Системный каталог в реляционных СУБД представляет собой совокупность специальных таблиц, которыми владеет сама СУБД.

Системный каталог (словарь данных) – совокупное описание данных, называемых метаданными (совокупность метаданных (данные о данных)).

Словарь данных (data dictionary) – подсистема банка данных, предназначенная для централизованного хранения информации о структурах данных, взаимосвязях файлов базы данных друг с другом, типах данных и форматах их представления, принадлежности пользователям, кодах защиты и разграничения доступа и т. п.

СУБД – специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных.

СУБД – программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ.

Сущность (essence) – объект любой природы, данные о котором хранятся в баз данных.

Схема отношения (scheme of relation) – список имен атрибутов отношения.

Схема – структура базы данных.

Таблица – структура данных, хранящая набор однотипных записей.

Технология клиент-сервер (client-server) – технология, при которой процесс обработки информации распределен между клиентом и сервером.

Транзакция (transaction) – последовательность операций над базой данных, отслеживаемая системой управления базами данных от начала до завершения как единое целое.

Транзакция – в информатике, группа последовательных операций, которая представляет собой логическую единицу работы с данными.

Файл – простой набор записей, содержащих логически связанные данные.

Файл (File) – логически связанная совокупность данных (программ, текстов, изображений и др.) определенной длины, имеющая имя.

Файл-сервер (file-server) – компьютер, предназначенный для организации управления файлами в сети.

Файловая система (картотека) – набор программ, которые выполняют для пользователей некоторые операции, каждая программа определяет свои собственные данные и управляет ими.

Физическое проектирование – определение особенностей хранения данных, методов доступа и т. д.

Форма (form) – объект базы данных, в котором разработчик размещает элементы управления, служащие для ввода, отображения и изменения данных в полях.

Форма HTML (HTML form) – основное средство организации интерактивного взаимодействия в Интернете при разработке Web-приложений, служащее для пересылки данных от пользователя к Web-серверу.

Хеширование – преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины.

Целостность базы данных – соответствие имеющейся в базе данных информации ее внутренней логике, структуре и всем явно заданным правилам.

ЛИТЕРАТУРА

1. Базы данных / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев; под редакцией профессора А. Д. Хомоненко. – 6-е изд. – Санкт-Петербург: КОРОНА-Век, 2010. – 736 с.
2. *Балтер, Э.* Microsoft office Access 2007: профессиональное программирование / Элисон Балтер. – Москва. Санкт-Петербург. Киев, 2009. – 1296 с.
3. *Гетц, К.* Access. Сборник рецептов для профессионалов / К. Гетц, П. Литвин, Э. Бэрон. – Москва. Санкт-Петербург. Нижний Новгород. Воронеж. Новосибирск. Ростов-на-Дону. Екатеринбург. Самара. Киев. Харьков. Минск, 2005. – 781 с.
4. *Дейт, К. Дж.* Введение в системы баз данных / К. Дж. Дейт. – 8-е издание. – Москва. Санкт-Петербург. Киев, 2008. – 1314 с.
5. *Змитрович, А. И.* Базы данных и знаний / А. И. Змитрович, В. В. Апансович, В. В. Скакун. – Минск : Издательский Центр БГУ, 2007. – 362 с.
6. *Кренке, Д.* Теория и практика построения баз данных / Д. Кренке. – 9-е изд. – Москва. Санкт-Петербург. Нижний Новгород. Воронеж. Новосибирск. Ростов-на-Дону. Екатеринбург. Самара. Киев. Харьков. Минск, 2005. – 858 с.
7. *Новалис, С.* Access : руководство по макроязыку и VBA / С. Новалис. – Москва : Лори, 2000. – 590 с.
8. Новые информационные технологии / под ред. В. П. Дьяконова. – Москва : СОЛОН-Пресс, 2005. – 639 с.
9. Официальный учебный курс Microsoft: Microsoft Office Access 2003 / Издательство ЭКОНОМ БИНОМ. Лаборатория знаний, 2006. – 526 с.
10. *Шакирин, А. И.* Информационные технологии. Разработка СУБД в Microsoft Office Access : лабораторный практикум / А. И. Шакирин, О. М. Львова. – Минск : БГАТУ, 2010. – 124 с.

ПРИЛОЖЕНИЯ

Репозиторий БГАТУ

Функции, используемые при групповых операциях в запросах

Таблица 1

Функции, используемые при групповых операциях в запросах

Функция	Описание
Avg	Возвращает среднее арифметическое всех значений данного поля в каждой группе
Count	Количество непустых записей запроса
First	Возвращает первое значение данного поля в группе
Last	Возвращает последнее значение данного поля в наборе
Max	Возвращает наибольшее значение, найденное в данном поле внутри каждой группы
Min	Возвращает наименьшее значение, найденное в данном поле внутри каждой группы
Sum	Возвращает сумму всех значений данного поля в каждой группе
StDev	Возвращает стандартное отклонение всех значений данного поля в каждой группе. Эта функция применяется только к числовым или денежным полям. Если в группе меньше двух строк, <i>Microsoft Access</i> возвращает значение Null
Var	Возвращает дисперсию значений данного поля в каждой группе. Эта функция применима только к числовым или денежным полям. Если в группе менее двух строк, <i>Access</i> возвращает значение Null

Функции для работы с данными типа дата/время

Date()

Функция возвращает текущую системную дату в виде *дд.мм.гггг*, где *дд* – день (01–31), *мм* – месяц (01–12), *гггг* – год.

Now()

Функция возвращает текущую дату и время в соответствии с системной датой компьютера.

Time()

Функция возвращает текущее время в соответствии с системным временем компьютера.

DateAdd (*интервал; количество; дата*)

Функция возвращает дату, к которой прибавлен указанный интервал времени. Аргументами функции являются:

интервал – строковое выражение, определяющее интервал времени (день, неделя, месяц);

количество – числовое значение, определяющее количество интервалов времени, которые следует добавить (*количество* – положительное число) или вычесть (*количество* – отрицательное число) из *даты*;

дата – переменная типа Variant (Date) или литерал, представляющий дату, к которой добавляется интервал.

Аргумент *интервал* принимает следующие значения: *yyyy* – год; *q* – квартал; *m* – месяц; *y* – день года; *c* – день; *w* – день недели; *ww* – неделя; *h* – час; *n* – минута; *s* – секунда.

Пример: **DateAdd("yyyy";2;Date())** – к текущей дате прибавить 2 года.

DateDiff(*интервал, дата1, дата2*)

Возвращает значение типа **Variant (Long)**, определяющее количество временных интервалов между двумя указанными датами.

В функцию **DateDiff** входят следующие аргументы (таблица 2).

Аргументы функции **DateDiff**

Аргумент	Описание
<i>интервал</i>	Обязательный аргумент. Строковое выражение, определяющее интервал времени, используемый для расчета разницы между значениями аргументов <i>дата1</i> и <i>дата2</i>
<i>дата1</i> , <i>дата2</i>	Обязательные аргументы типа Variant (Date) . Две даты, используемые в расчете

Значения аргумента *интервал* идентичны значениям предыдущей функции.

Пример: DateDiff("m";#19.02.2009#;#19.10.2009#) – будет получен результат 8 месяцев.

Day(дата)

Функция возвращает целое число в диапазоне от 1 до 31, обозначающее день месяца *даты*.

Пример: Day(#01.12.2009#) – будет получен результат «01».

Month(дата)

Функция возвращает целое число в диапазоне от 1 до 12, обозначающее месяц *даты*.

Пример: Month(#01.12.2009#) – будет получен результат «12».

Year(дата)

Функция возвращает целое число, обозначающее год *даты*.

Пример: Year(#01.12.2009#) – будет получен результат «2009».

Hour(время)

Функция возвращает целое число в диапазоне от 0 до 23, обозначающее час суток.

Minute(время)

Функция возвращает целое число в диапазоне от 0 до 59, обозначающее минуту часа.

Second(время)

Функция возвращает целое число в диапазоне от 0 до 59, обозначающее секунду минуты.

FormatDateTime(дата)

Возвращает выражение в формате даты и времени.

Функции для работы со строковыми данными

Lower (строка) – функция преобразует символы *строки* в нижний регистр.

Upper (строка) – функция преобразует символы *строки* в верхний регистр.

Len (строка) – функция используется для подсчета количества символов в *строке*.

Left (строка; длина) – функция возвращает из *строки* указанное число символов от левого края строки.

Right (строка; длина) – функция возвращает из *строки* указанное число символов от правого края строки.

Mid (строка; начало_поиска[, длина]) – функция возвращает из *строки* указанное число символов. Аргумент *начало_поиска* определяет место в строке, начиная с которого берутся символы.

Replace(выражение, искомая_строка, строка_замены)

Возвращает строку, в которой указанная подстрока заменена на другую подстроку. Функция **Replace** имеет следующие аргументы (таблица 3).

Таблица 3

Аргументы функции **Replace**

Аргумент	Описание
<i>выражение</i>	Обязательный аргумент. Строковое выражение, содержащее подстроку, которую нужно заменить
<i>искомая_строка</i>	Обязательный аргумент. Подстрока, которую требуется найти
<i>строка_замены</i>	Обязательный аргумент. Подстрока, на которую производится замена

Операторы для формирования условий отбора данных

Приведенные в таблице 4 операторы используются для формирования условий отбора записей при создании запросов в КОНСТРУКТОРЕ (в строке *Условие отбора*) или при написании запросов на языке SQL.

Таблица 4

Операторы для фильтрации данных

Оператор	Описание	Примеры
=	Равно	=180 Отберет только те записи, у которых в поле значение равно 180
> >=	Больше, больше или равно	>01.01.2010 Отберет только те записи, у которых в поле <i>Дата</i> находятся значения после 1 января 2010 года
< <=	Меньше, меньше или равно	<=01.02.2010 Отберет только те записи, у которых в поле <i>Дата</i> находятся значения до 1 февраля 2010 года, включая 1 февраля 2010 года
<>	Не равно	<> «Минск» Отберет только те записи, у которых в поле <i>Город</i> находятся значения, отличные от «Минск»
LIKE «шаблон»	Оператор LIKE можно использовать для поиска значений в полях, соответствующих указанному шаблону. Примеры шаблонов приведены в таблице 5.	LIKE "P[A-F]###" Возвращает записи, у которых данные начинаются с буквы «P», после которой идет любая буква между «A» и «F» и три цифры
AND	Записи, удовлетворяющие одному и другому (или нескольким условиям) условию одновременно	>=9.06.2010 AND <=15.06.2010 Отберет только те записи, у которых в поле <i>Дата</i> находятся значения в диапазоне с 9 июня 2010 года по 15 июня 2010 года
BETWEEN... AND	Записи, находящиеся в диапазоне значений	BETWEEN 9.06.2010 AND 15.06.2010 Отберет только те записи, у которых в поле <i>Дата</i> находятся значения в диапазоне с 9 июня 2010 года по 15 июня 2010 года
OR	Записи, удовлетворяющие хотя бы одному из двух или более условий	"75им" OR "76эи" Отберет только те записи, у которых в поле <i>НаименГруппы</i> находятся значения «75им» и «76эи»
NOT	Записи, не удовлетворяющие заданному условию	NOT "72эи" Отберет все записи кроме тех, которые в поле <i>НаименГруппы</i> имеют значение «72эи»

Оператор	Описание	Примеры
&	Слияние нескольких строковых выражений	[Фамилия] & [Имя] & [Отчество] Объединяет поля <i>Фамилия</i> , <i>Имя</i> , <i>Отчество</i> в одно поле
IS NULL	Записи, не имеющие значения в данном поле	IS NULL Отберет те записи, у которых в поле <i>Телефон</i> телефонный номер не был введен
IS NOT NULL	Записи, имеющие значение в данном поле	IS NOT NULL Отберет те записи, у которых в поле <i>Телефон</i> телефонный номер был введен
IS TRUE (IS FALSE)	Записи, имеющие значение истина-да (ложь-нет) в логическом поле	IS TRUE Отберет те записи, у которых в поле <i>ИмеетГрамоту</i> значение «истина»

Таблица 5

Различные типы шаблонов для оператора Like

Тип соответствия	Шаблон	Соответствует шаблону	Не соответствует шаблону
Несколько символов	a*a	aa, aBa, aBBBa	aBC
	ab	abc, AABb, Xab	aZb, bac
Специальные символы	a[*]a	a*a	aaa
Несколько символов	ab*	abcdefg, abc	cab, aab
Один символ	a?a	aaa, a3a, aBa	aBBBa
Одна цифра	a#a	a0a, a1a, a2a	aaa, a10a
Символы в определенном интервале	[a-z]	f, p, j	2, &
Символы вне определенного интервала	[!a-z]	9, &, %	b, a
Не цифра	[!0-9]	A, a, &, ~	0, 1, 9
Комбинация	a[!b-m]#	An9, az0, a99	abc, aj0

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
----------------	---

МОДУЛЬ 4

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ БАЗ ДАННЫХ И БАЗ ЗНАНИЙ. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ В СРЕДЕ СУБД «MS ACCESS». ФОРМИРОВАНИЕ ЗАПРОСОВ К БАЗЕ ДАННЫХ

КРАТКАЯ ХАРАКТЕРИСТИКА СУБД MS ACCESS.....	5
ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ	10

ЛАБОРАТОРНАЯ РАБОТА № 1

СОЗДАНИЕ БАЗЫ ДАННЫХ В ACCESS. ПОСТРОЕНИЕ ТАБЛИЦ И СХЕМЫ ДАННЫХ ЛАБОРАТОРНАЯ РАБОТА № 2 СОЗДАНИЕ ПРОСТЫХ ФОРМ. ЗАПОЛНЕНИЕ ТАБЛИЦ С ИСПОЛЬЗОВАНИЕМ ФОРМ.....	24
--	----

ЛАБОРАТОРНАЯ РАБОТА № 2

СОЗДАНИЕ ПРОСТЫХ ФОРМ. ЗАПОЛНЕНИЕ ТАБЛИЦ С ИСПОЛЬЗОВАНИЕМ ФОРМ	34
---	----

ЛАБОРАТОРНАЯ РАБОТА № 3

РЕДАКТИРОВАНИЕ БАЗЫ ДАННЫХ. ФИЛЬТРАЦИЯ ДАННЫХ.....	44
---	----

ЛАБОРАТОРНАЯ РАБОТА № 4

СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОР ДАННЫХ	51
ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ ПО МОДУЛЮ 4.....	62

МОДУЛЬ 5

ПРОЕКТИРОВАНИЕ ОБЪЕКТОВ БАЗЫ ДАННЫХ – ЗАПРОСОВ-ДЕЙСТВИЙ, ФОРМ, ОТЧЕТОВ, ЯЗЫК SQL

КРАТКИЕ СВЕДЕНИЯ О ЗАПРОСАХ – ДЕЙСТВИЯХ И ФОРМАХ	65
ОБЩИЕ СВЕДЕНИЯ ОБ ОТЧЕТАХ.....	67

ЛАБОРАТОРНАЯ РАБОТА № 5	
СОЗДАНИЕ ЗАПРОСОВ – ДЕЙСТВИЙ.....	70
ЛАБОРАТОРНАЯ РАБОТА № 6	
СОЗДАНИЕ ЗАПРОСОВ НА ЯЗЫКЕ SQL.....	78
ЛАБОРАТОРНАЯ РАБОТА № 7	
КОНСТРУИРОВАНИЕ СЛОЖНЫХ ФОРМ	83
ЛАБОРАТОРНАЯ РАБОТА № 8	
ПРОЕКТИРОВАНИЕ ОТЧЕТОВ	92
ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ ПО МОДУЛЮ 5.....	97
МОДУЛЬ 6	
АВТОМАТИЗАЦИЯ ОБРАБОТКИ БАЗЫ ДАННЫХ В СУБД. СИСТЕМЫ ОБРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ БАЗ ДАННЫХ	
ПРОГРАММИРОВАНИЕ НА VBA (<i>VISUAL BASIC FOR APPLICATION</i>)	100
СОВРЕМЕННЫЕ АРХИТЕКТУРЫ СУБД.....	103
СУБД MICROSOFT SQL SERVER	104
ЛАБОРАТОРНАЯ РАБОТА № 9	
СОЗДАНИЕ ПРОГРАММ НА ЯЗЫКЕ VBA	107
ЛАБОРАТОРНАЯ РАБОТА № 10	
СИСТЕМЫ ОБРАБОТКИ МНОГОПОЛЬЗОВАТЕЛЬСКИХ БАЗ ДАННЫХ	117
ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ ПО МОДУЛЮ 6.....	130
ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ	132
ГЛОССАРИЙ	189
ЛИТЕРАТУРА	194
ПРИЛОЖЕНИЯ	195

Учебное издание

КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Учебно-методический комплекс

Составители:

Ероховец Тамара Викторовна,
Гриневич Елена Георгиевна

Ответственный за выпуск *О. Л. Сапун*
Редактор *Н. А. Антипович*
Компьютерная верстка *А. И. Стебуля*

Подписано в печать 22.02.2011 г. Формат 60×84¹/₈.
Бумага офсетная. Ризография.
Усл. печ. л. 23,71. Уч.-изд. л. 9,27. Тираж 190 экз. Заказ 197.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный аграрный технический университет».
ЛИ № 02330/0552984 от 14.04.2010.
ЛП № 02330/0552743 от 02.02.2010.
Пр. Независимости, 99–2, 220023, Минск.