

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тихоокеанский государственный университет»

**И. А. Ледовских, Н. А. Шулика, Н. П. Табачук**

**Информационные системы  
как средство поиска и хранения данных**

*Утверждено издательско-библиотечным советом университета  
в качестве учебно-методического пособия*

Хабаровск  
Издательство ТОГУ  
2017

УДК 004.9(075.8)

ББК 397я73

Л39

*Рецензенты:*

канд. физ.-мат. наук, доцент, начальник учебно-методического отдела, доцент кафедры математических методов и информационных технологий Дальневосточный институт управления – филиал РАНХ и ГС *М. И. Жидкова;*

канд. пед. наук, доцент кафедры высшей математики Дальневосточного государственного университета путей сообщения *С. В. Коровина*

*Научный редактор:*

канд. физ.-мат. наук, доцент, завкафедрой математики и информационных технологий Тихоокеанского государственного университета *В. А. Казинец*

**Л39 Ледовских, И. А.** Информационные системы как средство поиска и хранения данных : учеб.-метод. пособие / И. А. Ледовских, Н. А. Шулика, Н. П. Табачук. – Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2017. – 140 с.

**ISBN 978-5-7389-2244-2**

Учебно-методическое пособие включает в себя материал, позволяющий получить представление о работе в СУБД MS Access 2010. В пособии даны основные понятия баз данных. Рассмотрены свойства и характеристики интерфейса и инструментов СУБД MS Access 2010.

Пособие предназначено для бакалавров направлений подготовки «Прикладная математика и информатика», «Педагогическое образование» по двум профилям, профиль «Математика», профиль «Информатика» очного и заочного отделений при изучении ими дисциплин «Базы данных», «Информационные системы».

В пособии рассмотрен пример создания базы данных «Университет», приведены задания для выполнения лабораторных работ, представлен материал для самостоятельной работы.

УДК 004.9(075.8)

ББК 397я73

**ISBN 978-5-7389-2244-2**

© Тихоокеанский государственный университет, 2017

© Ледовских И.А., Шулика Н.А., Табачук Н.П., 2017

## ВВЕДЕНИЕ

В настоящее время реляционные базы данных являются неотъемлемой частью большинства информационных систем. В этой области прикладного программирования стремительно развиваются новые технологии, платформы реализации и среды разработки приложений.

Важнейшей структурной частью информационных систем являются базы данных, создаваемые и функционирующие на основе использования специализированных программных систем – систем управления базами данных (СУБД). Все это обуславливает большую потребность в развитии профессиональных компетенций студентов, способных как создавать информационные системы на основе систем управления базами данных, так и обслуживать соответствующие информационные системы и базы данных.

Тематика курсов, связанных с базами данных, широка. Можно указать в качестве примера ряд возможных семестровых или годовых курсов, модулей по соответствующей тематике: «Введение в базы данных», «Проектирование баз данных», «Реляционные базы данных», «Клиент-серверные системы», работа в среде конкретной СУБД и т. п. В связи с этим невозможно в одном курсе детально раскрыть все стороны этой тематики. В то же время важно способствовать формированию у бакалавра достаточно полного представления об общей структуре тематики баз данных и важнейших понятиях в этой области.

Цель данного пособия состоит в формировании концептуальных представлений об основных принципах построения баз данных, систем управления базами данных; о реляционных моделях, описывающих базу данных; об этапах проектирования баз данных; а также анализе основных технологий реализации баз данных. Рассмотрение указанных вопросов иллюстрируется на примере конкретной системы управления базами данных – MS Access.

Пособие разработано с учетом Федеральных государственных образовательных стандартов. Структура пособия соответствует структуре дисциплин «Базы данных», «Информационные системы» для бакалавров направлений подготовки «Прикладная математика и информатика», «Педагогическое образование» по двум профилям «Математика», «Информатика» очного и заочного отделений.

Пособие состоит из четырех взаимосвязанных частей.

В первой части содержится теоретический материал об основных понятиях БД, сопровождаемый примерами.

Во второй части рассмотрены свойства и характеристики интерфейса и инструментов СУБД MS Access и этапы проектирования конкретной базы данных «Университет».

В третьей части представлены методические указания к проведению занятий по дисциплинам «Базы данных», «Информационные системы» в соответствии с пунктами рабочей программы, рейтинговая система оценивания достижений бакалавров. В формировании рейтинговой системы выделяются базовая и дополнительная части с максимальным и минимальным количеством баллов, задания текущей аттестационной работы, продукт текущей аттестации (различные объекты БД).

В четвертой части сформулированы вопросы для самоконтроля и задания для самостоятельного выполнения, тестовые задания; изложен материал творческого характера в форме венгерских кроссвордов, чайнвордов, тематики курсовых проектов по БД. Для формирования спектра заданий использовалось множество информационных ресурсов сети Интернет.

Пособие предназначено для проведения практических занятий, для самостоятельного изучения данных дисциплин и организации самостоятельной работы бакалавров.

Таким образом, в подготовленном учебно-методическом пособии систематизированы накопленные материалы по курсам «Базы данных» и «Информационные системы», и их представление ориентировано на современные тенденции в сфере образования.

# 1 ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ

## 1.1 Базы данных и системы управления базами данных

База данных – важнейшая составная часть информационных систем. Информационные системы предназначены для хранения и обработки больших объемов информации. Изначально такие системы существовали в письменном виде. Для этого использовались различные картотеки, папки, журналы, библиотечные каталоги и т.п. База данных – это реализованная с помощью компьютера информационная структура (модель) реальной системы, отражающая состояния объектов и их отношения. Такую систему называют предметной областью.

В общем смысле термин «база данных» можно применить к любой совокупности связанной информации, объединенной по определенному признаку.

Под базой данных понимают поименованную совокупность данных, отображающих состояние объектов и их отношений в рассматриваемой предметной области.

База данных – это информационная система, которая представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации в памяти компьютера, выполнении специфических, для данного приложения, преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса

База данных – это упорядоченная совокупность данных, предназначенных для хранения, накопления и обработки с помощью ЭВМ, которая предполагает какой либо метод сохранения информации на диске и возможности доступа и манипуляции с ней, а так же набор программных продуктов, предоставляющий пользователю все допустимые средства работы с данными.

База данных – упорядоченная совокупность данных, предназначенная для хранения, накопления и обработки с помощью ЭВМ. Для создания и обработки используется набор языковых и программных средств СУБД

Примерами баз данных могут служить адресная книга, картотека, словарь, энциклопедия или кафедральный журнал. При этом в качестве базы данных рассматривается только набор данных, организованных определенным образом.

База данных предполагает наличие комплекса программных средств, обслуживающих эту базу данных и позволяющих использовать содержащуюся в ней информацию. Такие комплексы программ называются СУБД (*системы управления базой данных*). Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее со-

держимым, редактирования содержимого и визуализации информации. Под *визуализацией информации* базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройства вывода или передача по каналам связи.

СУБД – это совокупность языковых и программных средств, предназначенных для ведения и совместного использования БД многими пользователями.

Существует множество систем управления базами данных.

К основным функциям СУБД относятся:

- определение данных – определить, какая именно информация будет храниться в базе данных, задать свойства данных, их тип, а также указать, как эти данные связаны между собой.

- обработка данных – данные могут обрабатываться разными способами: фильтрация, сортировка, вычисление итогового значения.

- управление данными – указание того, кому разрешено знакомиться с данными, корректировать их или добавлять новую информацию. Определение правил коллективного доступа.

СУБД поддерживают один из возможных типов моделей данных – сетевую, иерархическую или реляционную.

Основными средствами СУБД являются:

- средства задания структуры данных;
- средства конструирования экранных форм, предназначенных для ввода данных, просмотра и обработки в диалоговом режиме;

- средства создания запросов для выборки данных при заданных условиях, а также выполнения операций по их обработке;

- средства создания отчетов из базы данных для вывода на печать результатов обработки в удобном для пользователя виде;

- языковые средства – макросы, встроенный алгоритмический язык, язык запросов и т. п., которые используются для реализации нестандартных алгоритмов обработки данных, а также процедур обработки событий в задачах пользователя;

- средства создания приложений пользователя, позволяющие объединять различные операции работы с базой данных в единый технологический процесс.

СУБД – программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ.

Классификация СУБД по характеру использования:

– персональные СУБД – обеспечивают возможность создания *локальных БД*, работающих на одном компьютере (примеры: Paradox, dBase, FoxPro, MS Access, OpenOffice.org Base и др.);

– многопользовательские СУБД – позволяют создавать ИС, функционирующие в архитектуре «клиент-сервер» (примеры: Oracle, Informix, Microsoft SQL Server, MySQL, PostgreSQL, InterBase, SyBase и др.).

Архитектура СУБД состоит из двух основных компонентов: языка описания данных и языка манипулирования данными.

Язык описания данных (ЯОД) – средства описания данных в БД и связей между ними. Средствами этого языка описывается структура БД, форматы записей, пароли, защищающие данные.

Язык манипулирования данными (ЯМД) – язык для выполнения операций над данными, позволяющий менять их строение. Для разных СУБД реализация этих уровней языков может быть различной.

Обычно с базами данных работают две категории пользователей. Первая категория – проектировщики. Их задача состоит в разработке структуры таблиц базы данных и согласовании ее с заказчиком. Вторая категория исполнителей, работающих с базами данных, – пользователи. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием.

СУБД имеет два режима работы: *проектировочный и пользовательский*. Первый режим предназначен для создания или изменения структуры базы и ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

СУБД Microsoft Access является программой для создания баз данных и поддержания их в актуальном состоянии. Программа проста в использовании. Рассматриваемое в рамках программы курса офисное приложение Microsoft Access не требует знания каких-либо языков программирования, чтобы создавать реляционные базы данных, изучив Microsoft Access, можно без затруднений освоить OpenOffice.org Base и аналогичные ему продукты.

## 1.2 Классификация баз данных

Рассмотрим классификацию баз данных по различным признакам.

*По характеру хранимой информации:*

– фактографические БД – содержат данные, представляемые в краткой форме и строго фиксированных форматах (аналоги бумажных картотек);

– документальные БД – совокупность неструктурированных текстовых документов и графических объектов, снабженная формализован-

ным аппаратом поиска (аналог архива документов - архив судебных дел, исторических документов и пр.).

*По способу хранения данных:*

– централизованные – данные хранятся в памяти одной ЭВМ (автономный компьютер или сервер сети, к которому имеют доступ пользователи-клиенты);

– распределенные – разные части базы данных хранятся в различных ЭВМ вычислительной сети (используются в локальных и глобальных компьютерных сетях).

*По способу доступа к данным* – в зависимости от взаимного расположения приложения и БД: локальный доступ; удаленный доступ.

*По признаку структуры:*

– иерархические БД: данные представляются в виде ориентированного графа (дерева); основные понятия – уровни, узлы (вершины графа), связи.

– сетевые БД: свободная связь между элементами различных уровней (произвольный граф).

– реляционные (табличные) БД: воспринимается пользователем в виде набора таблиц; для формального определения таблицы используется понятие отношения (relation – отношение).

– объектно-ориентированные БД: отдельные записи БД представляются в виде объектов; между записями БД и функциями их обработки устанавливаются взаимосвязи с помощью механизмов, подобных соответствующим средствам в объектноориентированных языках программирования; сочетают особенности сетевой и реляционной моделей; используются для создания крупных БД со сложными структурами данных.

К современным базам данных относят *реляционные системы*. Реляционная модель данных основывается на математических принципах, вытекающих из теории множеств и логики предикатов. Эти принципы впервые были применены в области моделирования данных в конце 60-х г. доктором Коддом, в то время работавшим в ИВМ, а впервые опубликованы в 1970 г.

Реляционные базы данных характеризуются тем, что информация в них хранится в одной или нескольких таблицах. Между таблицами могут существовать связи по ключевым значениям (При дальнейшем изложении под базой данных мы будем понимать реляционную компьютерную базу данных).

Каждая таблица состоит из строк и столбцов, которые в компьютерной базе данных называются *записями и полями* соответственно. Каждая запись содержит информацию об отдельном объекте системы. А каждое поле – это определенная характеристика (свойство, атрибут) объектов. Поля таблицы должны иметь несовпадающие имена. Каждое поле таблицы имеет определенный тип. *Тип* – это множество значений, которые поле может принимать, и множество операций, которые можно выполнять над



этими значениями. Существуют четыре основных типа для полей баз данных: символьный, числовой, логический и дата.

Чтобы автоматизировать процесс поиска нужных записей в таблицах баз данных, необходимо указать поле или несколько полей, значения которых позволяют однозначно выбрать из всей совокупности записей в таблице именно ту запись, которая интересует запрашивающего информацию. Такое поле или группа полей называется *первичным ключом* или просто ключом таблицы. Значение первичного ключа должно быть уникальным. Для каждой таблицы реляционной базы данных должен быть определен первичный ключ. Первичный ключ не допускает значение Null. Если первичный ключ состоит из одного поля, он считается *простым*. В противном случае, если ключ образован из нескольких полей, его называют *составным*.

Важным требованием, предъявляемым к таблицам реляционной модели, является нормализация данных, представленных таблицей. Первично нормализованная таблица содержит строки, в которых для каждого атрибута может быть только одно значение. Это соответствует требованиям недопустимости множественных и повторяющихся структур данных. Ненормализованной таблице обычно соответствует одна или несколько нормализованных таблиц.

Понятие *нормальной формы* было введено Эдгаром Коддом при создании реляционной модели баз данных. Основное назначение нормальных форм – приведение структуры базы данных к виду, обеспечивающему минимальную избыточность. Устранение избыточности производится за счет декомпозиции отношений (таблиц) таким образом, чтобы свести к минимуму функциональные зависимости между их атрибутами (полями).

Приведем два примера первых двух нормальных форм.

*Первая нормальная форма (1NF)*. Таблица находится в первой нормальной форме, если каждый ее атрибут атомарен и все строки различны.

Под выражением «*атрибут атомарен*» понимается, что атрибут может содержать только одно значение. Таким образом, не соответствуют 1NF таблицы, в полях которых могут храниться списки значений.

Для приведения таблицы к 1NF требуется разбить таблицу на несколько отдельных таблиц.

Пример *второй нормальной формы (2NF)*. Таблица находится во второй нормальной форме, если она находится в первой нормальной форме, и при этом любой ее атрибут, не входящий в состав первичного ключа, функционально полно зависит от первичного ключа.

*Функционально полная зависимость* означает, что атрибут функционально зависит от всего первичного ключа, но при этом не находится в функциональной зависимости от какой-либо его части. Обычно выделяют пять нормальных форм.

Классификация баз данных по признаку структуры является важной и требует конкретизации. Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную. Приведем примеры каждой из моделей.

Ядром любой базы данных является модель данных. Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных — совокупность структур данных и операций их обработки.

СУБД основывается на использовании иерархической, сетевой или реляционной модели, на комбинации этих моделей или на некотором их подмножестве.

### *Иерархическая модель данных*

Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево), вид которого представлен на рис.1.

К основным понятиям иерархической структуры относятся: «уровень», «элемент (узел)», «связь». Узел — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис.1, для записи С4 путь проходит через записи А и В3.

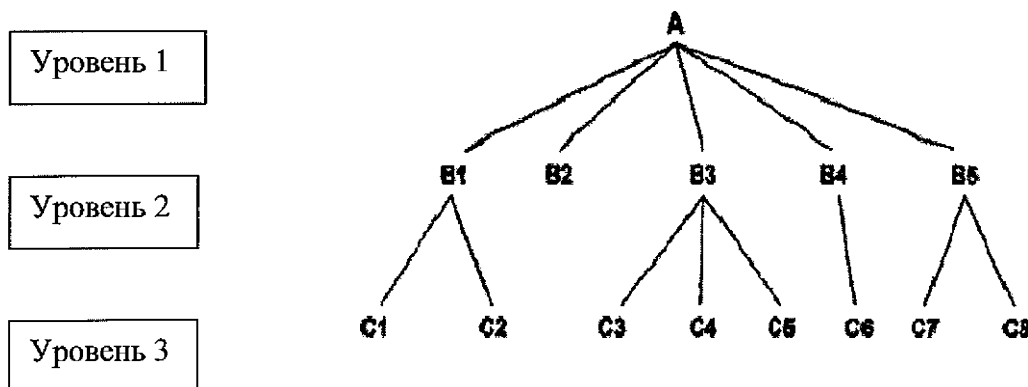


Рис. 1. Графическое изображение иерархической структуры БД

*Пример 1.* Пример, представленный на рис. 2, иллюстрирует использование иерархической модели базы данных. Для рассматриваемого примера иерархическая структура правомерна, так как каждый студент учится в определенной (только одной) группе, которая относится к определенному (только одному) институту.

Институт (специальность, название, директор)



Рис. 2. Пример иерархической структуры данных

### Сетевая модель данных

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

На рис. 3 изображена сетевая структура базы данных в виде графа.

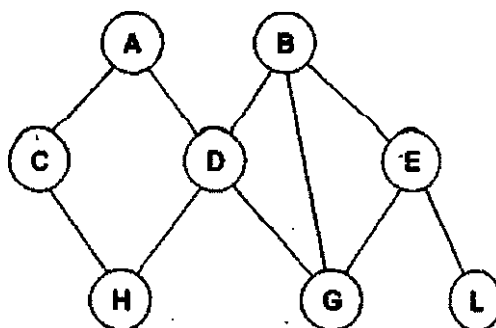


Рис. 3. Сетевая структура базы данных

*Пример 2.* Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о студентах, участвующих в научно-исследовательских работах (НИРС). Возможно участие одного студента в нескольких НИРС, а также участие нескольких студентов в разработке одной НИРС. Графическое изображение описанной в примере сетевой структуры, состоящей только из двух типов записей, показано на рис. 4. Единственное отношение представляет собой сложную связь между записями в обоих направлениях.

Студент (номер зачетной книжки, фамилия, группа)



Рис. 4. Пример сетевой структуры БД

### *Реляционная модель данных*

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы – один элемент данных;
- все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

*Пример 3.* Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе (табл. 1).

Пример реляционной таблицы

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.98	ПОМИ(б)-21
16593	Петрова	Анна	Владимировна	15.03.98	ПОМИ(б)-21
16693	Анохин	Андрей	Борисович	14.04.99	ПОМИ(б)-21

### Межтабличные связи

Необходимо создание межтабличных связей с целью обеспечения защиты целостности данных и автоматизации задач обслуживания БД. Связи между таблицами позволяют:

1) исключить возможность удаления или изменения данных в ключевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц;

2) при удалении данных в ключевом поле главной таблицы автоматически произойдет удаление соответствующих данных в полях связанных таблиц.

#### Типы связи между таблицами

Все информационные объекты предметной области связаны между собой. Различаются связи нескольких типов, для которых введены следующие обозначения:

- один к одному (1 : 1);
- один ко многим (1 : M);
- многие ко многим (M : M).

Большинство современных СУБД основано на *реляционной модели* представления данных. Реляционная ориентирована на организацию данных в виде *двумерных таблиц*. Каждая *реляционная таблица* представляет собой *двумерный массив* и обладает рядом свойств.

#### Структура таблицы

Понятие структуры таблицы включает в себя:

- описание полей;
- ключ — поле (совокупность полей), которое однозначно определяет соответствующую запись (простой, составной);
- индексы — (индексирование — задание порядка записей без изменения физического расположения записей);
- ограничения: на значения полей, ссылочной целостности между таблицами (рис. 5).

*Использование ключей и индексов позволяет:*

- однозначно идентифицировать записи (ключ);
- избегать дублирования значений в ключевых полях (ключ);

- выполнять сортировку таблиц (индекс);
- ускорять операции поиска в таблицах (ключ, индекс);
- устанавливать связи между отдельными таблицами БД (ключ, индекс);
- использовать ограничения ссылочной целостности (ключ, индекс).

*Термины описания структуры данных*

- Отношение – таблица.
- Кортеж – строка или запись.
- Кардинальность – количество записей.
- Атрибут – столбец или поле.
- Степень – количество атрибутов.
- Домен – тип данных, тип атрибутов, совокупность допустимых значений.
- Первичный ключ – уникальный идентификатор.



Рис. 5. Обобщающая таблица

*Требования к организации данных*

Успешное функционирование БД может быть осуществлено только при выполнении ряда требований к ее организации.

- **Неизбыточность данных:**
  - это требование тесно связано с «интегрированностью» базы данных;

– предполагается, что в идеале любые данные будут храниться в БД в одном экземпляре;

– дублирование данных не только увеличивает объем требуемой памяти, но и, самое главное, может легко привести к противоречивости данных, и, следовательно, неверной работе СУБД.

□ *Совместное использование данных:*

– одни и те же данные базы могут использоваться несколькими пользователями (задачами);

– желательно, чтобы каждый пользователь (задача) получал эти данные в удобном для него виде;

– в случае одновременного использования одних и тех же данных разными пользователями (задачами) должен быть обеспечен мультидоступ к данным базы.

□ *Расширяемость базы данных:*

– БД должна обладать способностью к расширению, которое может быть произведено за счет: увеличения числа экземпляров однотипных данных; введение в БД новых типов взаимосвязей;

– необходимо, чтобы введение новых типов объектов или связей в БД не требовало каких-либо изменений в уже существующих данных.

□ *Простота работы с базой данных:*

– структура данных должна быть логичной и ясной как для программистов, так и для пользователей;

– операции доступа к данным должны обладать ясными и четко очерченными функциями;

– различные обслуживающие операции (копирование, перепись с носителя на носитель, расширение БД и др.) должны выполняться без больших трудозатрат.

□ *Эффективность доступа к базе данных.* Под эффективностью здесь понимается скорость доступа к данным при ограничениях на объем занимаемой при этом оперативной памяти.

□ *Целостность базы данных.* Под целостностью базы данных в общем случае понимается ее готовность к работе.

– Основные аспекты целостности: *физическая целостность*, т.е. сохранность информации на магнитных носителях; *логическая целостность*, под которой понимают непротиворечивость данных в базе; *актуальность данных*, т.е. соответствие данных реальному положению вещей. Причины потери целостности: сбои в работе аппаратуры ЭВМ; ошибки в программном обеспечении; неверная технология ввода и корректировки данных; низкая достоверность самих данных и т.д.

- **Независимость данных от программы:**
  - принцип независимости данных требует, чтобы при различного рода модификациях, расширениях и реорганизациях БД не требовалось вносить изменения в уже существующие прикладные программы;
  - возможность использования базы данных, не зная внутреннего представления данных.
- **Секретность данных:**
  - защита данных от несанкционированного доступа;
  - различаются: собственно секретность данных, заключающаяся в том, что пользователю, не владеющему паролем доступа, полностью закрыт доступ к БД или какой-либо ее части; защита данных от модификации, допускающая для лиц, не владеющих паролем, только чтение из БД.

### 1.3 Объекты базы данных

Базы данных могут содержать различные объекты. Основными объектами любой базы данных являются ее таблицы. Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в простейшей базе данных являются поля и записи.

Если в базе нет никаких данных (пустая база), то это все равно база данных. Этот факт имеет методическое значение. Данных в базе и нет, но информация в ней есть — это структура базы. Она определяет методы занесения данных и хранения их в базе.

Рассмотрим объекты базы данных (на примере СУБД MS Access).

*Таблицы* – это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства). Таблица предназначена для хранения данных в виде записей (строк) и полей (столбцов). Обычно каждая таблица используется для хранения сведений по одному конкретному вопросу.

*Запросы.* Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции, как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразования данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и

*Формы* – это средства для ввода данных. Смысл их предоставить пользователю средства для заполнения только тех полей, которые ему возможно заполнять. В форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочее) для автоматизации ввода.



*Отчеты.* По своим свойствам и структуре отчеты похожи на формы, но предназначены только для вывода данных на принтер. В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов.

*Страницы.* Этот объект позволяет создать Web-страницы, компоненты которых осуществляют связь с базой данных.

*Макросы и модули.* Эти объекты предназначены как для автоматизации повторяющихся операций при работе с СУБД, так и для создания новых функций путем программирования. Макросы состоят из последовательности внутренних команд СУБД Access и являются одним из средств автоматизации работы с базой. Модули создаются средствами внешнего языка программирования Visual Basic for Applications.

Таким образом, основными понятиями баз данных являются: база данных; система управления базами данных; модели данных, такие как иерархическая, сетевая и реляционная. Модель представления данных служит основой для создания различных объектов баз данных. Для организации, обработки и хранения данных используются СУБД.

#### 1.4 Проектирование реляционной базы данных

Существуют разные подходы к проектированию баз данных. Мы рассмотрим проектирование базы данных методом «сущность-связь».

Проектирование включает в себя:

- системный анализ предметной области;
- анализ данных и построение модели данных.

Первый этап проектирования – *инфологическое моделирование*. Чтобы спроектировать структуру БД, необходима исходная информация о предметной области. Желательно, чтобы эта информация была представлена в формализованном виде. Описание предметной области, выполненное без ориентации на используемые в дальнейшем программные и технические средства, называется инфологической моделью предметной области.

*ER-модель* (E (*entity*) – сущность, R (*relationship*) – связь) является центральной компонентой инфологической модели предметной области.

На втором этапе проектирования на основе инфологической модели строится *даталогическая модель* БД. Даталогическая модель представляет собой отражение логических связей между элементами данных безотносительно к их содержанию и среде хранения. Этап создания ДЛМ называется *даталогическим проектированием*. Описание логической структуры БД на языке СУБД называется схемой.

Третий этап проектирования состоит в подвязке ДЛМ к среде хранения с помощью физической модели данных. Описание физической струк-

туры БД называется схемой хранения, соответствующий этап проектирования БД – *физическим проектированием*.

Дальнейшее изучение информационных систем и баз данных будем проводить на примере разработки ИС для учета коллекции имеющихся у Вас видео, CD или DVD-фильмов.

*Пример 1. Спроектировать БД «Домашняя видеотека». В данной базе данных должна содержаться информация о фильмах, имеющихся в вашей коллекции, о лицах, которые пользуются Вашей видеотекой, а также должна фиксироваться информация о дате выдачи и дате возврата фильма.*

При проектировании базы данных необходимо выполнить следующие действия:

1. Уточнить, какая именно информация о предметной области будет храниться в проектируемой базе данных. Выделить в предметной области объекты и их свойства. Зафиксировать связи между объектами и их свойствами и связи между объектами разных классов. Построить ER-модель.

2. Осуществить переход от инфологической модели предметной области к даталогической модели базы данных.

3. Выяснить, в какой нормальной форме находятся полученные отношения (отобразить функциональные зависимости между атрибутами каждого отношения).

4. На основе полученного набора отношений, используя конкретную СУБД, создать базу данных.

Выделим в предметной области простые объекты и их свойства. Зафиксируем связи между объектами. Осуществим переход от инфологической модели предметной области к даталогической модели базы данных. Даталогическая модель базы данных может иметь следующий вид:

*Пользователи* (Код пользователя, ФИО пользователя, Адрес, Телефон)

*Жанр* (Код жанра, Название жанра)

*Страны* (Код страны, Название страны)

*Тип носителя* (Код носителя, Название носителя, Формат)

*Фильмы* (Код фильма, Название фильма, Режиссер, Код страны, Код жанра, Продолжительность, Код носителя)

*Выдача возврат* (Код фильма, Дата выдачи, Код пользователя, Дата возврата).

Таблица 2

Атрибуты, их тип и размер	
Название поля	Тип данных
Код пользователя	Числовой
ФИО пользователя	Текстовый
Адрес	Текстовый
Телефон	Текстовый
Код жанра	Числовой
Название жанра	Текстовый

Окончание табл. 2

Название поля	Тип данных
Код страны	Числовой
Название страны	Тестовый
Код носителя	Числовой
Название носителя	Текстовый
Формат	Текстовый
Код фильма	Числовой
Название фильма	Текстовый
Режиссер	Текстовый
Продолжительность	Дата/время
Дата выдачи	Дата/время
Дата возврата	Дата/время

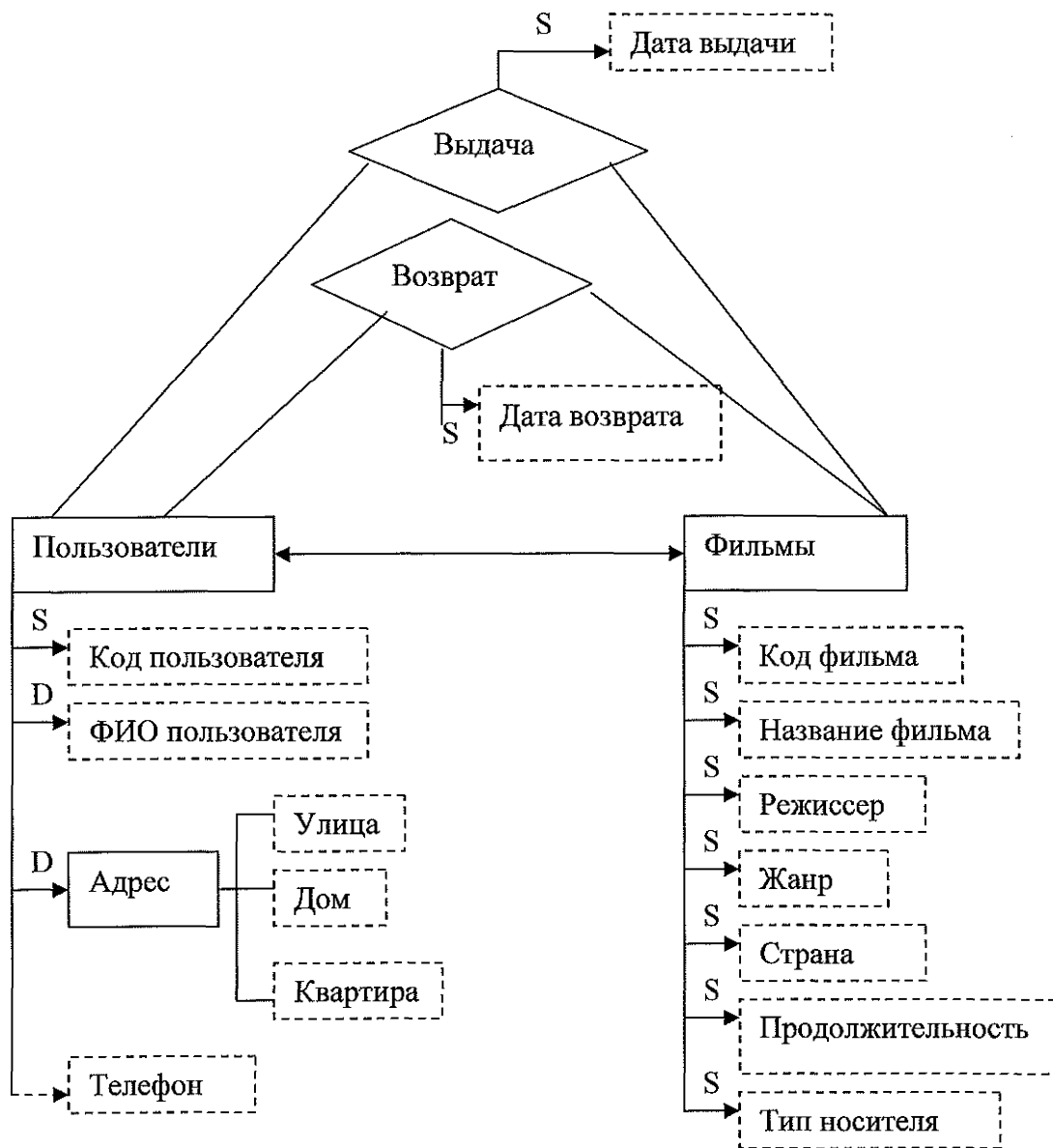


Рис. 6. ER-модель рассматриваемой предметной области

Проектирование БД «Домашняя видеотека» методом «сущность-связь» завершено, теперь можно приступить к физической реализации базы данных в среде конкретной СУБД.

База данных для домашней видеотеки будет содержать шесть таблиц.

– В первой таблице (*Пользователи*) будут регистрироваться данные о лицах, которые берут у Вас фильмы для просмотра: код пользователя, его фамилия, имя, отчество, адрес, телефон.

– Во второй таблице (*Жанр*) будут содержаться сведения о жанре фильма, а именно: код жанра и название жанра (например, мелодрама, фантастика, комедия, боевик, блокбастер и т.п.).

– В третьей таблице (*Страна*) будут храниться данные о названии страны, где был снят фильм: код страны, название страны.

– В четвертой таблице (*Тип носителя*) будут храниться данные о том, на каком носителе находится фильм: код носителя, название носителя, формат (например: оптический диск формата DVD, оптический диск формата CD или видеокассета формата VHS).

– В пятой таблице (*Фильмы*) будут храниться данные о фильмах, имеющихся в домашней видеотеке: код фильма, название фильма, код страны, где был снят фильм, код жанра фильма, продолжительность, код носителя).

– И в шестой таблице (*Выдача-возврат*) будет вестись учет выдаваемых фильмов: код выданного фильма, код пользователя, кому выдан фильм, дата выдачи фильма, дата возврата фильма.

Таким образом, при проектировании баз данных выделяют три этапа: создание инфологической модели предметной области, даталогическое и физическое проектирование. В представленном примере изображена инфологическая модель и описаны свойства даталогической модели предметной области такой, как «Домашняя видеотека».

## 2 ОСНОВЫ РАБОТЫ С MS ACCESS

### 2.1 Этапы проектирования базы данных в MS Access

Процесс проектирования базы данных с помощью СУБД можно представить в виде следующих основных этапов.

1. Определение цели создания базы данных. На первом этапе проектирования базы данных необходимо определить словесную и документальную характеристику предметной области, назначение базы данных, какие сведения будут в ней храниться. Разрабатываемая модель предметной области представляется в виде графической схемы.

2. Выделение таблиц базы данных.

3. Определение в таблице необходимых полей и их типов.

4. Выявление полей с уникальными значениями в каждой записи (первичные ключи).

5. Определение связей между таблицами.

6. Усовершенствование структуры базы данных до заполнения таблиц данными.

7. Ввод данных и создание других объектов базы данных (запросы, формы, отчеты).

#### *Состав специалистов, обслуживающих БД*

⇒ Администратор имеет представление об информационных потребностях пользователей. *Функции администратора:* защита БД; поддержание целостности; обучение пользователей; загрузка данных из других БД; резервное копирование и восстановление базы данных; внесение изменений в БД.

⇒ Аналитик в контакте с пользователем строит математические модели для задач конечного пользователя.

⇒ Прикладные программисты на основе математической модели задачи разрабатывают прикладные программы.

⇒ Системные программисты обеспечивают работоспособность ОС, СУБД; разрабатывают сервисные программы.

### 2.2 Основные принципы работы MS Access

Программа MS Access является системой по хранению, редактированию и обработке данных. База данных MS Access позволяет добавлять данные из внешних ресурсов.

Для освоения навыков работы в программе определим структуру построения любой базы. Данные в базе хранятся в виде таблиц, основой которых является поле.

*Поле* – это элемент таблицы, который содержит данные определенного рода, например фамилию студента. Имя поля является заголовком столбца таблицы.

*Запись* – полный набор данных об определенном объекте. В режиме таблицы запись изображается как строка.

База данных может состоять из нескольких таблиц, связанных между собой по различным полям.

СУБД MS Access входит в состав пакета Microsoft Office. Внешний интерфейс MS Access соответствует общим принципам пакета Microsoft Office.

Главный элемент пользовательского интерфейса MS Access 2010 представляет собой *Ленту*, которая идет вдоль верхней части окна каждого приложения (рис. 7).

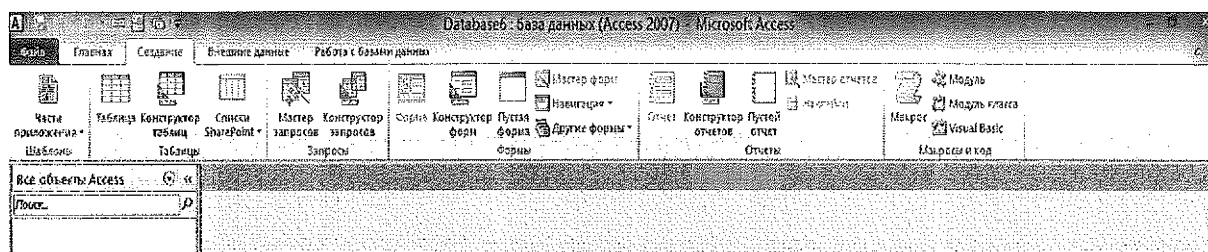


Рис. 7. Лента с вкладками в MS Access 2010

Лента управления содержит вкладки. По умолчанию их пять: *Файл*, *Главная*, *Создание*, *Внешние данные*, *Работа с базами данных*. Каждая вкладка связана с видом выполняемого действия.

*Панель быстрого доступа*, находящаяся по умолчанию над лентой, обеспечивает доступ к командам одним щелчком мыши. Набор по умолчанию включает команды *Сохранение*, *Отмена* и *Возврат*, при этом можно настроить панель быстрого доступа для добавления в нее наиболее часто используемых команд.

*Область навигации*, расположенная по левому краю окна Access. Она предназначена для отображения объектов или групп объектов открытой базы данных, а также для перехода от объекта к объекту. Управлять объектами можно командами ленты и командами контекстного

## 2.3 Создание базы данных в MS Access

Процесс создания базы данных начинается с запуска MS Office, MS Access и выбора объекта «*Новая база данных*» (рис. 8).

Процесс создания базы данных рассмотрим на примере описания структуры университета. Создание базы данных, раскрывающей структуру университета, позволяет определить не только особенности его организа-

ции, но и раскрыть индивидуальное своеобразие факультетов, групп, направлений подготовки.

### Доступные шаблоны



Рис. 8. Первоначальный этап создания базы данных

База данных «Университет» состоит из 4 таблиц (**Факультет**, **Направление подготовки**, **Группа**, **Студенты**), содержащих информацию о факультетах Педагогического института Тихоокеанского государственного университета, направлениях подготовки, группах, студентах вуза. Для создания базы данных необходимы фотографии студентов в формате .jpg или .gif. Создание новых таблиц происходит в режиме конструктора или с помощью мастера. В режиме конструктора необходимо представить структуру новой таблицы, в режиме мастера можно выбирать поля из шаблонного списка. Для создания перечисленных выше 4 таблиц будем работать в режиме конструктора. Таблицы будут содержать поля и их типы, представленные в табл. 3.

Таблица 3

### Таблицы базы данных «Университет» с полями и их типами

Таблица в режиме конструктора <b>Факультет</b>		Таблица в режиме конструктора <b>Направление подготовки</b>																							
<table border="1"> <thead> <tr> <th colspan="2">Факультет</th> </tr> <tr> <th>Имя поля</th> <th>Тип данных</th> </tr> </thead> <tbody> <tr> <td>Номер факультета</td> <td>Числовой</td> </tr> <tr> <td>Наименование факультета</td> <td>Текстовый</td> </tr> <tr> <td>Декан</td> <td>Текстовый</td> </tr> </tbody> </table>		Факультет		Имя поля	Тип данных	Номер факультета	Числовой	Наименование факультета	Текстовый	Декан	Текстовый	<table border="1"> <thead> <tr> <th colspan="2">Направление подготовки</th> </tr> <tr> <th>Имя поля</th> <th>Тип данных</th> </tr> </thead> <tbody> <tr> <td>Номер направления подготовки</td> <td>Числовой</td> </tr> <tr> <td>Наименование направления</td> <td>Текстовый</td> </tr> <tr> <td>Стоимость Обучения</td> <td>Денежный</td> </tr> <tr> <td>Профиль</td> <td>Текстовый</td> </tr> </tbody> </table>		Направление подготовки		Имя поля	Тип данных	Номер направления подготовки	Числовой	Наименование направления	Текстовый	Стоимость Обучения	Денежный	Профиль	Текстовый
Факультет																									
Имя поля	Тип данных																								
Номер факультета	Числовой																								
Наименование факультета	Текстовый																								
Декан	Текстовый																								
Направление подготовки																									
Имя поля	Тип данных																								
Номер направления подготовки	Числовой																								
Наименование направления	Текстовый																								
Стоимость Обучения	Денежный																								
Профиль	Текстовый																								

Таблица в режиме конструктора  
Группа

Группа	
Имя поля	Тип данных
Номер группы	Числовой
Номер направления подготовки	Числовой
Номер факультета	Числовой
Номер курса	Числовой

Таблица в режиме конструктора  
Студенты

Студенты	
Имя поля	Тип данных
Номер зачетной книжки	Числовой
Фамилия, имя, отчество	Текстовый
Коммерческий	Логический
Дата рождения	Дата/время
Номер группы	Числовой

## 2.4 Создание новой таблицы

**Цель работы:** ознакомление с основными понятиями СУБД Access; освоение технологии конструирования реляционных таблиц.

### Краткие сведения

СУБД Access использует реляционную модель базы данных, в которой данные представлены в виде взаимосвязанных таблиц (отношений по англ. – relations).

Важнейшим этапом проектирования базы данных является разработка информационно-логической (*инфологической*) модели предметной области, не ориентированной на СУБД, но отражающей предметную область в виде совокупности информационных объектов и их информационных связей.

СУБД Access позволяет работать с объектами базы данных, к которым относятся *таблицы, запросы, формы, отчеты, страницы, макросы и модули*.

**Таблицы** служат для хранения данных в определенной структуре.

**Запросы** создаются для выборки данных из одной или нескольких связанных таблиц.

**Формы** предназначены для ввода, редактирования и просмотра табличных данных на экране в удобном виде.

**Отчеты** являются выходными документами, предназначенными для вывода на принтер.

**Страницы** доступа к данным – это Web- страницы, обеспечивающие функциональность стандартных форм и отчетов Access: ввод, редактирование и представление данных. Страницы доступа к данным можно открывать в программах просмотра Web- страниц (например, Internet Explorer) и использовать для ввода, просмотра и отбора информации. **Макросы** создаются для автоматизации различных процедур обработки данных, являются программами, состоящими из макрокоманд высокого уровня. Макропрограммирование в Access не требует знания языка



Visual Basic. Имеющийся в Access набор из около 60 макрокоманд обеспечивает практически любые действия, необходимые для решения задач.

**Модули** являются программами на языке, которые служат для реализации нестандартных процедур обработки данных.

Все данные БД Microsoft Access и средства их отображения хранятся в одном файле с расширением **MDB**.

### *Таблица*

Таблицы составляют основу базы данных – именно в них хранятся все данные. Таблицы должны быть тщательно спланированы. Прежде всего, должна быть спланирована структура каждой таблицы. Структура таблиц определяется содержанием тех выходных форм и отчетов, которые должны быть затем получены. При планировании таблиц необходимо избежать дублирования информации в разных таблицах.

**Таблица** – это объект БД, который хранит данные определенной структуры. Таблица состоит из записей (строк), каждая из которых описывает одну сущность. Каждый столбец таблицы – это **поле**. Столбец содержит однотипную информацию.

Длина имени таблицы – не более 64 символов.

Длина имени поля – не более 64 символов.

Количество полей в одной таблице – не более 255.

Количество записей – неограниченно.

Суммарный объем информации во всей БД – не более 1 гигабайта.

Для каждого поля необходимо указать тип данных. Тип данных определяет вид и диапазон допустимых значений, которые могут быть введены в поле, а также объем памяти, выделяющийся для этого поля.

Таблица может содержать следующие типы полей (всего 8):

**Текстовый.** Короткий текст. Текст и числа, например, имена и адреса, номера телефонов и почтовые индексы. Текстовое поле может содержать до 255 символов.

**Поле Мемо.** Длинный текст и числа, например, комментарии и пояснения. Мемо-поле может содержать до 65 535 символов.

**Числовой.** Общий тип для числовых данных, допускающих проведение математических расчетов, за исключением расчетов для денежных значений. Свойство Размер поля позволяет указать различные типы числовых данных. Длина – до 8 байт. Точность – до 15 знаков.

**Дата/время.** Денежные значения. Числа представляются с двумя знаками после запятой. Не рекомендуется использовать для проведения денежных расчетов значения, принадлежащие к числовому типу данных, так как последние могут округляться при расчетах. Значения типа «Денежный» всегда выводятся с указанным числом десятичных знаков после запятой. Длина – 8 байт.

**Счетчик.** Автоматически вставляющиеся последовательные номера. Счетчик увеличивается на единицу для каждой следующей записи. Нумерация начинается с 1. Поле счетчика удобно для создания ключа. В таблице может быть только одно такое поле. Длина – 4 байта.

**Логический.** Значения «Да» / «Нет», «Истина» / «Ложь», «Вкл» / «Выкл», т.е. одно из двух возможных значений. Длина – 1 байт.

**Поле объекта OLE.** Объекты, созданные в других программах, поддерживающих протокол OLE, например графики, рисунки и т.п. Объекты связываются или внедряются в базу данных Microsoft Access через элемент управления в форме или отчете.

**Гиперссылка** - позволяет вставлять в поле гиперссылку, с помощью которой можно сослаться на произвольный фрагмент данных внутри поля или страницы на том же компьютере, в локальной сети или в Internet.

**Индексирование полей таблицы.** Индексирование позволяет ускорить сортировку и поиск данных в таблице. Можно индексировать числовые, денежные, текстовые, логические поля, а также поля типа Счетчик и Дата. Не следует создавать слишком много индексов для одной таблицы, т.к. это замедлит ввод и редактирование ее данных.

**Первичный ключ** – это специальный тип индекса, который однозначно идентифицирует каждую запись. В первичный ключ могут входить несколько полей, но значение первичного ключа должно быть уникальным для каждой записи. Первичные ключи используются для установления связей между таблицами.

**Связи между таблицами.** Таблицы могут быть связаны отношениями *один-к-одному*, *один-ко-многим* и *многие-ко-многим*. Access позволяет использовать только отношения первых двух типов.

При установлении связей нужно определить какая таблица является *главной*, а какая – *подчиненной*.

Отношение *один-к-одному* означает, что одной записи подчиненной таблицы соответствует только одна запись в главной таблице. Такие отношения встречаются очень редко, т. к. требуют неоправданно много места в БД. Вместо них можно просто добавить поля подчиненной таблицы к полям главной.

Наиболее часто используются отношения *один-ко-многим*. В этом случае одной записи в главной таблице соответствует несколько записей в подчиненной таблице.

Для создания отношений необходимо указать поля в двух таблицах, которые содержат одни и те же данные. Обычно такое поле в одной из таблиц (главной) является ключевым. Имена связываемых полей могут отличаться, но типы и свойства должны совпадать. Возможна связь между полем типа Счетчик и полем типа Число с форматом Длинное целое.

**Задание 1.** Создать базу данных **Университет** на основе инфологической модели, приведенной на рис. 9. База данных должна содержать 4 взаимосвязанных таблицы: *Факультет*, *Направление подготовки*, *Группа*, *Студенты*.

**Таблица Факультет:**

- Номер факультета – ключевое поле числового типа, байтовое;
- Наименование факультета – текстовое поле;
- Декан – текстовое поле.

**Таблица Направление подготовки:**

- Номер направления – ключевое поле числового типа, длинное целое;
- Наименование направления подготовки – текстовое поле;
- Стоимость обучения – денежного типа;
- Название профиля обучения – текстовое поле.

**Таблица Группа:**

- Номер группы – ключевое поле числового типа, целое;
- Номер направления – числовое поле, длинное целое;
- Номер факультета – числовое поле, байтовое;
- Номер курса – числовое поле, байтовое.

**Таблица Студенты:**

- Номер зачетной книжки – ключевое поле числового типа, длинное целое;
- Номер группы – числовое поле, целое; мастер подстановок
- Фамилия, имя, отчество – текстовое поле;
- Дата рождения – поле типа «дата»;
- Коммерческий – логическое поле (да/нет).

**Технология** создания таблицы *Факультет*:

1. Создать новую базу данных.
2. Ввести имя базы данных «Университет.mdb».
3. В окне базы данных выбрать таблицы, *режим конструктора*.
4. Ввести названия полей таблицы *Факультет* и типы данных для каждого из них.
5. Отметить ключевое поле *Номер факультета*.

Факультет		
	Имя поля	Тип данных
✓	Номер факультета	Числовой
	Наименование факультета	Текстовый
	Декан	Текстовый

Рис. 10. Окно таблицы *Факультет* в режиме *Конструктора*

Аналогично создаются таблицы: *Направление подготовки, Группа, Студенты*.

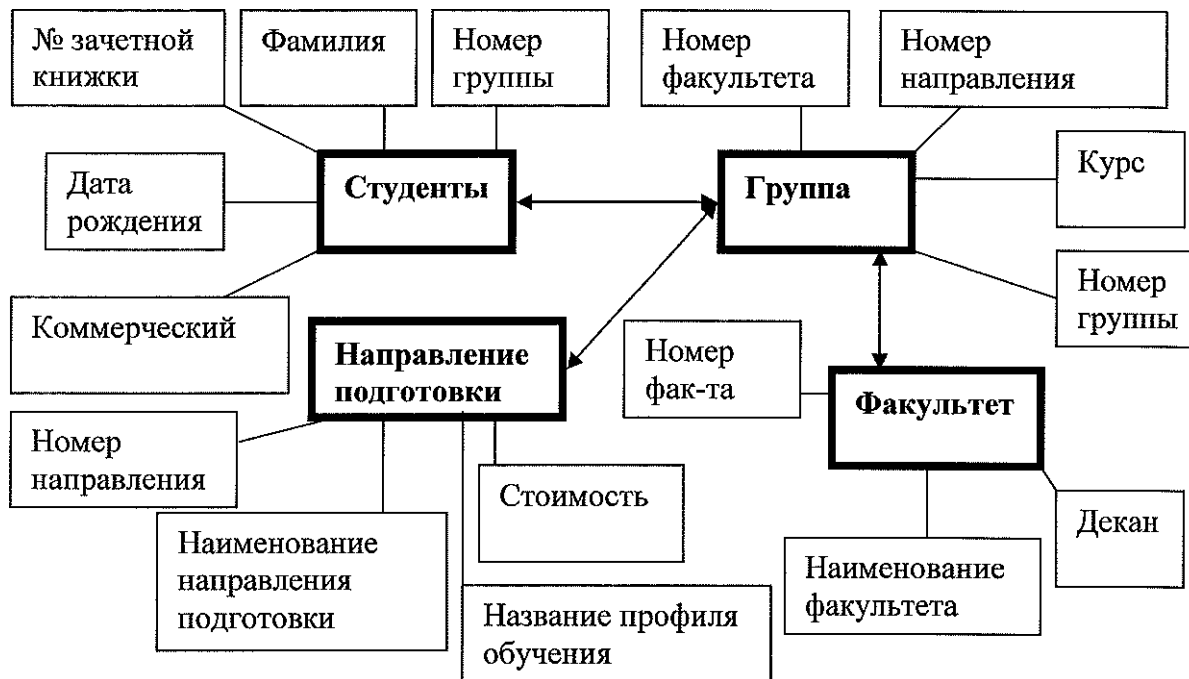


Рис. 9. Инфологическая модель базы данных *Университет*

## 2.5 Связи между таблицами

Таблицы могут быть связаны отношениями один-к-одному, один-ко-многим и многие-к-многим.

При установлении связи нужно определить, какая таблица является *главной*, а какая – *подчиненной*.

Наиболее часто используются отношения *один-ко-многим*. В этом случае одной записи в главной таблице соответствует несколько записей в подчиненной таблице.

Для создания отношений необходимо указать поля в двух таблицах, которые содержат одни и те же данные. Обычно такое поле в одной из таблиц (главной) является ключевым. Имена связывающих полей могут отличаться, но типы и свойства должны совпадать.

**Задание 2.** Установить связи между данными таблиц *Факультет, Направление подготовки, Группа и Студенты*.

### **Технология.**

1. Выбрать на *Ленте* вкладку *Работа с базами данных* и найти команду *Схема данных*.

2. В окне *Добавить таблицу* последовательно выбирать указанные выше таблицы, выбирая кнопку *Добавить*.

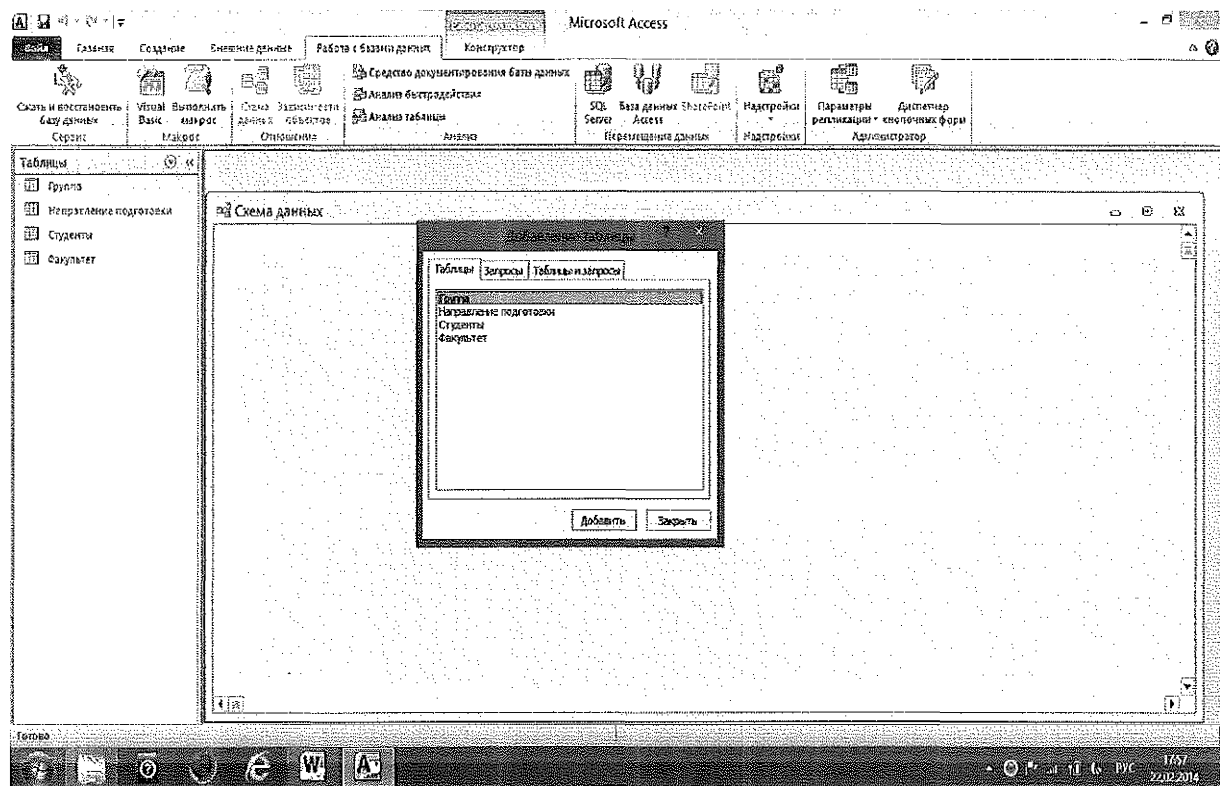


Рис. 11. Окно Добавление таблицы

3. Добавив последнюю таблицу, выбрать кнопку *Закрыть*.

4. В окне *Схема данных* установить связь между таблицей *Факультет* и таблицей *Группа*. Для этого выделить поле *Номер факультета* в таблице *Факультет* и, удерживая нажатой левую клавишу мыши, переместить указатель на поле с тем же названием в таблице *Группа*; отпустить клавишу мыши, появится окно *Изменение связи*, в котором необходимо включить флажок *Обеспечение целостности данных* и нажать кнопку *ОК*.

В окне *Схема данных* появится связь между таблицами типа **ОДИН-КО-МНОГИМ**.

*Примечание 1.* Для удаления ошибочно установленной связи следует выделить связь и, нажав правую кнопку мыши, вызвать контекстное меню, в котором выбрать пункт *Удалить связь*.

*Примечание 2.* Целостность данных может быть обеспечена при условии, что поля таблиц, которые обеспечивают связь, имеют одинаковый тип. Если при установлении связи обнаруживается разный тип полей, следует щелкнуть по таблице на схеме данных правой клавишей мыши, и в контекстном меню выбрать пункт *Конструктор таблиц*, с помощью которого изменить тип поля.

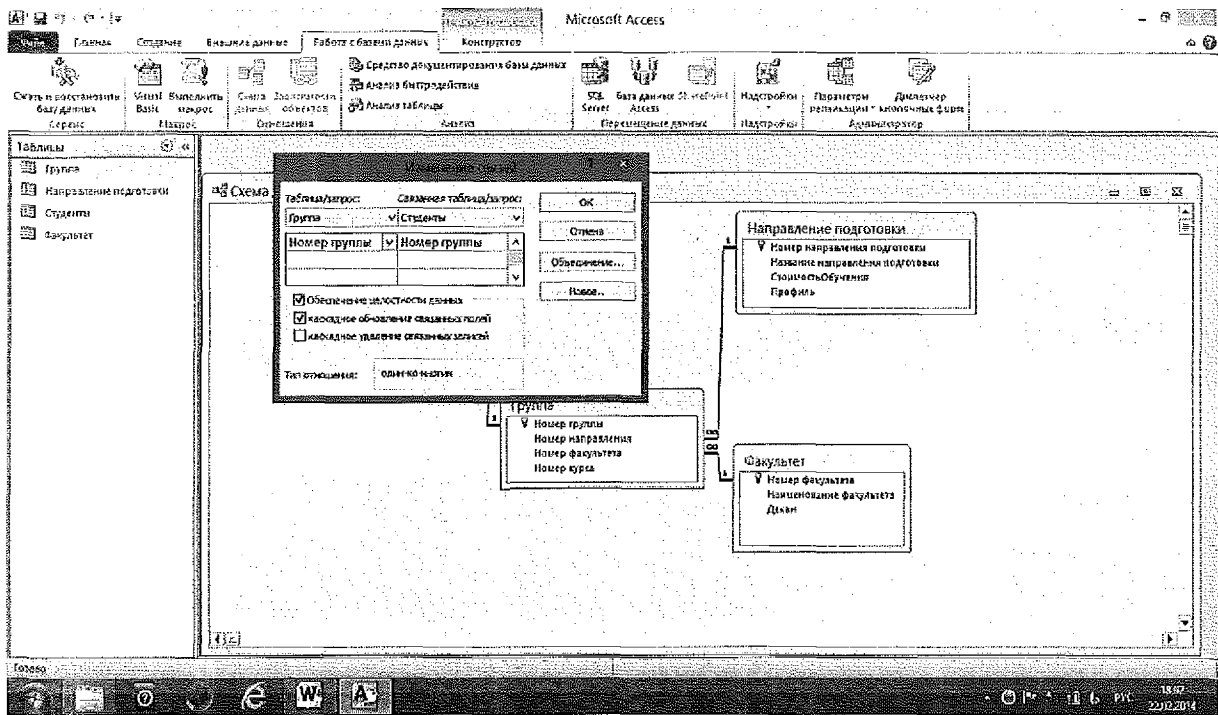


Рис. 12. Окно Изменение связей

5. Аналогично установить связь между таблицами *Направление подготовки* и *Группа* по полю *Номер направления*. Установить связь между таблицами *Группа* и *Студенты* по полю *Номер группы*.

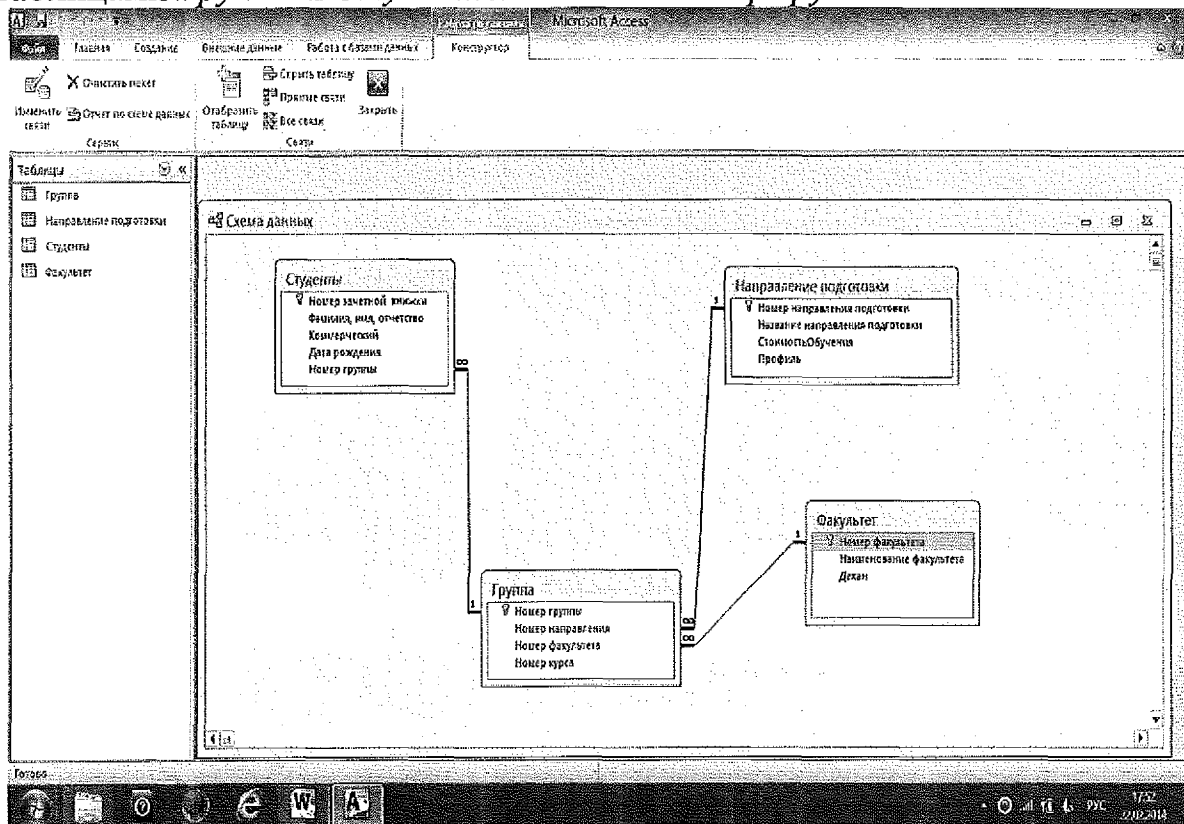


Рис. 13. Окно Схема данных

## Ввод данных в таблицы

**Задание 3.** Выбрать *Режим таблицы* на панели инструментов и заполнить таблицы данными.

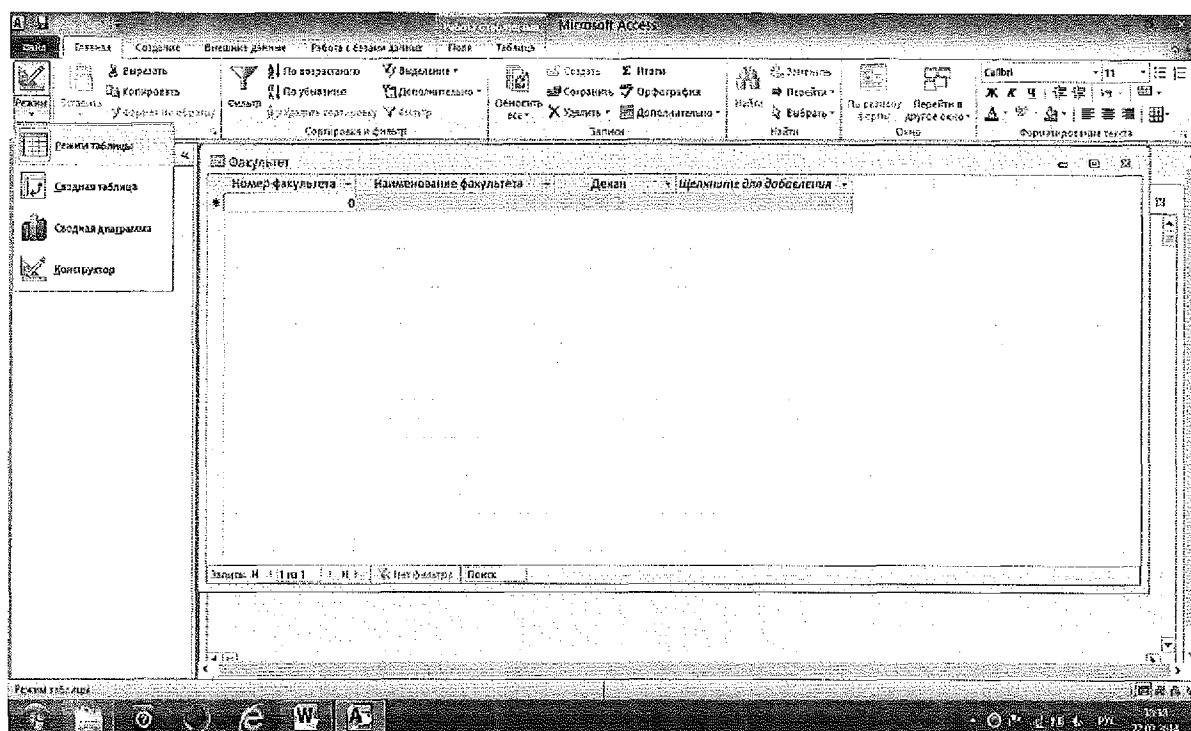


Рис. 14. Таблица Факультет

В таблице *Факультет* должно быть 4–6 строк.

В таблице *Направление подготовки* должно быть 6–10 строк.

В таблицу *Группа* следует ввести 5–6 строк, в том числе строки с теми же номерами направления, что и в таблице *Направление подготовки* и номерами факультетов из таблицы *Факультет*.

Аналогично, ввести в таблицу *Студенты* данные для 3-х групп по 10 студентов в каждой с различным набором признаков, в том числе строки с такими же номерами группы, что и в таблице *Группа*.

После ввода данных сохранить базу данных.

## 2.6 Создание запросов

**Цель работы:** конструирование и использование запросов к базе данных. Конструирование и использование запросов на изменение базы данных.

### Краткие сведения

*Запрос* – это средство для выборки данных из базы данных в форме таблицы, выполняемой по заданному условию, а также для выполнения определенных действий над табличными данными.

Условие может определять:

- порядок сортировки выводимых данных;
- фильтрацию данных;
- вычисляемые поля;
- вывод данных из нескольких связанных таблиц;

Применение запросов позволяет избежать дублирования данных в таблицах и обеспечивает максимальную гибкость при поиске и отображении данных БД. С помощью запроса можно осуществить выборку данных сразу из нескольких таблиц.

Все запросы можно разделить на две группы:

- запросы-выборки;
- запросы действия.

**Запросы-выборки** извлекают данные из таблиц в соответствии с заданными условиями.

Основные виды запросов-выборок:

- запрос-выборка по одной таблице с заданными ключами сортировки;
- запрос с критерием поиска;
- запрос перекрестный;
- запрос с параметром;
- запрос с вычисляемым полем;
- запрос с итогами;
- запрос к связанным таблицам.

**Запросы-действия** предназначены для выполнения требуемых действий над данными таблиц. Они позволяют добавлять, изменять или удалять данные. В Access существует 4 вида запросов-действия:

- запрос на удаление;
- запрос на замену (обновление);
- запрос на создание новой таблицы;
- запрос на добавление новых записей в таблицу.

При составлении запроса используется таблица описания запроса, имеющая 4 основные строки.

1-ая строка таблицы называется *Поле*. В ней содержатся имена полей. Поле может быть вычисляемым. Тогда в соответствующей ячейке содержится выражение для вычисления значения этого поля.

2-ая строка таблицы называется *Сортировка*. Она определяет способ сортировки по соответствующему полю (по возрастанию, по убыванию или без сортировки).

3-ья строка таблицы управляет *Выводом на экран* значения соответствующего поля.



4-ая строка таблицы называется *Условием отбора*. Содержит критерии, по которым производится отбор записей в динамический набор данных.

При записи в запросе условия можно объединять критерии при помощи операций **И** (AND) или **ИЛИ** (OR).

### **Использование вычисляемых полей в запросах**

Допускается создание вычисляемых полей в запросе с помощью ввода выражения в строку с именами полей.

1. В режиме конструктора запроса введите *Выражение* в пустую ячейку строки «Поле». После нажатия клавиши Enter или перевода курсора в другую ячейку будет выведено имя поля «Выражение N», где N – целое число, увеличивающееся на единицу для каждого нового создаваемого вычисляемого поля в запросе. Это имя выводится перед выражением и отделяется от него двоеточием. В режиме таблицы данное имя становится заголовком столбца.

2. Пользователь имеет возможность переименовать вычисляемое поле.

Например, для создания поля «Сумма» следует ввести в ячейку в строке «Поле» выражение: *Сумма: [Цена]\*[Количество]*

При выполнении запроса в поле «Сумма» будет занесено новое значение, найденное путем перемножения содержимого полей «Цена» и «Количество».

Вычисляемые поля позволяют:

- рассчитывать числовые значения и даты;
- комбинировать значения в текстовых полях;
- создавать подчиненные запросы;
- рассчитывать значения итоговых полей с помощью групповых функций

Для вычисляемых полей допускается сортировка, задание условий отбора и расчет итоговых значений. Для расчета итоговых значений следует выбрать в ячейке строки «Групповая операция» элемент «Выражение».

Ввод данных в вычисляемые поля или их изменение в режимах таблицы или формы невозможен.

В Microsoft Access определена специальная функция, **Count**, обеспечивающая подсчет числа записей в запросе. Для этого следует ввести **Count(\*)** в ячейку в строке полей.

Окно «Область ввода», выводящееся нажатием клавиш Shift+F2, позволяет просматривать выражения целиком (без прокрутки).

### **Элементы выражений**

Элементами выражения могут быть операторы, идентификаторы, функции, литералы и константы.

**Оператор** =, <, >, &, And, Or, Like определяет операцию, выполняемую над одним или несколькими элементами.

**Идентификатор** задает ссылку на значение поля элемента управления или свойства.

**Функция** Date, Sum, DLookup возвращает значение, найденное в результате расчета или другой операции.

**Литерал** 100, #1-января-94#, "New York" представляет значение, например, число, строку или дату, которое используется в Microsoft Access, именно в том виде, как оно записано. Даты заключаются в символы номера (#), а строки в прямые кавычки (").

**Константа** True, False, Да, Нет, Null представляет значение, остающееся неизменным.

**Задание 1.** Создать простой запрос для выборки сведений из базы данных, включающий следующие поля:

- Наименование факультета,
- Номер группы,
- Наименование направления подготовки,
- Номер зачетной книжки,
- Фамилия, имя, отчество.

### **Технология.**

1. В окне *База данных* выбрать объект *Запросы*.
2. В появившемся списке выбрать *Создание запроса в режиме конструктора*.
3. В появившемся окне *Добавление таблицы* добавить таблицы *Студент*, *Группа*, *Направление подготовки*, *Факультет*, затем закрыть окно.

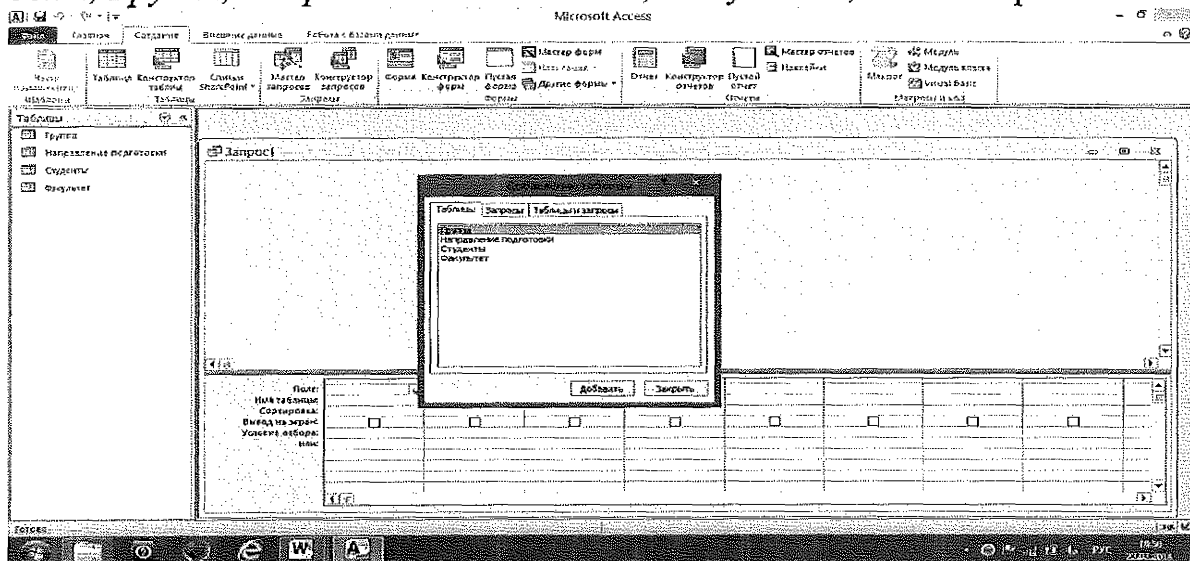


Рис. 15. Добавление таблиц в окно запроса

4. Перетащить левой кнопкой мыши соответствующие названия полей (см. задание 1) из окон таблиц в бланк запроса (поле, имя таблицы, сортировка, вывод на экран, условие отбора), расположенный под схемой данных, соблюдая заданный их порядок.

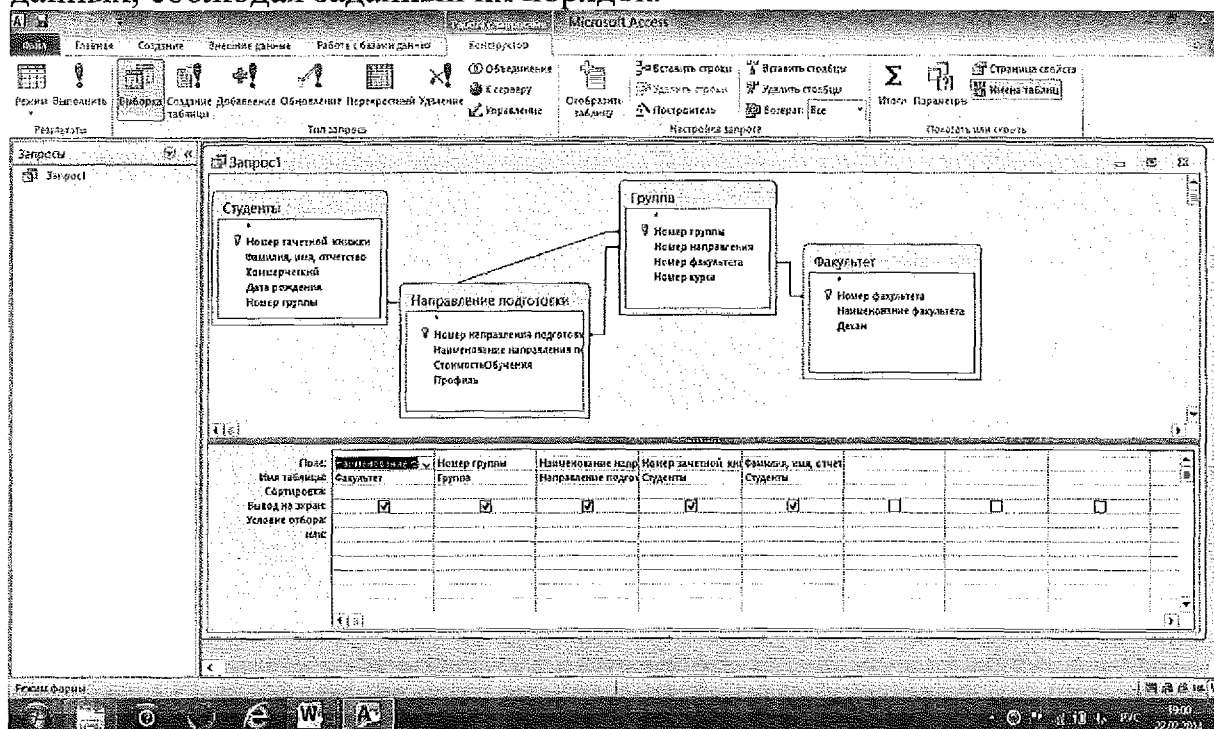


Рис. 16. Добавление полей в окно запроса

*Примечание.* Вместо перетаскивания названия поля можно использовать двойной щелчок левой клавишей мыши по соответствующей строке в окне таблицы на схеме данных.

5. Для просмотра полученной выборки, выполнить команду **Режим таблицы**.

6. Вернуться в режим **Конструктора запросов** в случае, если выборка содержит ошибки, и откорректировать запрос.

7. Закрыть запрос. Появится диалоговое окно, в котором следует подтвердить необходимость его **сохранения**. Присвоить запросу имя «Запрос 1».

**Задание 2.** Создать простой запрос для выборки сведений из базы данных, включающий те же поля, что и предыдущий запрос, но содержащий сведения только о студентах, обучающихся на коммерческой основе. Такой запрос называется **запросом по условию**.

Для выполнения этого задания необходимо на бланке запроса добавить поле «Коммерческий» и ввести по нему соответствующее условие (в *Условие отбора*, где поле *Коммерческий* ввести «Да»). Присвоить запросу имя «Запрос 2».

**Задание 3.** Составить запрос для подсчета количества студентов, обучающихся на коммерческой основе, в каждой группе.

**Технология.**

1. Создать в режиме конструктора новый запрос с использованием таблиц *Факультет, Группа, Студент*.

2. Ввести в 1-ый столбец бланка запроса поле *Наименование факультета*, во 2-ой столбец – поле *Номер группы*, в 3-ий столбец – поле *Коммерческий*.

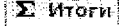
3. Установить в качестве *Условия отбора* для 3-го столбца значение «Да».

4. Установить для 1-го и 2-го столбцов бланка сортировку по возрастанию.

5. Отключить вывод на экран данных 3-го столбца.

6. Ввести в 4-ый столбец поле *Коммерческий* и замените название столбца на *Количество коммерческих: Коммерческий*. Для этого новое и старое названия столбца следует разделить символом «двоеточие», т. е. ячейка с названием поля должна содержать (без пробелов):

**Количество коммерческих:Коммерческий**

7. На ленте выбрать команду *Итоги*  . В появившейся строке запроса *Групповые операции* выбрать для 4-го столбца операцию *Count*.

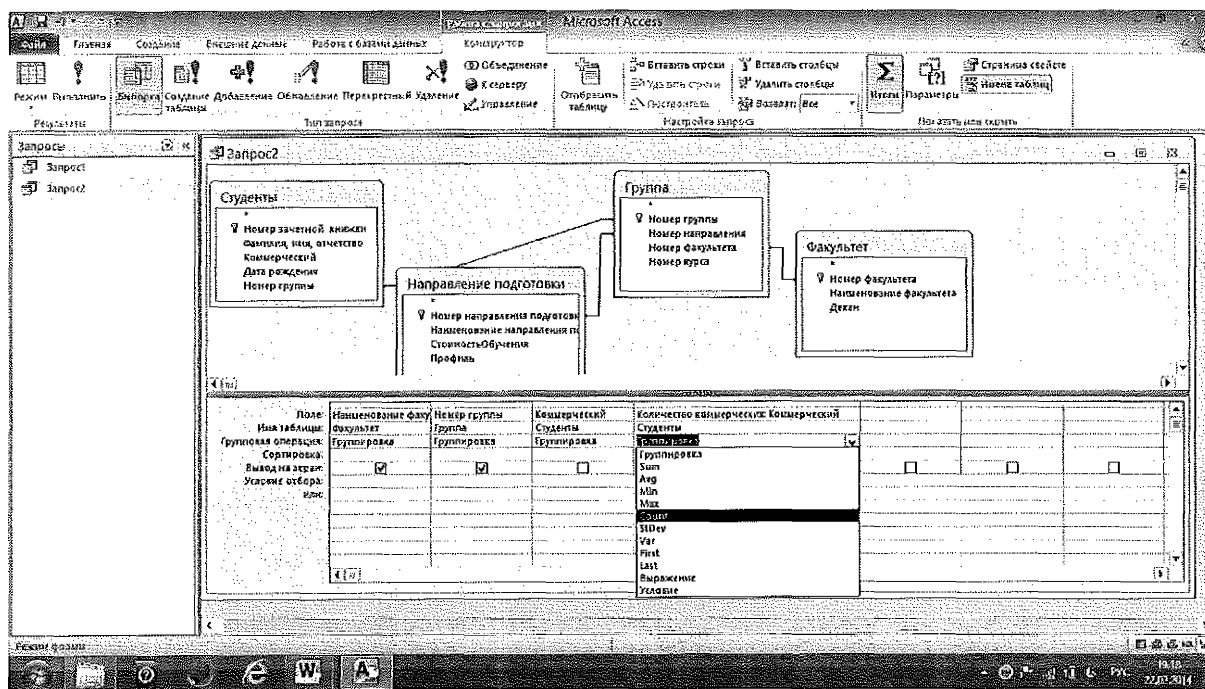


Рис. 17. Бланк запроса для подсчета коммерческих студентов по группам

8. Просмотреть выборку с подсчетом итогов в *Режиме таблицы*.

9. Вернуться в режим конструктора запросов.

10. Сохранить запрос, присвоив ему имя «Запрос 3 Подсчет коммерческих студентов по группам».

**Задание 4.** Составить запрос на выборку для подсчета количества коммерческих студентов по каждому факультету. Сохранить как «Запрос 4 Подсчет Коммерческих студентов по факультетам».

**Задание 5.** Составить запрос, позволяющий увидеть выборку, отражающую количество студентов по каждому факультету и каждой группе. Заголовки столбцов должны соответствовать названиям факультетов, заголовки строк – номерам групп. Такой вид выборки может быть реализован **перекрестным запросом**. Для применения подобного запроса желательно иметь в базе данных сведения по 5–6 группам, отнесенным к 3 факультетам.

**Технология.**

1. Создать с помощью конструктора новый запрос с использованием таблиц *Факультет*, *Группа*, *Студент*.

2. Ввести в 1-ый столбец бланка запроса поле *Наименование факультета*, во 2-ой столбец – поле *Номер группы*, в 3-ий столбец – поле *Коммерческий*.

3. На панели инструментов найти тип запроса **Перекрестный запрос**, появится дополнительно строка в бланке запроса **Перекрестная таблица**.

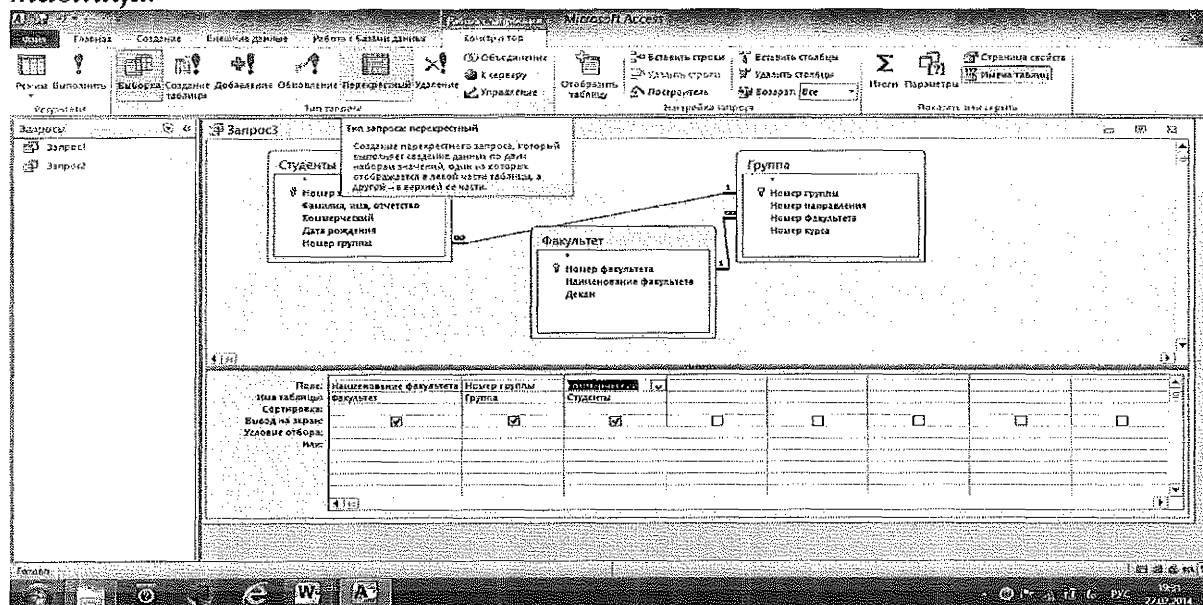


Рис. 18. Выбор вида запроса **Перекрестный**

4. Выбрать значения в строке бланка **Перекрестная таблица**, развернув список в ячейках:

- для 1-го столбца выбрать значение для *Перекрестной таблицы заголовков строк*,
  - для 2-го столбца – *заголовок столбцов*,
  - для 3-го столбца – *значение*.
5. Выбрать функцию **Count** для групповой операции в 3-м столбце.

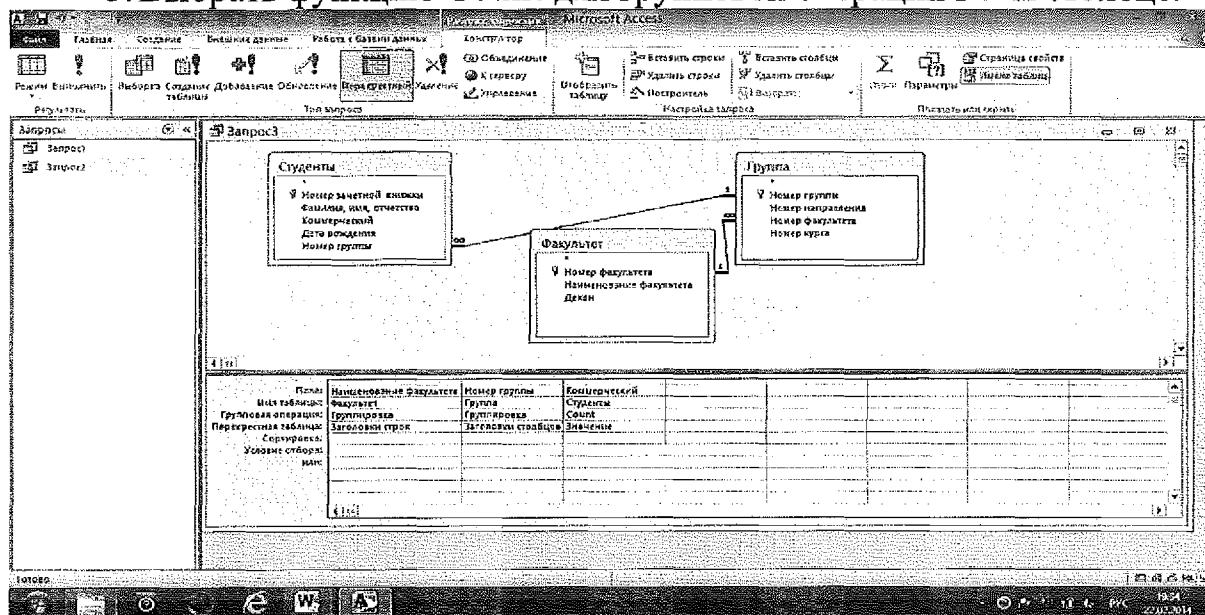


Рис. 19. Бланк запроса

6. Просмотреть перекрестную выборку, запустив запрос в режиме таблицы.
7. Сохранить запрос, присвоив ему имя «**Запрос 5 перекрестный**».

**Задание 6.** Составить запрос для вывода списков групп, причем номер группы должен запрашиваться в процессе выполнения запроса. Такой запрос называется **запросом с параметром**. Параметром является **номер группы**. Значение параметра вводится в диалоговом окне. Для создания запроса необходимо ввести в ячейку с условием текст условия отбора в квадратных скобках.

#### **Технология.**

1. Создать с помощью *конструктора* новый запрос с использованием одной таблицы *Студент*.
2. Ввести в 1-ую строку бланка запроса все поля таблицы.
3. Ввести в ячейку строки *Условие отбора* для поля *Номер группы* текст: **[Введите номер группы]**.

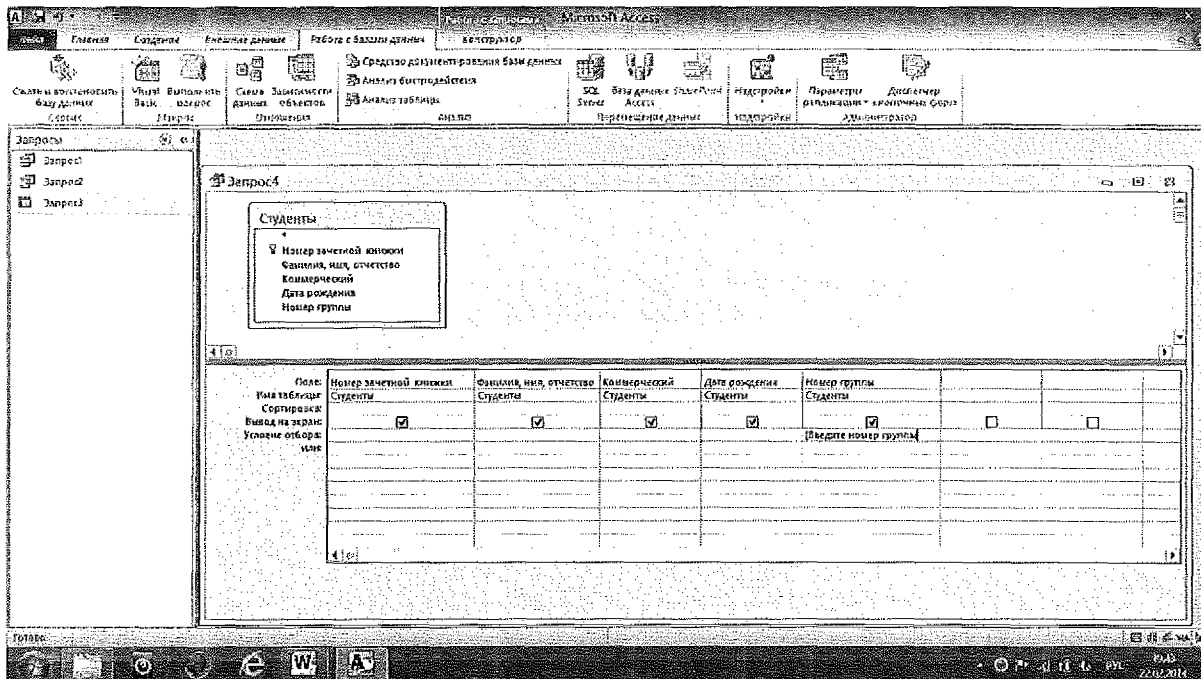


Рис. 20. Бланк запроса с параметром

4. Запустить запрос и в появившееся диалоговое окно ввести номер одной из групп. Просмотреть выборку. Сохранить запрос, присвоив ему имя «Запрос 6 с параметром».

**Задание 7.** Составить запрос для получения справки по стоимости обучения коммерческих студентов с учетом НДС (налоговая добавленная стоимость). Принять, что стоимость обучения с учетом НДС коммерческого студента равна стоимости обучения по выбранному им направлению подготовки, умноженной на  $(1+0,2)$ , где 0,2 – величина НДС.

Такой запрос называется **запросом с вычисляемым полем**.

**Технология.**

1. Создать с помощью **конструктора** новый запрос с использованием таблиц *Направление подготовки*, *Группа*, *Студент*.

2. Ввести в бланк запроса поля: *Номер группы*, *Номер направления подготовки*, *Фамилия, имя, отчество*, *Номер зачетной книжки*, *Коммерческий*, *Стоимость обучения*.

3. Ввести в поле *Коммерческий* условие отбора «Да» и отключить вывод содержимого этого поля на экран.

4. Выбрать поле следующего свободного столбца. Вызвать контекстное меню правой кнопкой мыши нажав по этому пустому полю и выбрать **Построить**. Будет вызвано окно **Построителя выражений**.

5. Ввести в окно построителя выражений: **НДС: [Стоимость обучения]\*0,2**

где НДС: – заголовок столбца с вычисляемым полем, [Стоимость обучения] – поле таблицы *Направление подготовки*, 0,2 – величина НДС.

6. Выбрать поле следующего свободного столбца, перейти в окно **Построителя выражений** и ввести в нем формулу (без пробелов):

**Итого:[Стоимость обучения]+[НДС]**

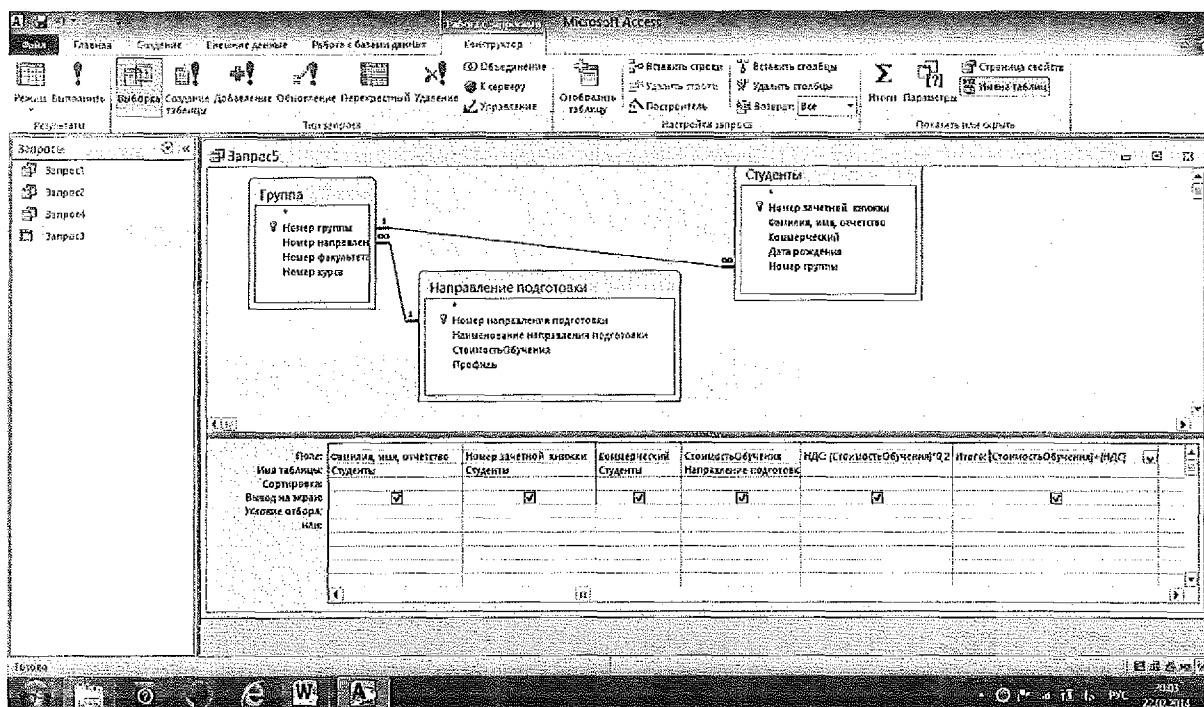


Рис. 21. Бланк запроса с вычисляемым полем

7. Включить флажки вывода на экран вычисляемых полей.

8. Запустить запрос и сохранить его с именем «**Запрос 7 с вычисляемым полем**».

**Задание 8.** Составить запрос на обновление базы данных. Необходимо создать запрос, при выполнении которого произойдет увеличение стоимости обучения в 1,5 раза по всем направлениям подготовки.

**Технология.**

1. Создать с помощью конструктора новый запрос. Выбрать в качестве исходной таблицы одну таблицу – *Направление подготовки*.

2. Указать тип запроса **Тип запроса/Обновление**.

3. Добавить в запрос поле *Стоимость обучения* и ввести в него с помощью **Построителя выражений** в строку *Обновление* выражение: **[Стоимость обучения]\*1,5**



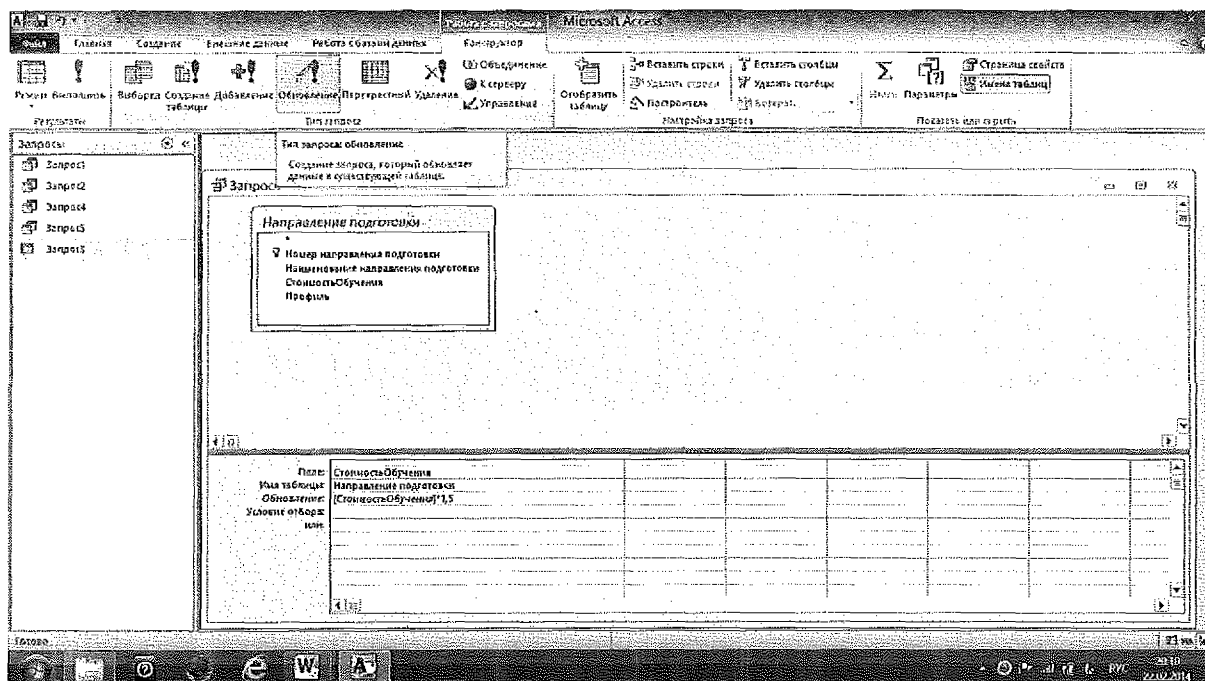


Рис. 22. Бланк запроса на обновление

4. Выполнить запрос и сохранить, присвоив имя «Запрос 8 на обновление».

5. После запуска запроса просмотреть результаты в таблице *Направление подготовки*, где должна измениться в сторону увеличения стоимость обучения в 1,5 раза.

**Примечание:** Для того чтобы запрос на обновление был запущен необходимо выбрать вкладку *Работа с БД* и поставить галочку на ленте, выбрав команду *Панель сообщений*. Под лентой выбрать кнопку *Параметры сообщений* и в открывшемся окне выбрать *Включить это содержимое*.

Если панель сообщений не активизирована, то выбрать в левом верхнем углу *Файл / Параметры / Центр управления безопасностью* кнопку *Параметры центра управления безопасностью / Панель сообщений* / поставить точку *Показывать панель сообщений*, поставить галочку *Включить запись в журнал* / ОК.

**Задание 9.** Составить запрос на обновление базы данных. Необходимо создать запрос, при выполнении которого произойдет уменьшение стоимости обучения по двум направлениям подготовки на 10 %. Сохранить как «Запрос 9 на уменьшение стоимости обучения».

**Задание 10.** Составить запрос на создание новой таблицы – *Ведомость 1*. Предварительно необходимо дополнить базу данных еще двумя таблицами *Дисциплина* и *Вспомогательная*.

Создать таблицу *Дисциплина*, которая должна иметь следующий состав полей:

- *Код дисциплины* – ключевое поле, длинное целое;
- *Наименование дисциплины* – текстовое поле;
- *Лекции (количество часов)* – числовое поле;
- *Практика (количество часов)* – числовое поле;

Ввести в таблицу *Дисциплина* 5–6 строк с различными кодами и наименованиями дисциплин.

Создать таблицу *Вспомогательная*. В таблицу *Вспомогательная* включить только одно поле:

- *Оценка* – числовое, ключевое поле не указывать.

Ввести в таблицу *Вспомогательная* одну запись, причем значение поля *Оценка* не заполнять.

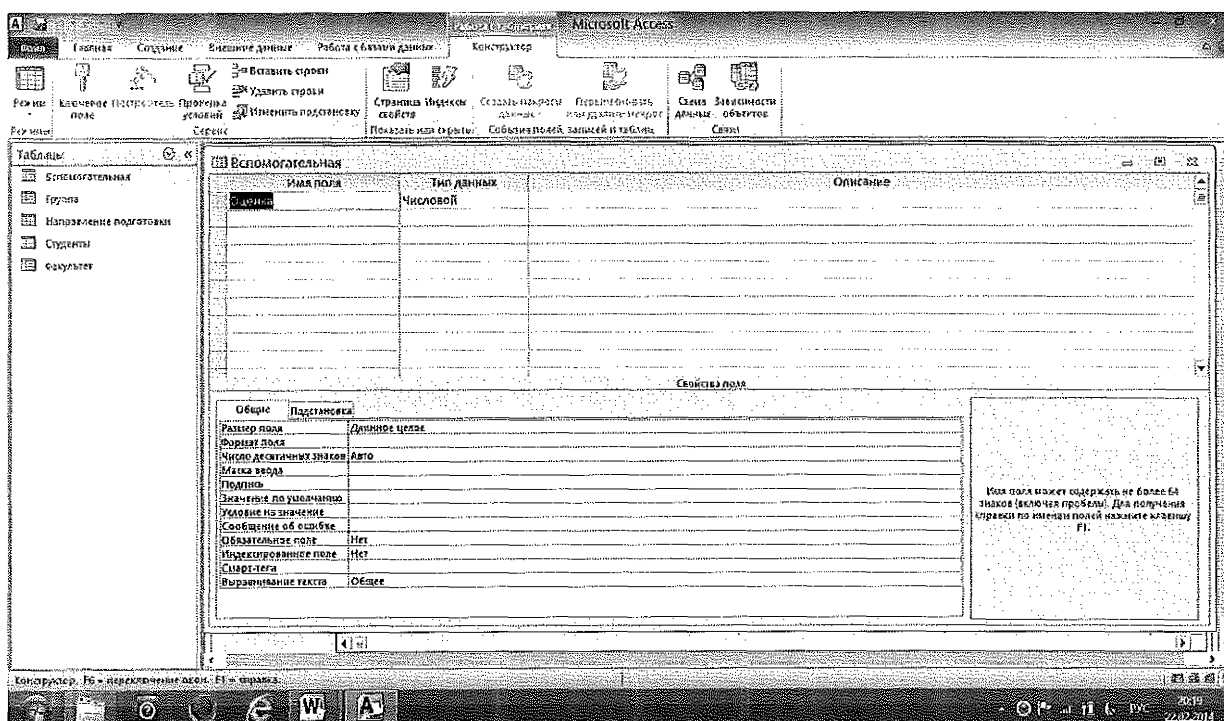


Рис. 23. Вспомогательная таблица в режиме Конструктора

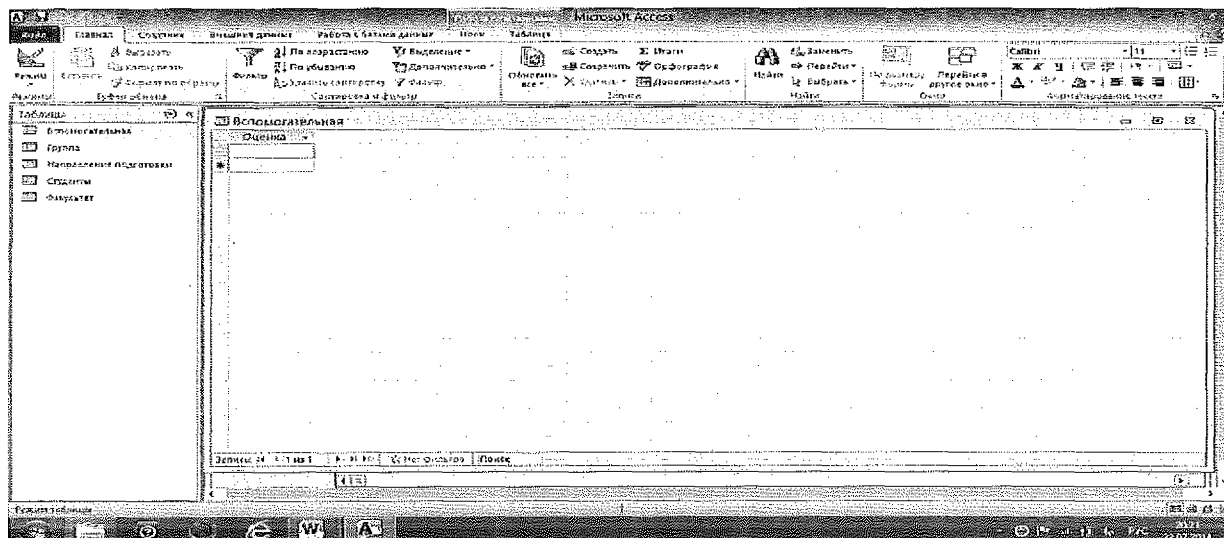


Рис. 24. Заполненная Вспомогательная таблица с пустой записью

### Технология.

1. Создать с помощью конструктора новый запрос. Выбрать качестве исходных таблицы – *Студент*, *Дисциплина* и *Вспомогательная*.
2. Указать тип запроса, выбрав его на ленте сверху, **Создание таблицы** и присвоить таблице имя *Ведомость 1*.

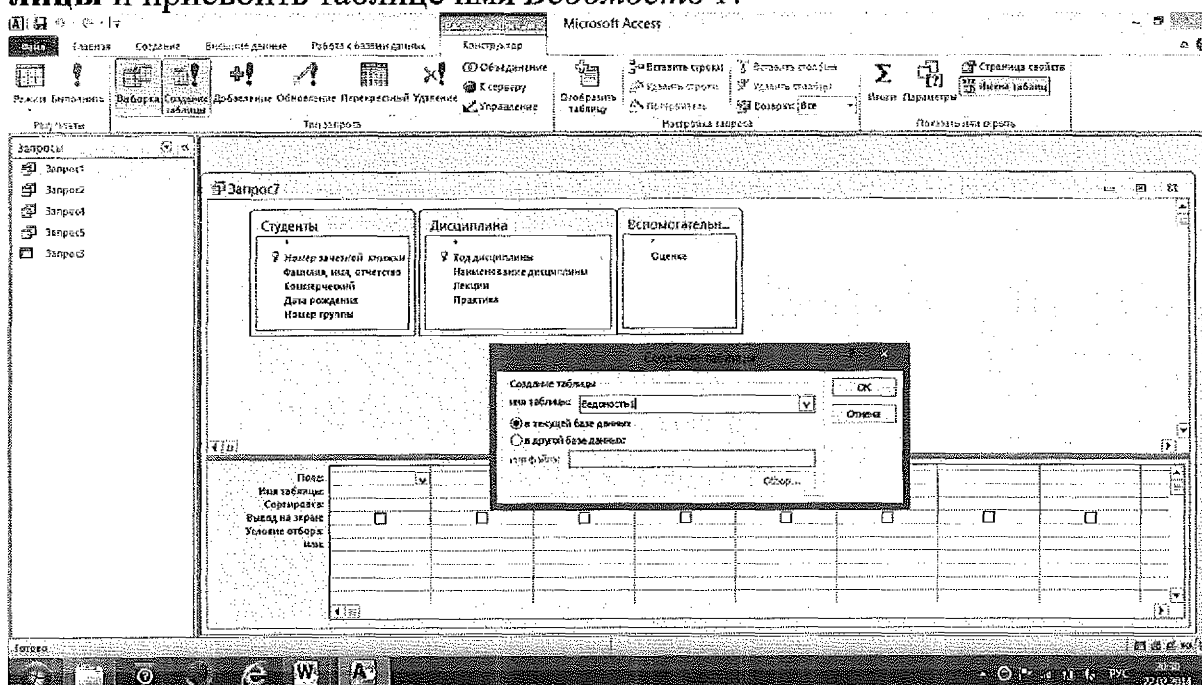


Рис. 25. Бланк запроса на создание новой таблицы

3. Ввести в бланк запроса поля:

- Номер группы;
- Код дисциплины;

- Наименование дисциплины;
- Фамилия, имя, отчество;
- Номер зачетной книжки;
- Оценка.

4. Ввести для поля *Номер группы* параметрическое условие отбора:  
**[Введите номер группы:]**

5. Ввести для поля *Код дисциплины* параметрическое условие отбора:  
**[Введите код дисциплины:]**

6. Сохранить запрос как «**Запрос 10**».

7. Перед запуском запроса прочитать **замечание**. Запустить запрос 6 раз, т. е. с использованием запроса **создать 6 ведомостей: по двум дисциплинам для каждой из 3-х групп**. Номера групп и коды дисциплин задавать в окне ввода параметров. Присвоить созданным таблицам имена *Ведомость 1*, *Ведомость 2* и т. д.

8. Заполнить созданные таблицы (*Ведомость 1*, *Ведомость 2* и т. д.) **сведениями об оценках**, полученных студентами при сдаче экзаменов.

*Замечание.*

1. Поскольку при повторном использовании запроса на создание таблицы ранее созданная таблица удаляется необходимо перед созданием новой таблицы переименовывать ранее созданную таблицу. Т.е. при запуске запроса 10, создается таблица *Ведомость 1*. Ее нужно переименовать в *Ведомость 2* и опять запустить запрос 10, после запуска которого опять появится таблица *Ведомость 1*, но уже с другими данным. *Ведомость 1* переименовываем в *ведомость 3* и т. д. И так нужно создать 6 ведомостей для двух дисциплин и 3 групп.

Для переименования необходимо:

- выделить исходную таблицу и вызвать правой кнопкой мыши контекстное меню;
- выполнить пункт **Переименовать**.

**Задание 11.** Составить запрос на добавление записей из одной таблицы в другую. Объединить с помощью такого запроса все заполненные экзаменационные ведомости в одну, назвав ее, *Общая ведомость*.

*Технология.*

1. Создать в окне базы данных на вкладке *Таблицы* копию таблицы *Ведомость 1* и заменить ее имя *Общая ведомость* (щелчок правой клавишей по имени таблицы и вызов соответствующей команды из контекстного меню).

2. Создать с помощью **конструктора** новый запрос. Выбрать в качестве исходной таблицу *Ведомость 2*.

3. Указать тип запроса **Добавление**. В окне *Добавление* ввести имя таблицы *Общая ведомость*, к которой должны добавляться строки исходной таблицы.

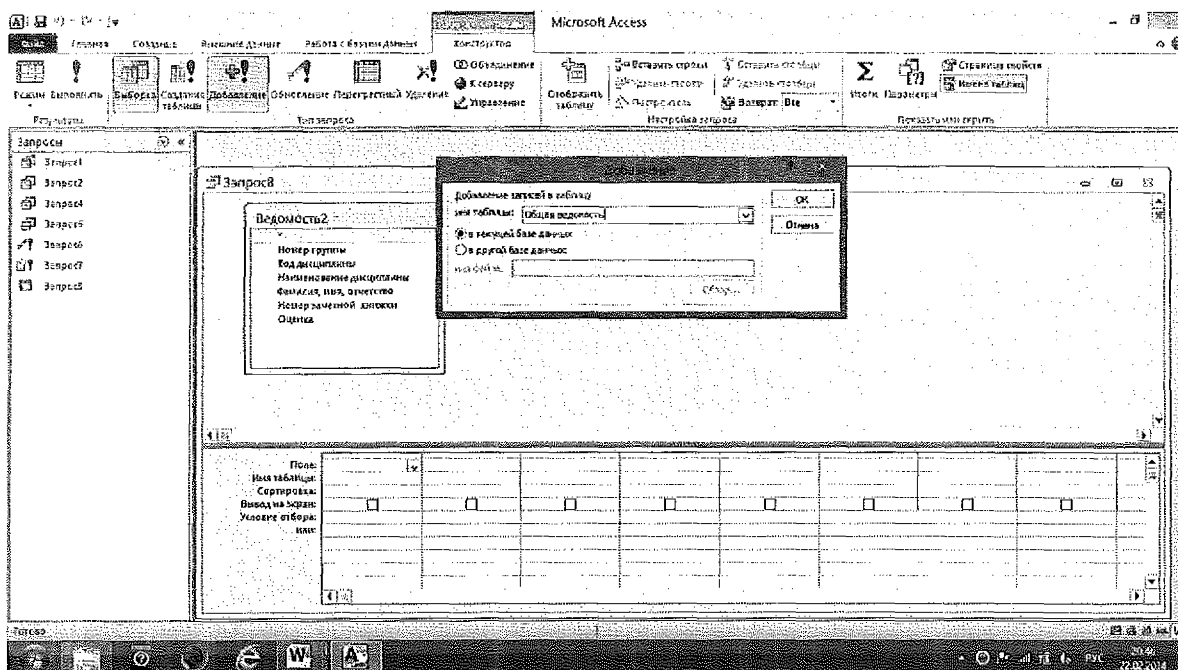


Рис. 26. Бланк запроса на добавление записей из одной таблицы в другую

4. Ввести в строку *Поле* бланка запроса все поля таблицы *Ведомость 2*.

5. Закрывать запрос, присвоив ему имя **«Запрос 11 на добавление данных в таблицу»**.

6. Выполнить запрос и просмотреть таблицу *Общая ведомость*, которая должна пополниться записями из таблицы *Ведомость 2*.

7. Временно переименовывая другие ведомости в *Ведомость 2* и запуская **Запрос 11** каждый раз, добавить их содержимое в *Общую ведомость*.

**Задание 12.** Составить параметрический запрос на удаление данных из таблицы *Общая ведомость* о студентах одной из групп по заданной дисциплине.

**Технология.**

1. Создать с помощью конструктора новый запрос. Выбрать в качестве исходной таблицу *Общая ведомость*.

2. Ввести в бланк запроса поля *Номер группы* и *Код дисциплины*.

3. Указать тип запроса **Удаление**.

4. Ввести в строку *Условие отбора* соответственно для каждого из полей параметрические условия **[Введите номер группы:]** и **[Введите код дисциплины:]**

5. Закрывать и сохранять запрос под именем «Запрос 12 на удаление данных из таблицы».

6. Выполнить запрос и просмотреть результаты его выполнения.

7. Воспользовавшись таблицами *Ведомость 1 – Ведомость 6* и *Запросом на добавление* добавить в *Общую ведомость* удаленную часть.

## 2.7 Создание форм

**Цель работы:** освоение технологии конструирования и использования форм.

### **Краткие сведения**

Форма является удобным средством для просмотра БД, а также для ввода данных и их корректировки. Форма обычно отображает поля одной строки таблицы или запроса. В форме можно отображать данные нескольких таблиц или запросов.

Форма содержит следующие разделы: *заголовок, область данных, верхний и нижний колонтитулы и примечание.*

**Заголовок формы** может содержать название формы, инструкции по ее использованию, а также кнопки, предназначенные для открытия подчиненных форм и выполнения других задач. В режиме формы заголовок отображается в верхней части экрана, а при печати – в верхней части первой страницы.

**Область данных** содержит записи. На экране одновременно их может располагаться от одной до нескольких.

**Примечание формы** может содержать кнопки и инструкции по использованию формы. В режиме формы примечание отображается в нижней части экрана, а при печати – на последней странице после области данных.

**Верхний колонтитул** может содержать заголовок формы, графические объекты, заголовки столбцов и любые другие сведения, которые должны изображаться в верхней части каждой страницы. Верхние колонтитулы используются только при печати формы.

**Нижний колонтитул** может содержать дату, номер страницы и любые другие сведения, которые должны отображаться в нижней части каждой страницы. Нижний колонтитул появляется только при печати формы.

При проектировании формы можно использовать текстовые строки, рисунки и линии, кнопки, списковые окна и т.п. Форма проектируется при помощи панели элементов.

В форме может быть разрешено или запрещено корректировать определенные данные или вообще ввод новых записей.

С формой можно работать в 3-х режимах: в режиме **конструктора**; в режиме **формы**; в режиме **таблицы**.

### **Основные виды форм:**

- простая форма по одной таблице;
- простая форма по связанным таблицам;
- простая форма на основании запроса;
- форма со списком или с полем для списка;
- составная форма;
- форма-меню с кнопками управления.

Составная форма содержит главную форму и подчиненные ей формы. Подчиненные формы позволяют решать следующие задачи:

- отображать отношения один-к-одному или один-ко-многим между таблицами или запросами, например, в главной форме выводят только категорию товаров, а в подчиненной – полный список товаров, относящихся к этой категории;
- обновлять записи в разных таблицах с помощью одной формы.

*Примечание.* Для отказа от данных только что введенных в запись необходимо 2 раза нажать клавишу Esc. В режиме конструктора форму можно преобразовать в отчет, если выполнить команду **Файл / Сохранить как отчет**.

**Задание 1.** Создать простую форму по одной таблице *Студенты*, включив в нее все поля таблицы.

### **Технология.**

1. Выбрать в окне базы данных в списке объектов *Формы* и выбрать режим *Создание формы с помощью мастера*.
2. В окне *Создание форм* в списке *Таблицы и запросы* выбрать таблицу *Студент*, в списке доступных полей выбрать все поля таблицы и нажать кнопку *Далее*.
3. Выбрать внешний вид формы *В один столбец* и нажать кнопку *Далее*.
4. Выбрать любой стиль оформления и нажать кнопку *Далее*.
5. Присвоить имя форме *Студенты* и нажать кнопку *Готово*.
6. Просмотреть информацию из таблицы *Студент* с помощью созданной формы.

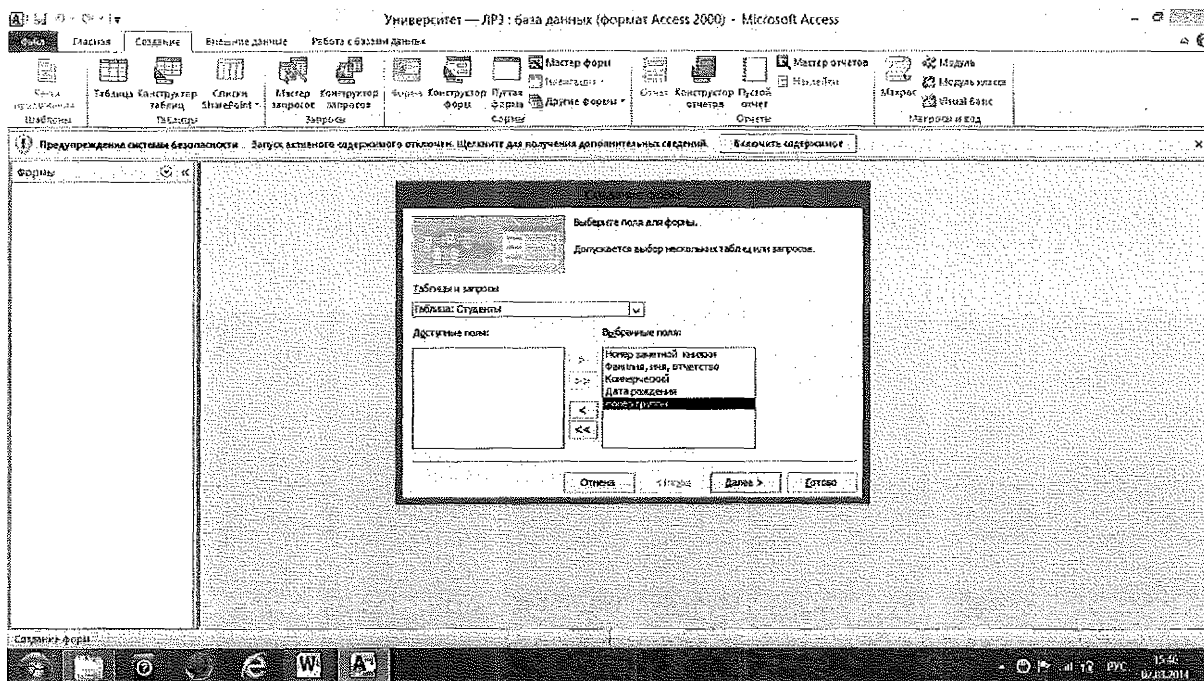


Рис. 27. Простая форма по таблице *Студенты* в режиме мастера форм

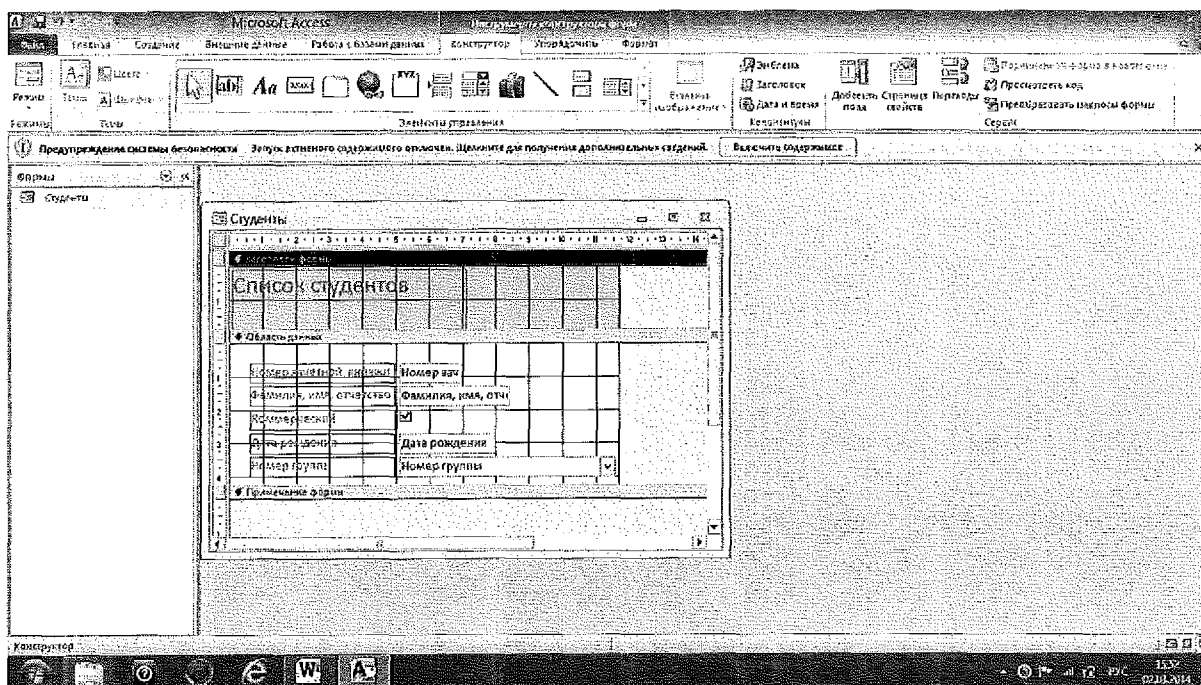


Рис. 28. Простая форма по таблице *Студент* в режиме конструктора

7. Выполнить редактирование формы *в режиме конструктора* – ввести заголовок формы «Список студентов».

Для этого:

– установить режим *Конструктор*;



- с помощью вертикальной линейки создать достаточную область для ввода заголовка формы;
- выбрать на панели элементов управления (на ленте) *Надпись* и отметить границы надписи в области заголовка, ввести текст *Список студентов* в рамку надписи;
- выделить рамку надписи и отрегулировать ее размер и местоположение.

Просмотреть форму в режиме *Форма*.

Закрывать форму, сохранив изменения и назвав форму *Студенты*.

**Задание 2. Создать простую форму по связанным таблицам.** Создать форму для просмотра перечня групп с расшифровкой направлений подготовки и названий факультетов. Для создания формы – использовать 3 таблицы: *Группа*, *Направление подготовки* и *Факультет*.

**Технология.**

1. В окне базы данных установить объект *Формы* и выбрать режим *Создание формы с помощью мастера*.

2. В окне *Создание форм* выбрать из таблицы *Группа* поле *Номер группы*, из таблицы *Факультет* – поле *Наименование факультета*, из таблицы *Направление подготовки* – поля *Номер направления подготовки* и *Наименование направления подготовки*. Нажать кнопку *Далее*.

3. На следующем шаге в диалоговом окне выбрать вид представления данных: *Группа*. Нажать кнопку *Далее*.

4. Выбрать внешний вид формы – *в один столбец* и стиль – *стандартный*. Нажать кнопку *Далее*.

5. Присвоить имя форме – **Простая форма по 3 связанным таблицам**. Нажать кнопку *Готово*.

6. Просмотреть данные, выводимые в форму.

7. Открыть форму в режиме конструктора и установить дату. Создать поле, используя значок *Поле* на панели элементов управления ленты, вместо надписи *Поле1* ввести слово *Дата*, а по полю нажать правой кнопкой мыши и вызвать *Окно свойств*. Выделить в *Окне свойств* для данного поля на вкладке *Данные* в строке *Данные* ввести значение *=Now()*.

8. Ввести в заголовке формы надпись *Список групп по факультетам*.

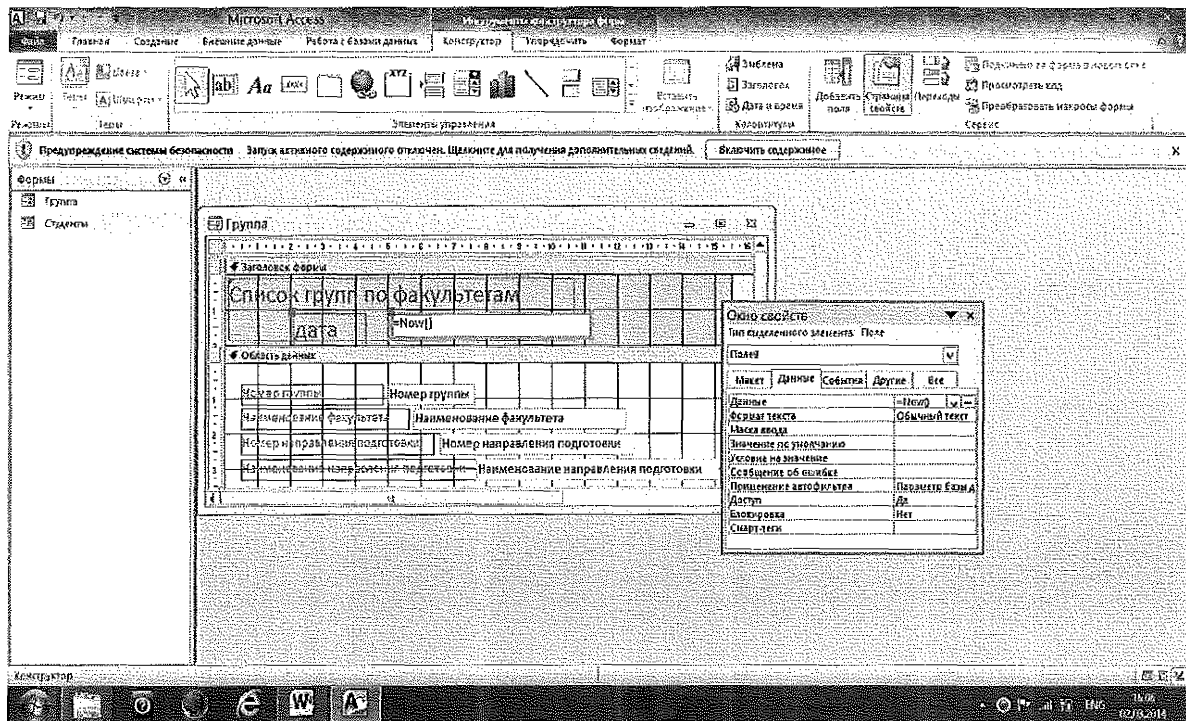


Рис. 29. Простая форма по трём связанным таблицам

### **Задание 3. Создать простую форму на основании запроса.**

Необходимо создать форму с помощью *Мастера форм* по запросу с вычисляемым полем (запрос для получения справки по стоимости обучения коммерческих студентов с учетом НДС – задание № 7 из лабораторной работы № 2), включив в форму все поля запроса. Порядок действий аналогичен порядку, рассмотренному в предыдущем задании. Присвоить форме имя **Простая форма по запросу**.

### **Задание 4. Создать форму со списком.**

Необходимо создать форму, позволяющую просматривать и редактировать данные каждого студента. Форма должна быть снабжена списком групп, в котором для каждой группы указывается ее номер направления подготовки и наименование факультета. Для создания формы следует использовать таблицу *Студенты* и специально подготовленный запрос *Группа-факультет*. Образец формы приведен на рисунке 12.

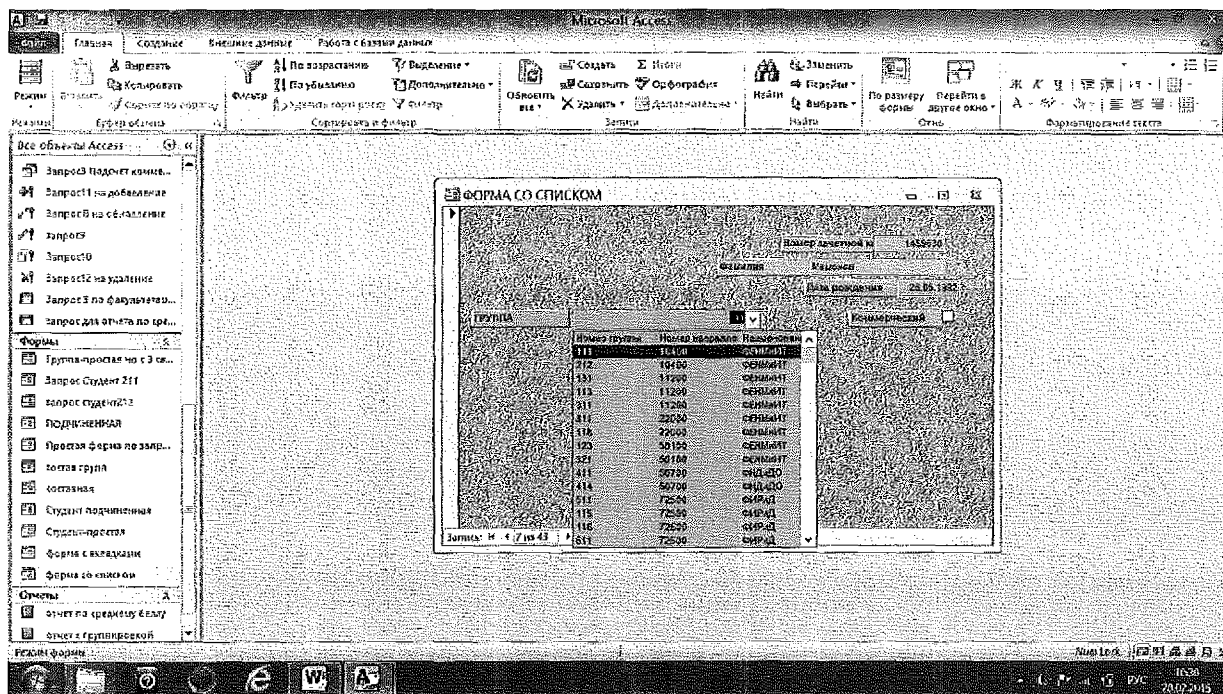


Рис. 30. Форма со списком

**Технология.**

1. Подготовить запрос на выборку по таблицам *Группа*, *Факультет*, включив в него поля:

- Номер группы;
- Номер направления подготовки;
- Наименование факультета.

Присвоить запросу имя *Группа-факультет*.

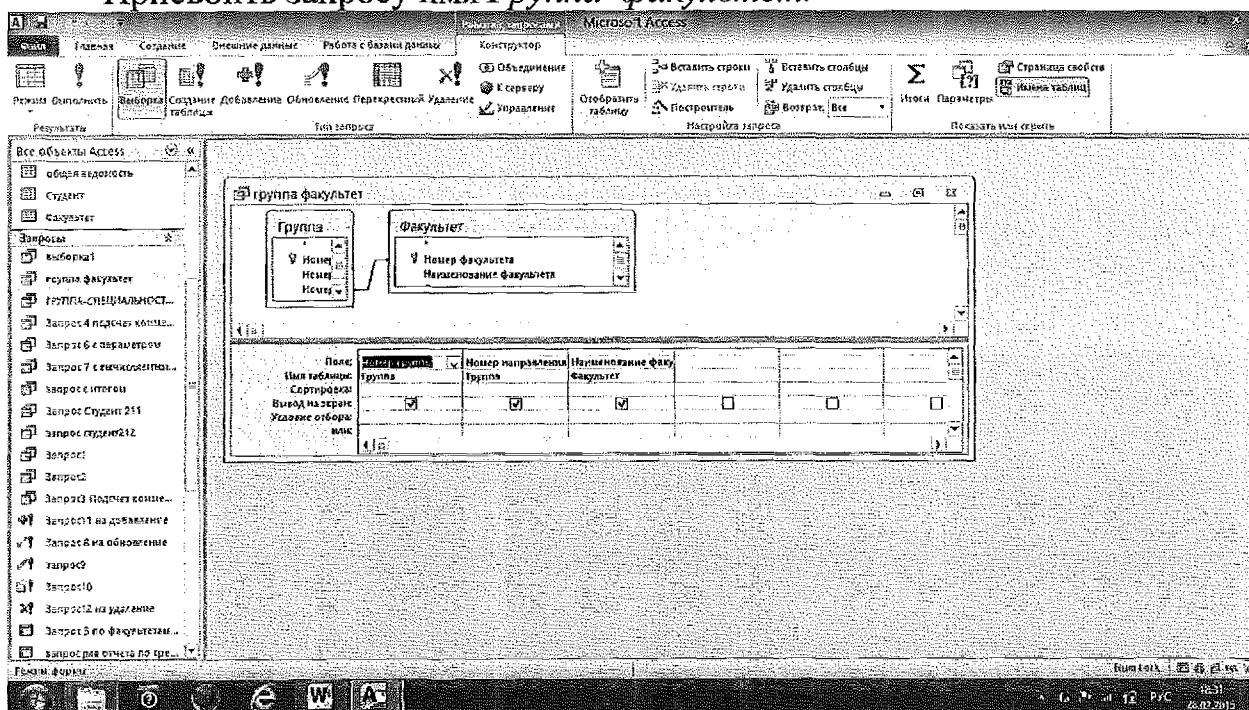


Рис. 31. Запрос *Группа-Факультет* в режиме конструктора

2. Создать с помощью *Мастера форм* простую форму в один столбец по таблице *Студенты*, включив в нее поля:

- Фамилия, имя, отчество;
- Номер зачетной книжки;
- Дата рождения;
- Коммерческий.

3. В окне *Создание форм* присвоить форме имя *Форма со списком*. Включить флажок *Изменение макета формы* для непосредственного перехода в режим конструктора.

4. В режиме конструктора переместить вниз горизонтальную линию, разделяющую область данных и примечание.

5. Выбрать на панели элементов управления формы элемент *Поле со списком*. Указать местоположение поля со списком в *Области данных* формы. В окне *Создание полей со списком* нажать кнопку *Отмена*.

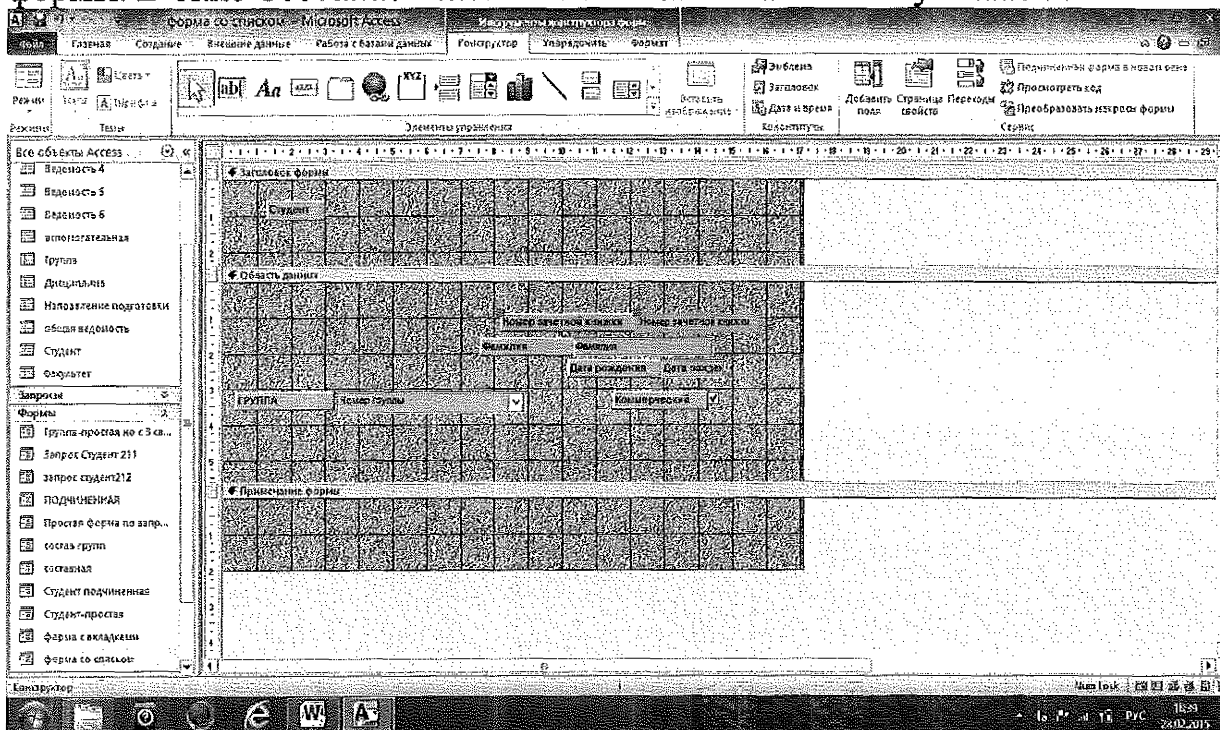


Рис. 32. Форма со списком в режиме конструктора

6. Разъединить контуры названия поля и самого поля, перетаскивая мышью левый верхний маркер одного из контуров. Заменить название поля в контуре надписи на *Группа*.

7. Нажать правой клавишей мыши внутри контура поля со списком и выбрать в контекстном меню пункт *Свойства*. Появится окно *Поле со списком* для настройки свойств поля. Выполнить настройку. Для этого:

- установить вкладку *Данные* в окне *Поле со списком*,

- в строке *Данные* раскрыть список полей и выбрать поле *Номер группы*;
- в строке *тип источника строк* выберите *таблица/запрос*;
- в строке *источник строк* выбрать запрос *Группа-факультет*;
- на вкладке *Макет* указать число столбцов – 3, заглавия столбцов – да, ширина списка – 7,2 см, ширина столбцов – 2; 2,2; 3.

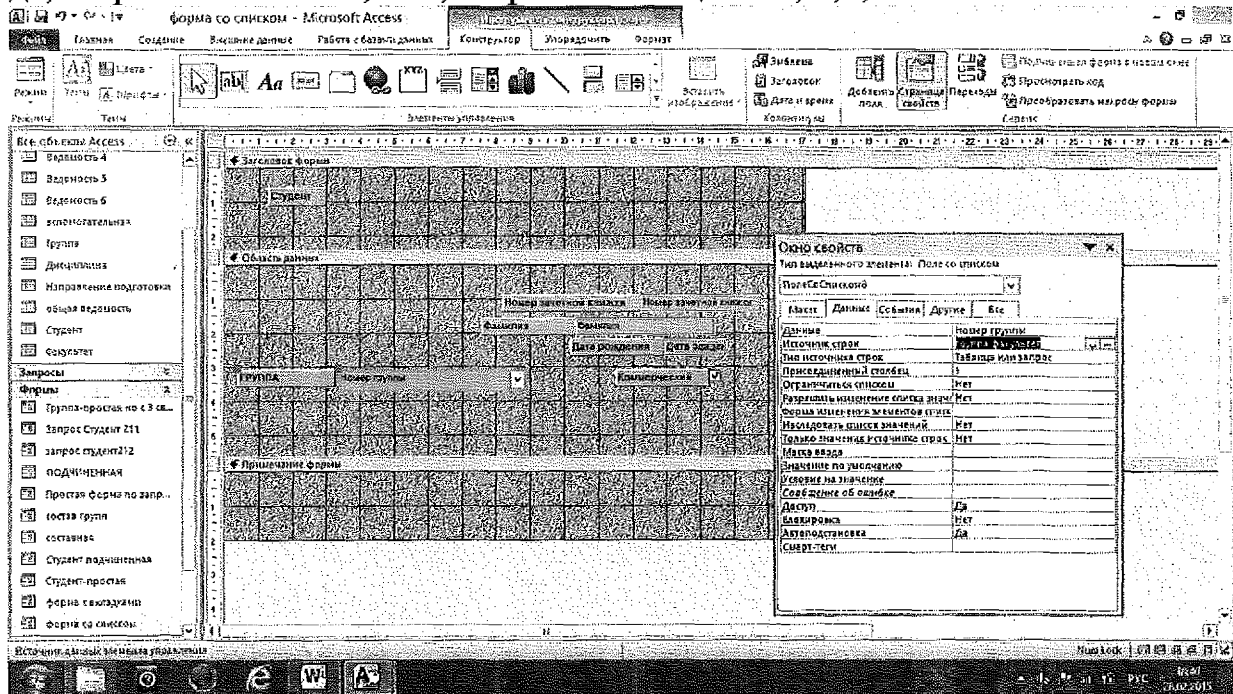


Рис. 33. Форма со списком в режиме конструктора

8. Просмотреть готовую форму, выбрав *режим формы*.
9. Сохранить форму под именем **Форма со списком**.

**Задание 5.** Создать **составную форму**. Необходимо создать составную форму для вывода информации о группе. Составная форма должна содержать главную форму и подчиненную форму.

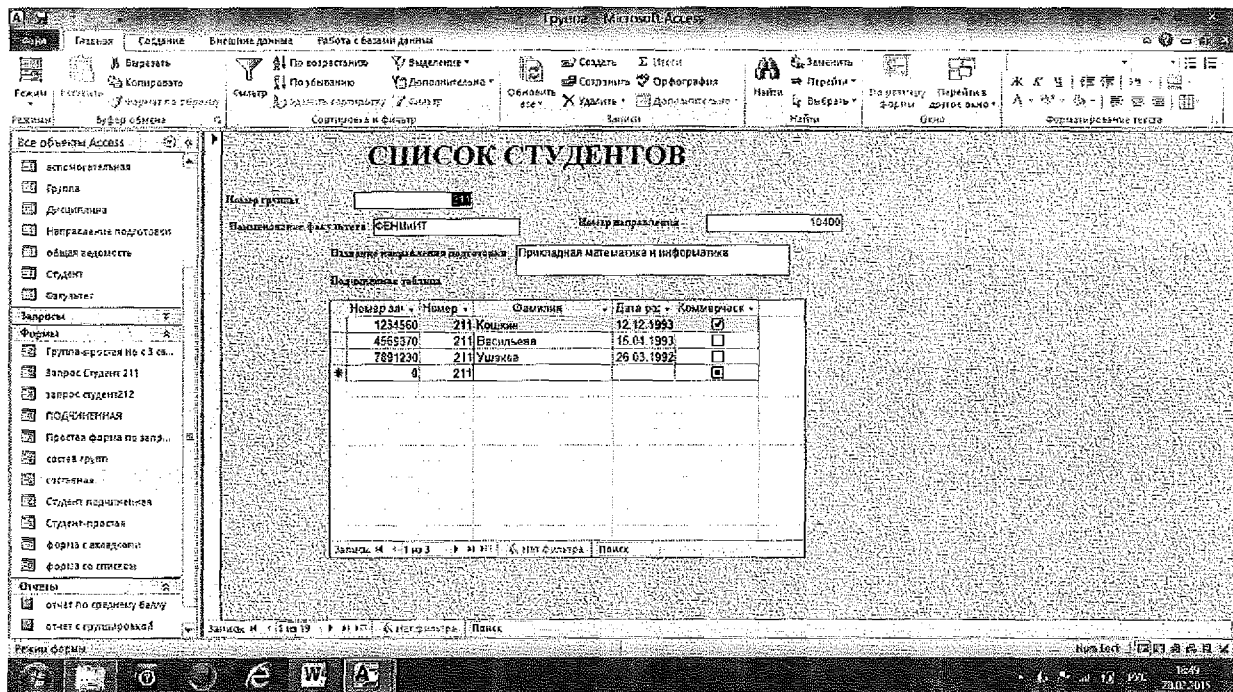


Рис. 34. Составная форма

### Технология.

1. Создать с помощью *Мастера форм* **табличную форму** с именем *Подчиненная* со списком студентов из таблицы *Студенты*.

**Подчиненная** форма должна иметь табличную форму и содержать список группы, в который необходимо включить поля:

- Фамилия, имя, отчество;
- Номер зачетной книжки;
- Дата рождения;
- Коммерческий.

2. Создать простой запрос на выборку *Группа–Направление подготовки–Факультет*, включив в него поля:

- Номер группы;
- Номер направления подготовки;
- Наименование направления;
- Наименование факультета.

3. Создать с помощью *Мастера форм* **главную форму** на основе запроса *Группа–Направление подготовки–Факультет*, включив в нее все поля запроса. Главную форму создать в один столбец стандартным стилем, используя тип представления данных *Группа*. Включить флажок *Изменить макета формы*.

4. В режиме конструктора:

- увеличить область данных;
- ввести элемент *Надпись*, располагающийся на ленте вверху, с текстом *Список студентов*;

- в область данных добавить *Подчиненная форма/Отчет*, для этого с ленты наверху на панели элементов управления выбрать *Подчиненная форма/Отчет* и активизировать команду *Использовать мастера*;
- в открывшемся окне *Мастер подчиненных форм* выбрать созданную ранее форму *Подчиненная / Далее / Готово*.

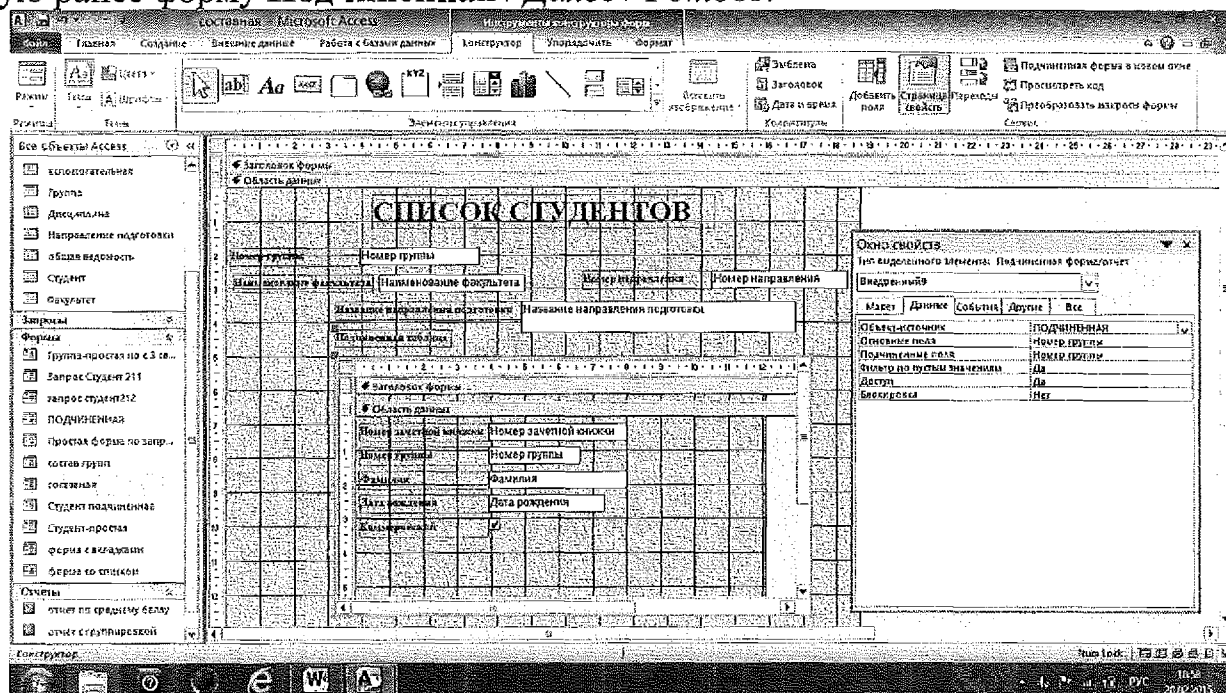


Рис. 35. Составная форма в режиме конструктора

5. Переходя из *режима формы* в *режим конструктора* и обратно, отрегулировать размеры и местоположение полей главной и подчиненной формы, изменить размеры и типы шрифтов и линий.

6. Сохранить форму как **Составная форма**.

**Задание 6.** Создать составную форму для просмотра состава групп.

Для создания такой формы необходимо подготовить *Запрос с итогом*, подчиненную форму *Студент подчиненная* в табличном виде.

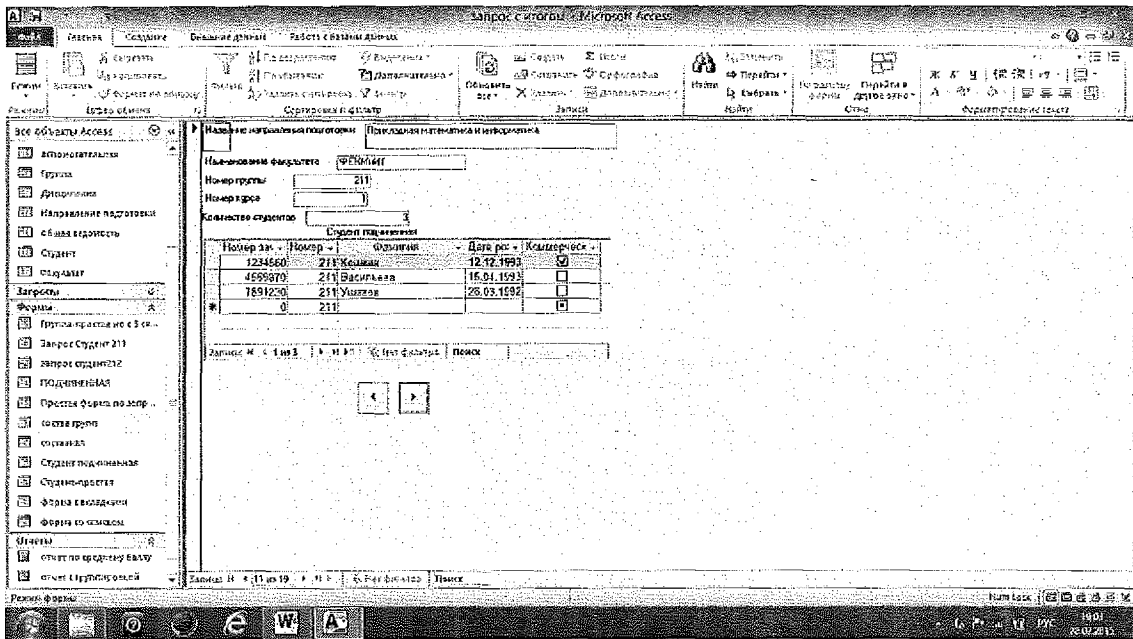


Рис. 36. Составная форма *Состав группы*

### Технология.

1. Создать **Запрос с итогом**. *Запрос с итогом по количеству студентов в группах* создать на основе таблиц *Факультет*, *Направление подготовки*, *Группа*, *Студенты*, включив в него поля *Наименование направления подготовки*, *Наименование факультета*, *Номер группы*, *Номер курса* и итоговое поле *Количество студентов в группе*. Выбрать команду на ленте *Итоги*, подключив групповые операции в запросе, в поле *Количество студентов в группе*: *Фамилия*, *имя*, *отчество* установить групповую операцию *Count* для подсчета студентов в группе.

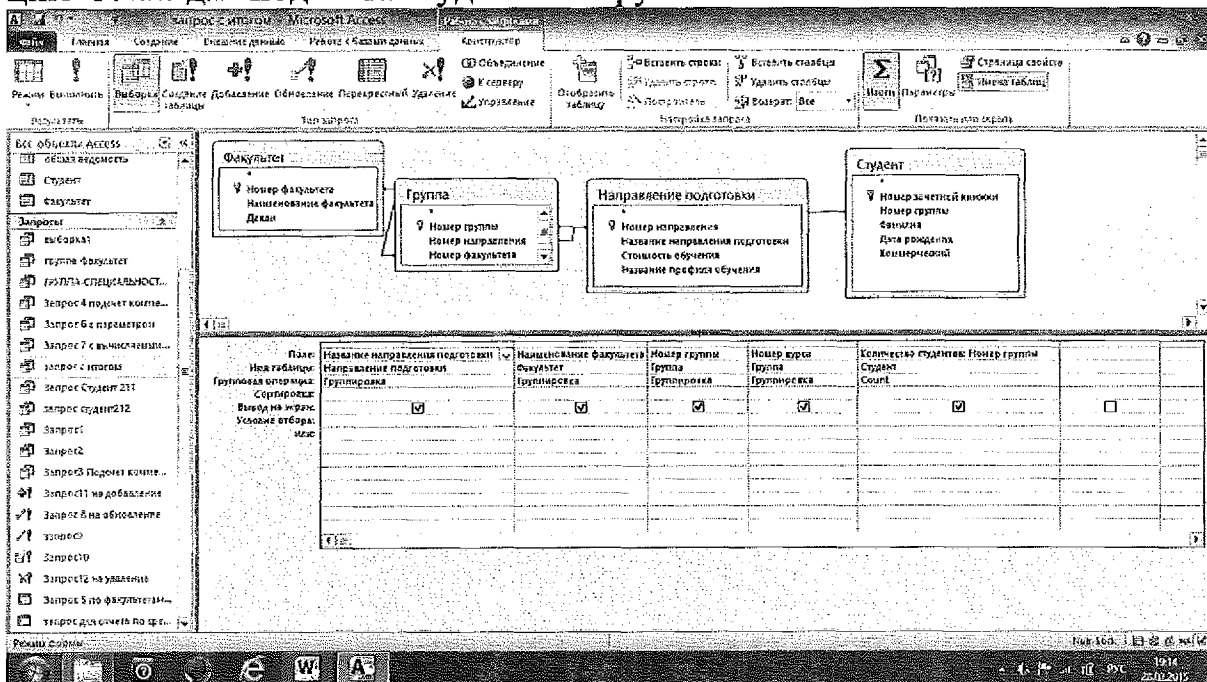


Рис. 37. Запрос с итогом в режиме конструктора



2. Создать подчиненную форму *Студент подчиненная* в табличном виде, содержащую список студентов, и подготовленную на основании таблицы *Студенты* (смотри предыдущее задание: создание подчиненной формы). Подчиненная форма *Студент подчиненная* в табличном виде, содержит список студентов, и подготовлена на основании таблицы *Студенты*.

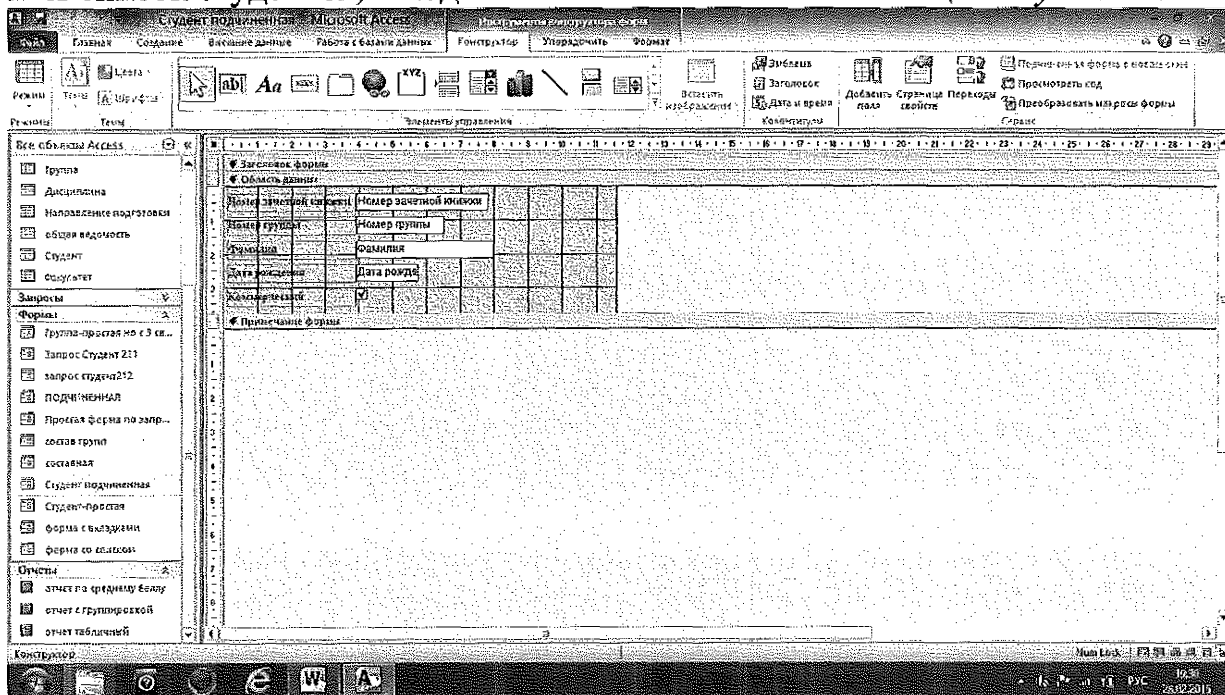


Рис. 38. Форма *Студент подчиненная* в режиме конструктора

3. Создать **главную форму**. Для этого выбрать создание формы с помощью *Мастера форм*.

4. Выбрать в качестве источника данных **Запрос с итогом**, созданный на основе таблиц *Факультет-Направление подготовки-Группа-Количество студентов*.

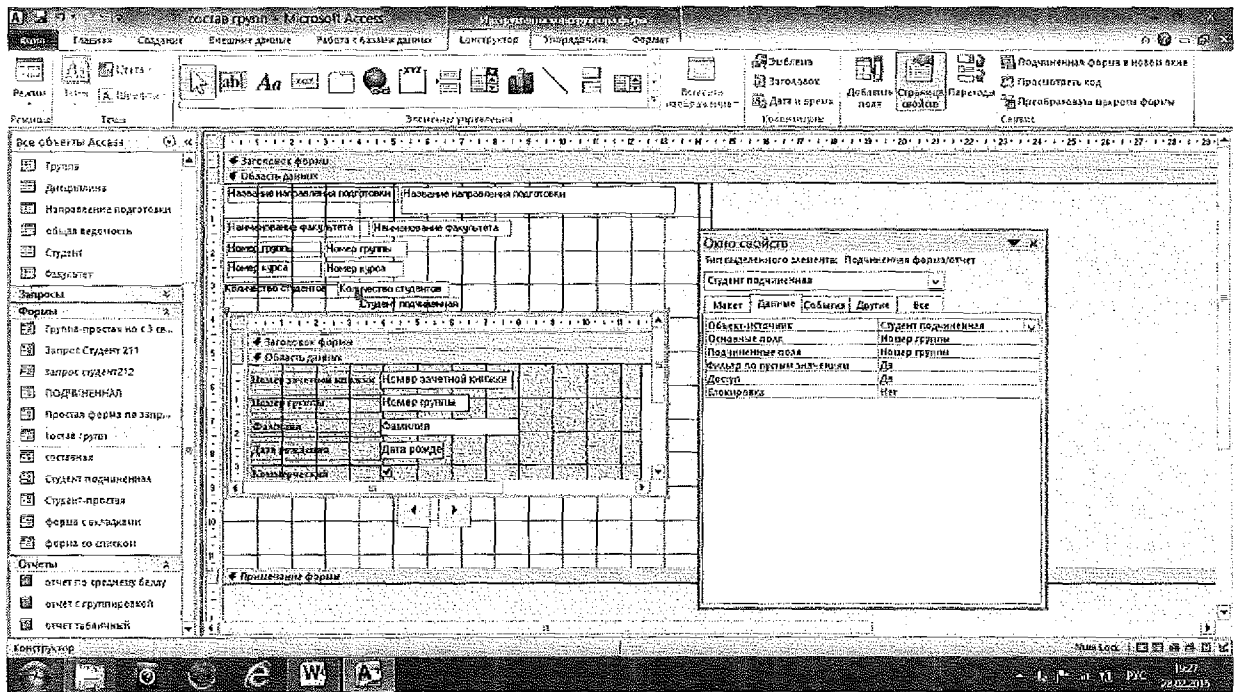


Рис. 39. Форма *Состав групп* в режиме конструктора

5. Ввести в главную форму, где отображается **Запрос с итогом**, заранее подготовленную починенную форму *Студент подчиненная*. Для этого нажать вверху на панели элементов управления по кнопке *Подчиненная форма/Отчет* и расположить ее на главной форме. После отпускания клавиши мыши появится окно с заголовком *Создание подчиненных форм и отчетов*. Включить в нем флажок *Формы* и выбрать в раскрывающемся списке имя формы *Студент подчиненная*. Нажать по кнопке *Готово*.

6. Установить связь главной и подчиненной форм по полю *Номер группы*. Для этого вызовите *Окно свойств* подчиненной формы и на вкладке *Данные* выбрать *Объект-источник Студент подчиненная*, нажать в строке *Подчиненные поля* и установить в ней значение *Номер группы*, в строке *Основные поля* и установить также значение *Номер группы*.

7. Дать название созданной форме **Состав групп**.

8. Создать в форме кнопку со стрелкой для перехода к следующей группе студентов. Для этого *Использовать мастера* в инструментах конструктора форм, нажав по соответствующей кнопке этих инструментов. Выбрать вверху на панели управления элемент *Кнопка* и очертить контур будущей кнопки при нажатой левой клавише мыши. После отпускания клавиши мыши появится окно *Создание кнопок*.

9. В окне *Создание кнопок* выбрать категорию *Переходы по записям*, а в нем – действие *Следующая запись*. Нажать по кнопке *Далее*. В следующем окне включить флажок *Показать все рисунки* и выбрать рисунок *Стрелка влево*. Нажать по кнопке *Далее* и *Готово*.

10. Аналогичным образом создать кнопку с рисунком *Стрелка вправо* для перехода к предыдущей записи и кнопку выхода из формы.

**Задание 7.** Создать форму с вкладками для просмотра списка студентов двух групп. На каждой вкладке должен располагаться список студентов только одной группы. На ярлычках вкладок должны отображаться номера групп.

**Технология.**

1. С помощью *Мастера запросов* создать два простых запроса, например, *Запрос-Студент 212* и *Запрос-Студент 211*, для выборки из таблицы *Студенты* списка студентов двух групп (212 и 211). Включить в запросы все поля таблицы *Студенты*. В режиме конструктора ввести в запросы условия отбора записей в поле номер группы, соответственно 212 или 211.

2. С помощью *Мастера форм* создать две табличные формы, например *Форма Студент 212* и *Форма Студент 211* на основании запросов *Запрос Студент 212* и *Запрос Студент 211* с использованием стандартного стиля.

3. Создать с помощью *Конструктора* новую форму (главную), не используя в качестве источника таблицу или запрос. В окне конструктора вверху на ленте выбрать элемент *Набор вкладок* (или *вставить вкладку*) и очертить мышкой прямоугольный контур в области данных.

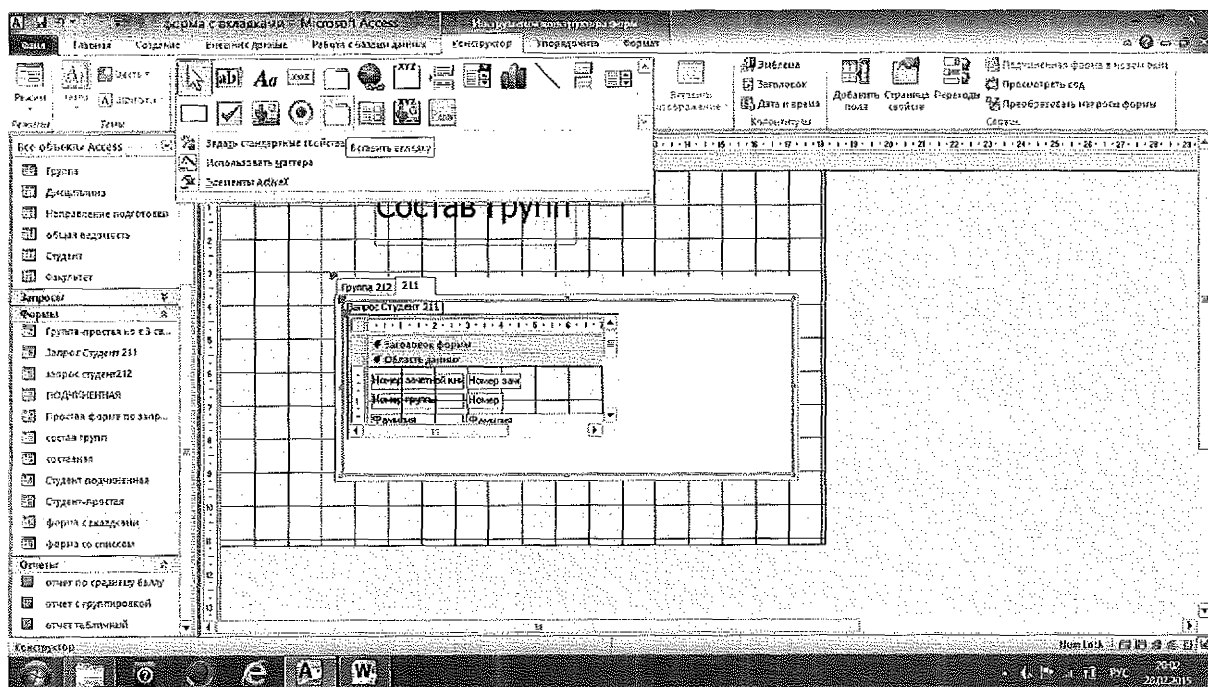


Рис. 40. Форма с вкладками в режиме конструктора

4. Заменить названия ярлычков вкладок. Для замены имени ярлычка необходимо нажать по ярлычку правой клавишей мыши и выбрать в контекстном меню пункт *Свойства*, а затем ввести в *Окне свойств* элемента новое имя на вкладке *Другие*.

5. Выделить первую вкладку и разместить на ней подчиненную форму *Форма Студент 212*. Удалить связанную с подчиненной формой подпись формы.

6. Выделить вторую вкладку и разместить на ней подчиненную форму *Форма Студент 211*. Удалить связанную с подчиненной формой подпись формы.

7. Просмотреть форму в режиме формы. Отрегулировать размеры подчиненных форм. Внешний вид формы, которая должна быть получена, представлен на рис. 41.

8. Сохранить форму с именем **Форма с вкладками**.

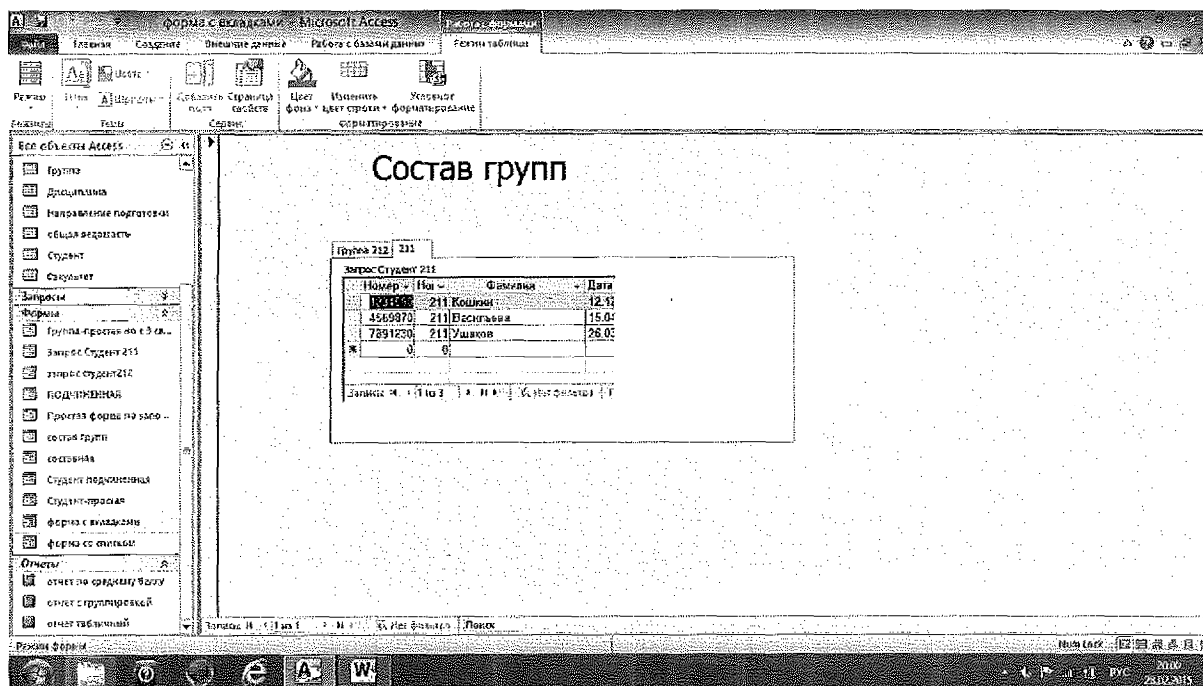


Рис. 41. Форма с вкладками

## 2.8 Создание отчетов

**Цель работы:** конструирование и использование отчетов.

Отчеты предназначены для вывода информации из базы данных на принтер. Перед выводом на принтер отчет можно просмотреть на экране. Отчет строится на основании таблиц и запросов.

Основные виды отчетов:

- одноколонный (простой) отчет;

- многоколонный отчет;
- табличный отчет;
- отчет с группировкой данных и подведением итогов;
- отчет по связанным таблицам;
- связанный отчет, т.е. отчет, содержащий другой (подчиненный отчет);
- отчет слиянием с Word (составной документ);
- перекрестный отчет.

*Основные разделы отчета:*

- заголовок отчета (начало отчета);
- верхний колонтитул (печатается в начале каждой страницы);
- область заголовка группы (отображается перед первой записью каждой группы);
- область данных (основная часть отчета);
- область примечания группы (отображается после области данных последней записи каждой группы);
- нижний колонтитул (печатается в конце каждой страницы);
- область примечаний (печатается в конце отчета).

В режиме конструктора доступны кнопки панели инструментов: сортировка и группировка; список полей; свойства.

**Окно сортировки и группировки** позволяет определить условия сортировки и группировки данных в отчете. В ячейках верхней половины окна указываются поля, по которым выполняется сортировка, и порядок сортировки. Ячейки нижней половины окна позволяют задать условия группировки. Допускается выполнение сортировки записей без их группировки; однако, условия группировки можно задать только для сортируемых полей или выражений.

**Столбец «Поле/выражение».** В ячейках столбца выбирают поле, по которому проводится сортировка, или вводят выражение. При сортировке по нескольким полям следует расположить поля сверху вниз в порядке проведения сортировки.

**Столбец «Порядок сортировки».** В ячейках выбирают порядок сортировки «По возрастанию» или «По убыванию» для поля или выражения в этой строке.

**Свойства группы.** В ячейках задаются параметры группировки для выбранного поля или выражения. В поле в правом нижнем углу окна вводится описание выбранного столбца или параметра группировки.

Для построения многоколонного отчета выполнить в режиме «*Конструктор*»:

- команду *Файл / Настройка печати*;
- нажать кнопку *Дополнительно*;
- в поле ввода *По горизонтали* указать количество колонок.

**Задание 1.** Создать **табличный отчет**, модифицировав запрос на выборку, подготовленный в задании 1 лабораторной работы 2, которому было присвоено имя *Запрос 1*. Отчет должен включать следующие столбцы:

- Наименование факультета,
- Номер группы,
- Наименование направления подготовки,
- Номер зачетной книжки,
- Фамилия, имя, отчество.

**Технология.**

1. Выбрать в окне базы данных вкладку *Создание* на ней *Мастер отчетов* и в качестве источника данных – *Запрос 1*.

2. В окне *Создание отчета* выполнить следующие шаги:

– выбрать все поля запроса, в заданной по условию последовательности;

– выбрать тип представления данных *по факультету*;

– добавить уровень группировки *по номеру направления подготовки*;

– выбрать порядок сортировки по полю *Фамилия, имя, отчество*;

– выбрать макет *ступенчатый*;

– выбрать тип заголовка *стандартный*;

– присвоить имя отчету **Отчет табличный**;

– нажать кнопку **Готово**. Просмотреть отчет.

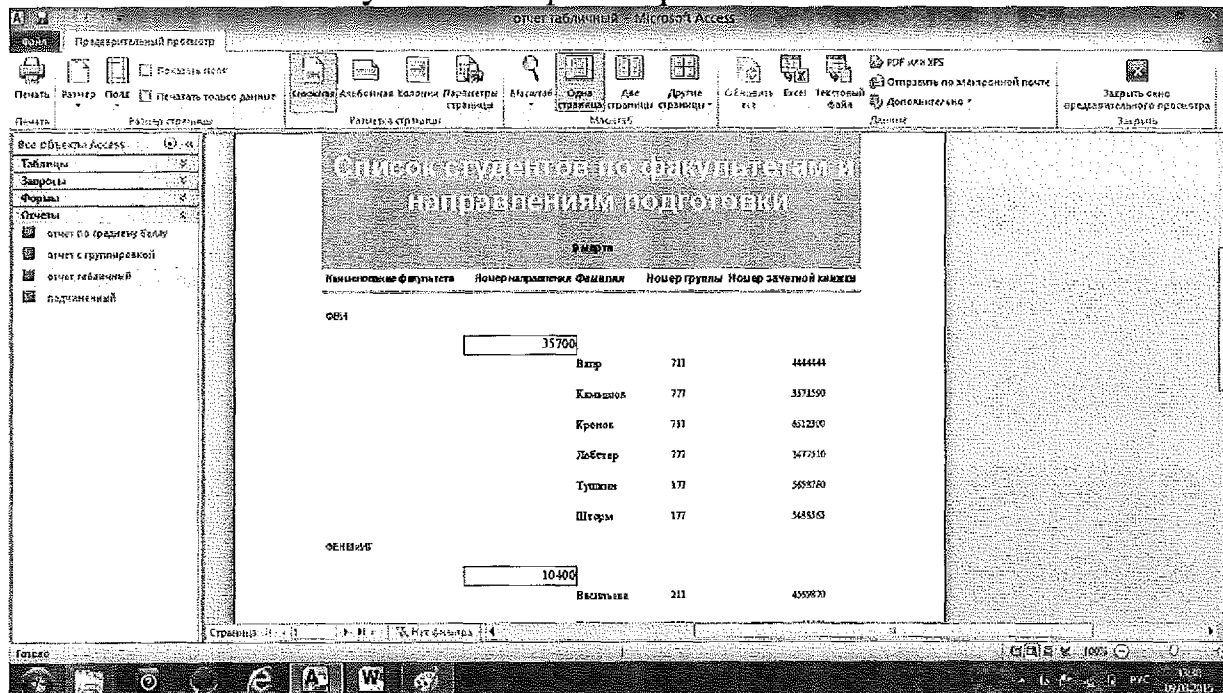


Рис. 42. Отчет табличный

4. Изменить заголовок отчета, введя текст *Список студентов по факультетам и направлениям подготовки*.

5. Ввести надпись в нижний колонтитул «Ведомость составил: ».
6. Закрыть и сохранить отчет.

**Задание 2.** Создать отчет с группировкой данных и подведением итогов, используя запрос на выборку, подготовленный в задании № 7 лабораторной работы № 2 с именем **Запрос 7 с вычисляемым полем**. В отчете необходимо подсчитать общую суммарную плату за обучение коммерческих студентов и в том числе по каждому направлению подготовки и по каждой группе.

Отчет должен содержать поля:

- Номер направления подготовки;
- Номер группы;
- Фамилия, имя, отчество;
- Номер зачетной книжки.

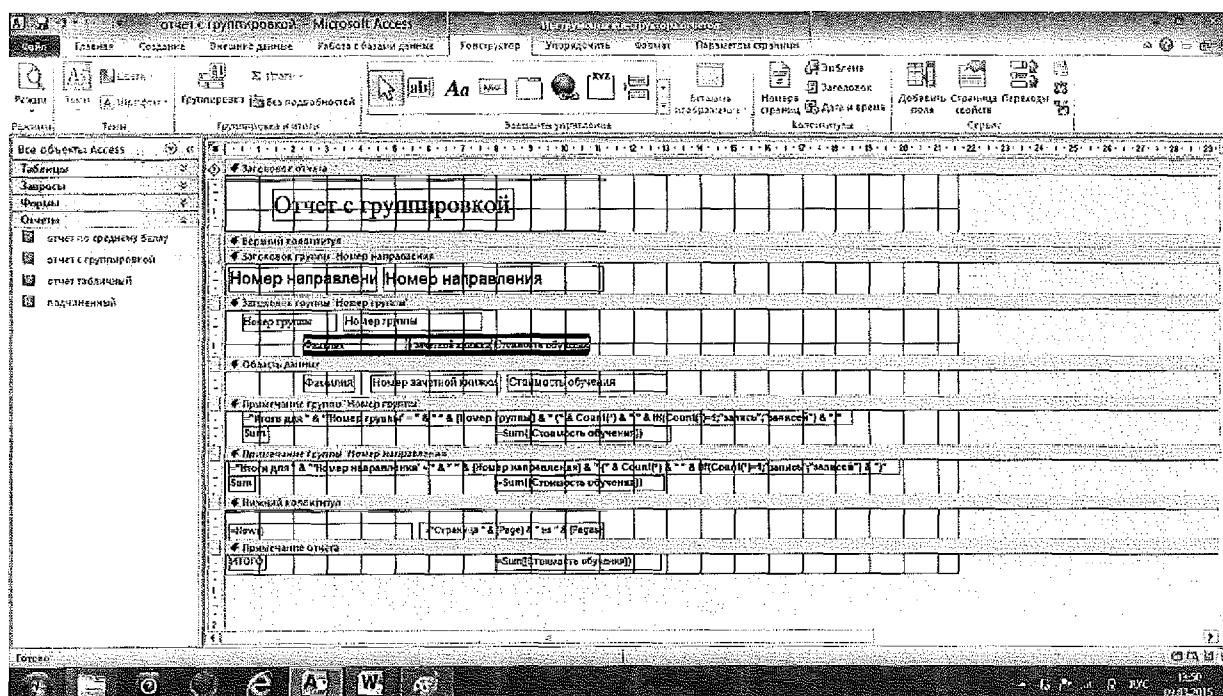


Рис. 43. Отчет с группировкой в режиме конструктора

### Технология.

1. Создать с помощью *Мастера отчетов* новый отчет на основании указанного выше запроса (Запрос 7 с вычисляемым полем). Включить в отчет все поля запроса в заданном порядке.
2. В окне создания отчета выполнить следующие шаги:
  - выбрать тип представления данных – по таблице *Студенты*;
  - добавить уровни группировки *по направлению подготовки и группе*;

- настроить режим сортировки в *алфавитном порядке фамилий* и, нажав по кнопке *Итоги*, включить флажки *Sum* и *показать данные и итоги*;
  - выбрать макет отчета *Структура*;
  - выбрать один из предложенных *стиль* заголовка.
3. Нажать кнопку *Готово* и просмотреть отчет.
  4. Закрыть отчет и сохранить его под именем **Отчет с группировкой**.
  5. Выполнить редактирование отчета, используя режим конструктора. Отредактировать заголовок отчета, ширину столбцов. Изменить шрифты отдельных элементов отчета. Заменить слово *Sum* в итоговых строках на слово «*Всего*». Изменить ширину полей с итогами.

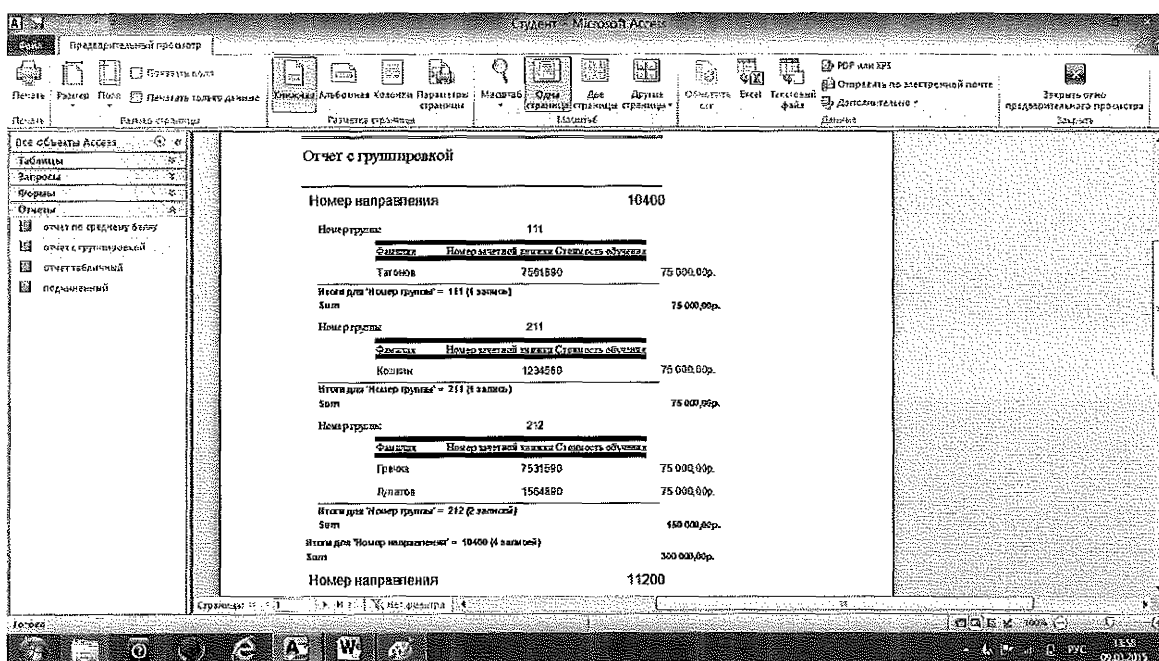


Рис. 44. Откорректированный отчет с группировкой

**Задание 3.** Создать **перекрестный отчет**, используя таблицу **Общая ведомость**, созданную в лабораторной работе № 2. Отчет должен иметь следующие столбцы:

- Номер группы;
- Фамилия, имя, отчество;
- Номер зачетной книжки;
- Наименование дисциплины 1 (указать ту, которая упоминается в базе данных);
- Наименование дисциплины 2 (указать ту, которая упоминается в базе данных);
- Средний балл.



В столбцах с наименованиями дисциплин должны отображаться оценки, полученные студентами на экзамене. В столбце *Средний балл* для каждой строки должно производиться вычисление среднего балла каждого студента по результатам сдачи 2-х экзаменов.

Вычислить в отчете средний балл по группе студентов и по университету в целом.

**Технология.**

1. Предварительно необходимо создать запрос. С помощью *Мастера запросов* создать перекрестный запрос по таблице *Общая ведомость*, в котором должны иметься оценки для всех студентов 3-х групп по двум дисциплинам.

Для этого:

- выбрать поля *Номер группы*, *Фамилия*, *имя*, *отчество*, *Номер зачетной книжки* в качестве заголовков строк;
- выбрать поле *Наименование дисциплины* в качестве заголовка столбцов;
- снять флажок *Да* вычисления итоговых значений для каждой строки;
- выбрать для вычислений поле *Оценка* и функцию вычисления *First* (Первый);
- сохранить запрос, присвоив ему имя **Запрос для отчета по среднему баллу**.

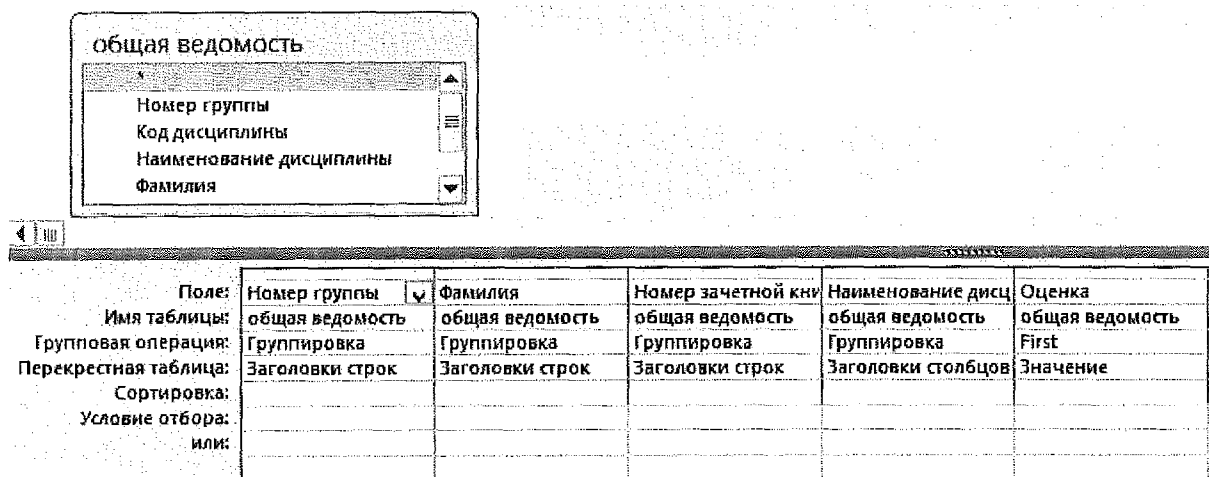


Рис. 45. Запрос для отчета по среднему баллу в режиме конструктора

2. С помощью *Мастера отчетов* создать отчет по запросу **Запрос для отчета по среднему баллу**.

3. Выбрать все поля запроса.
4. Добавить уровень группировки по номеру группы.
5. Выбрать порядок сортировки по фамилиям студентов.
6. Выбрать стандартный вид отчета.

7. Присвоить отчету имя **Отчет по среднему баллу**. Просмотреть отчет.

8. Перейти в режим конструктора и отрегулировать ширину столбцов и их местоположение. Выровнять оценки по центру столбца.

9. С помощью инструментальной панели элементов управления ввести в правую часть области верхнего колонтитула надпись *Средний балл*.

10. С помощью инструментальной панели элементов управления ввести в область данных *поле* для вычисления среднего балла. Нажать правой кнопкой мыши внутри созданного поля и вызвать окно Свойства поля.

11. Установить вкладку *Данные*. Для вызова построителя выражений на вкладке *Данные* в строке данные щелкнуть по кнопке «...».

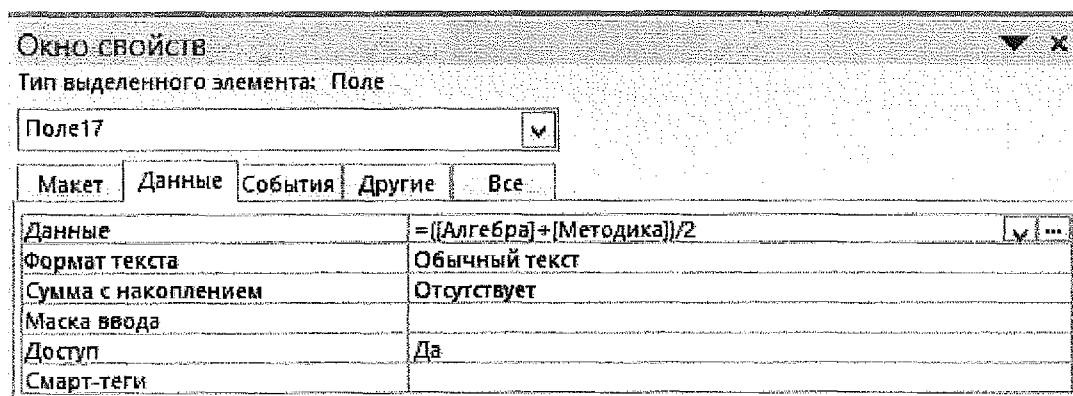


Рис. 46. Окно свойств поля

12. Используя инструменты построителя, ввести в поле формулу расчета среднего балла:

$$=([\text{наименование дисциплины1}] + [\text{наименование дисциплины2}]) / 2$$

*Примечание.* Вместо **наименования дисциплины 1** внесите название дисциплины из созданной базы данных.

Наименования дисциплин вставляются в формулы двойным щелчком левой клавиши мыши по соответствующей строке в подокне построителя выражений *Отчет*.

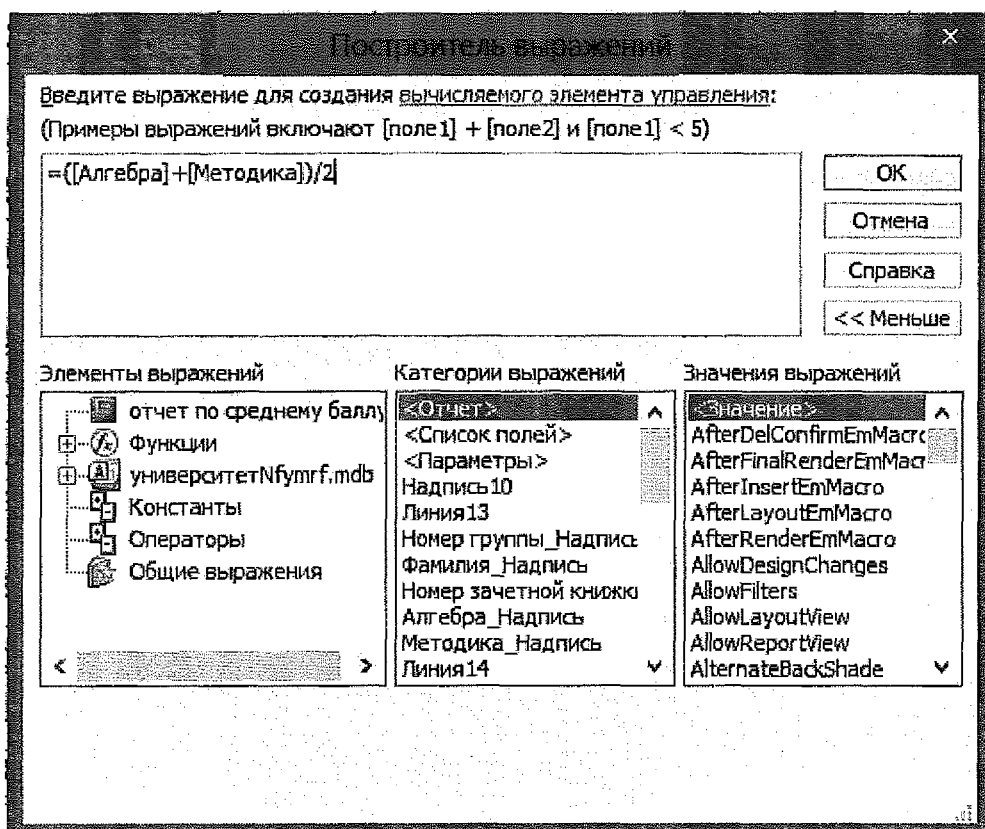


Рис. 47. Окно построителя выражений для вычисления *Среднего балла*

13. Просмотреть и отредактировать отчет. Отрегулировать формат, местоположение и выравнивание вычисленного значения. Установить для поля *Средний балл* в окне свойств вкладку *Макет* и выбрать формат поля фиксированный с 2-мя дробными знаками.

14. Выполнить **условное форматирование** для поля *Средний балл*. Выделить красным цветом значения среднего балла, большие или равные 4 и синим цветом – меньшие или равные 3. Для этого в режиме конструктора выделить поле *Средний балл*, вызвать правой кнопкой мыши контекстное меню и выбрать команду *Условное форматирование*. В окне *Условное форматирование* создать правила: Условие 1 – Значение поля, операция сравнения – больше или равно, значение для сравнения – 4. В строке задания формата установить цвет текста – красный. Повторить указанные действия для Условия 2, установив синий цвет для всех значений поля, меньших или равных 3.

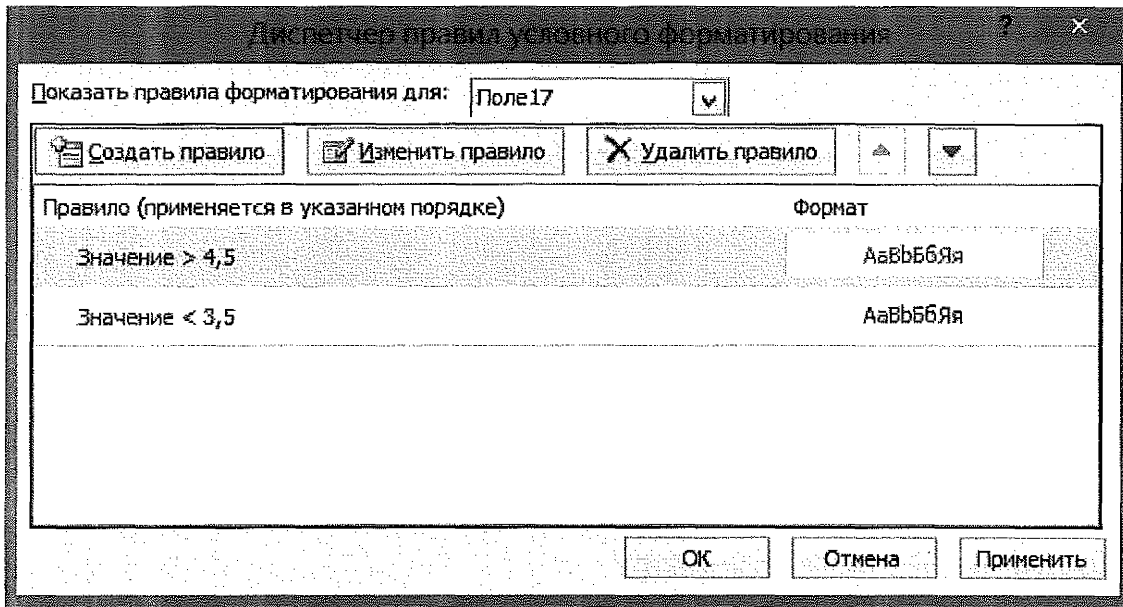


Рис. 48. Диалоговое окно команды Условное форматирование

15. Для подсчета среднего балла по группе добавить в окне Конструктора раздел *Примечание группы*. Для этого вызвать окно настройки *Сортировка и группировка*, нажав правой кнопкой мыши и вызвав контекстное меню. В окне *Сортировка и группировка* включить отображение области *Примечаний группы* (выбрать с разделом *примечания*).



Рис. 49. Окно команды Сортировка и группировка

16. В области *Примечаний группы* в столбце, соответствующем среднему баллу студента, создать два поля. Для этого в область *Примечаний группы* вставить поле и ввести в него формулу:

= «Средний балл группы» & [Номер группы]

Для второго поля ввести формулу вычислений:

=Sum ([наимен. дисц.1] + [наимен. дисц.2]) / 2 / Count([Фамилия])

*Примечание.*

Для ввода названий функции групповых операций Sum и Count дважды щелкнуть по строке *Функции* в левом подокне *Построителя выражений*, а затем двойным щелчком раскрыть список встроенных функций. В среднем подокне выбрать категорию *Статистические функции* и затем в правом подокне нужную функцию.

17. Удалить контур с надписью вычисляемого поля. Отрегулировать формат, местоположение.

18. Подсчитать в отчете средний балл по всем группам. Для этого в область *Примечание отчета* вставить надпись *Средний балл по университету* и добавить поле

=Sum ([наимен. дисц.1] + [наимен. дисц.2]) / 2 / Count([Фамилия])

<b>Расчет среднего балла группы</b>					
☛ Верхний колонтитул					
Номер группы	Фамилия	Номер зачетной книжки	Алгебра	Методика	Средний балл
☛ Заголовок группы: Номер группы					
☛ Область данных					
Номер группы	Фамилия	Номер зачетной книжки	Алгебра	Методика	=(Алгебра)+[Методика]*2
☛ Примечание группы: Номер группы					
="Средний балл группы" & [Номер группы]		=Sum([Алгебра]+[Методика])/2/Count([Фамилия])			
☛ Нижний колонтитул					
=Nov()	& [Pages]				
☛ Примечание отчета					
<i>Средний балл по университету</i>		=Sum([Алгебра]+[Методика])/2/Count([Фамилия])			

Рис. 50. Отчет по среднему баллу в режиме конструктора

19. Присвоить отчету имя **Расчет среднего балла групп**. Ввести в заголовок отчета дату формирования отчета. Для этого вставьте под строкой с названием отчета поле и введите в него с помощью *Построителя выражений* функцию **=DATE()**.

20. Для того чтобы сделать отчет более компактным, перенести номер группы из области заголовка группы в область данных. Для этого отключить область заголовка группы, вызвав окно *Сортировка и группировка*. Вставить поле *Номер группы* в соответствующий столбец области данных. Настроить свойства поля. Для того, чтобы номер поля не повторялся в каждой строке отчета установить на вкладке *Макет* свойство *Не выводить повторы* в состояние *Да*. Примерный вид отчета представлен на рис. 51.

Номер группы	Фамилия	Номер зачетной книжки	Алгебра	Математика	Средний балл
111	Машонев	1439630	4	5	4,50
111	Новик	2530630	4	4	4,00
111	Тетюха	7591809	4	3	3,50
Средний балл группы 111				4	
115	Ваш	7899870	4	5	4,50
115	Швал	9877890	4	5	4,50
Средний балл группы 115				4,5	
116	Зрех	1477419	3	4	3,50
116	Швал	6565409	3	5	4,00

Рис. 51. Отчет по среднему баллу в режиме просмотра

### Создание связанного отчета

Связанный отчет состоит из главного и одного или нескольких подчиненных отчетов. Подчиненным отчетом называют отчет, созданный внутри другого отчета. Подчиненные отчеты позволяют решать следующие задачи:

- вывести сводные данные или конкретные записи, связанные с содержанием главного отчета;
- объединить несколько независимых отчетов в главном отчете.

Для создания подчиненного отчета следует создать отчет, который будет использоваться как подчиненный, и сохранить его. Открыть главный

отчет в режиме конструктора. Перейти в окно базы данных, например, нажатием клавиши F11. Переместить с помощью мыши значок созданного подчиненного отчета из окна базы данных в соответствующую позицию в главном отчете. Например, для печати подчиненного отчета перед каждой группой в главном отчете следует поместить подчиненный отчет в заголовок группы. Подчиненный отчет выводится как отдельный элемент управления.

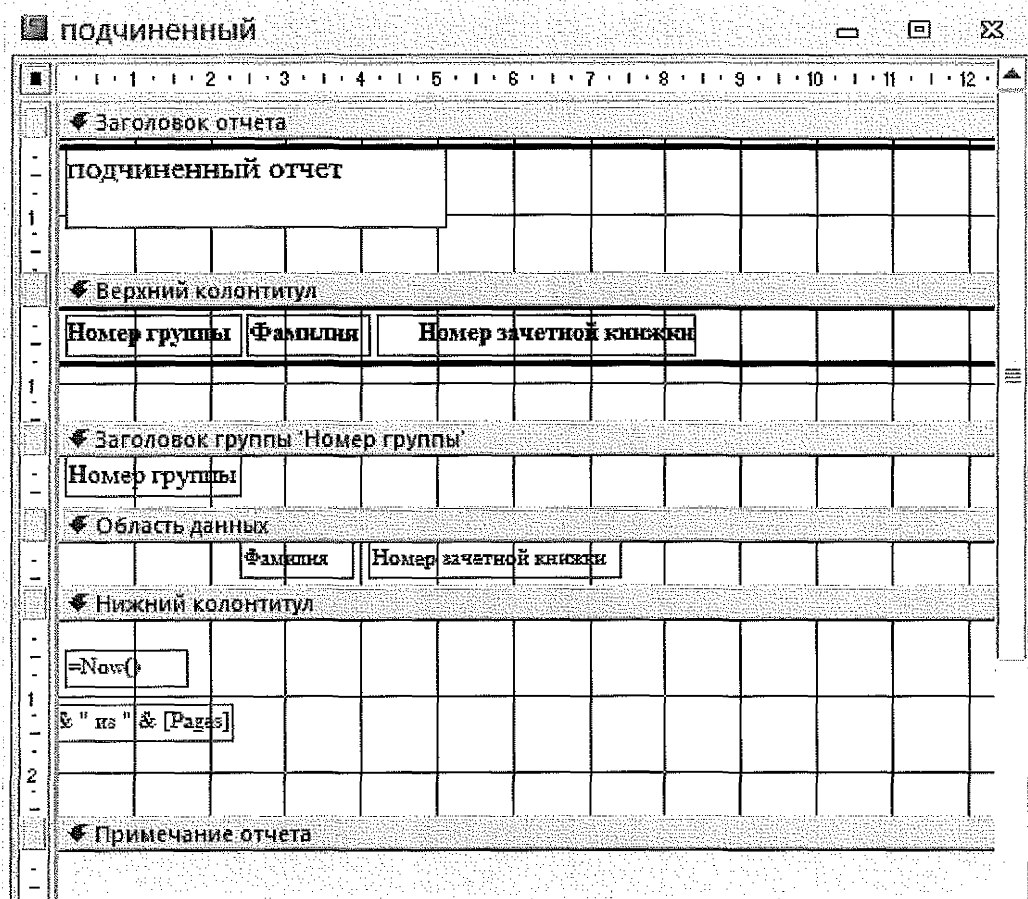


Рис. 52. Подчиненный отчет в режиме конструктора

Если в подчиненном отчете должны выводиться данные, связанные с главным отчетом, установить связь элемента управления подчиненного отчета с главным отчетом. Для связывания подчиненного отчета с главным отчетом открыть окно свойств подчиненного отчета и задать связь свойств *Подчиненного поля* и *Основного поля* по номеру группы. По возможности, значения этих свойств задаются автоматически средствами Microsoft Access.

Например, подчиненный отчет – это отчет, содержащий сведения о студентах, сгруппированные по номерам групп. Главный отчет – это отчет, содержащий сведения о группах (номер, факультет, направление подготовки).

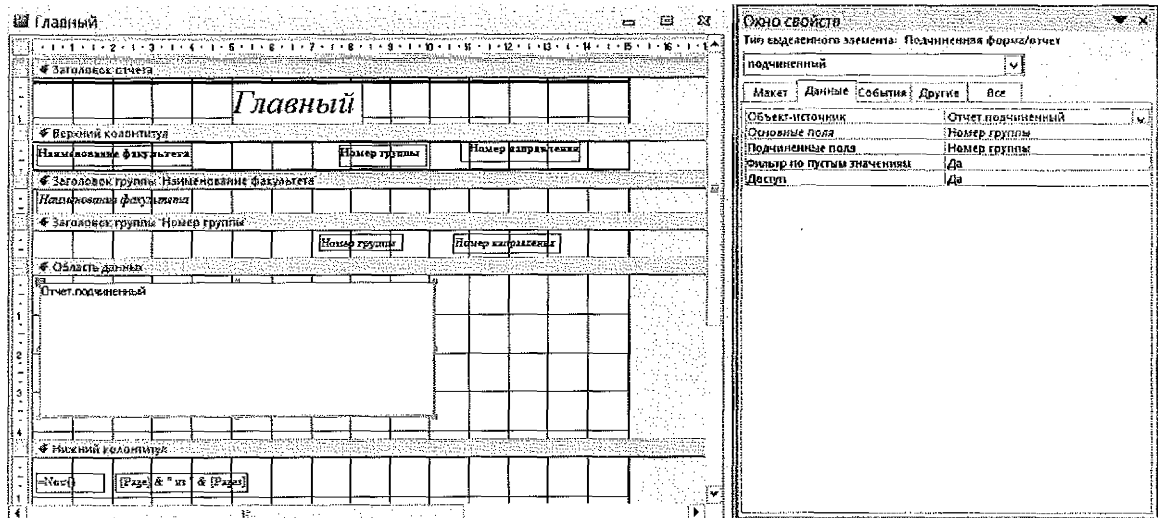


Рис. 53. Связь полей главного и подчинённого отчетов

Связь осуществляется через поле *Номер группы*. Главный отчет создается мастером на базе запроса *Факультет-группа* как простой отчет. Подчиненный отчет также создается мастером на базе таблицы *Студенты* как отчет с группировкой.

*Примечание.* Допускается вставка в отчет подчиненной формы. Главный отчет может содержать несколько подчиненных отчетов или подчиненных форм. В главном отчете допускается существование двух уровней подчиненных отчетов. Например, отчет может содержать подчиненный отчет, который в свою очередь содержит подчиненный

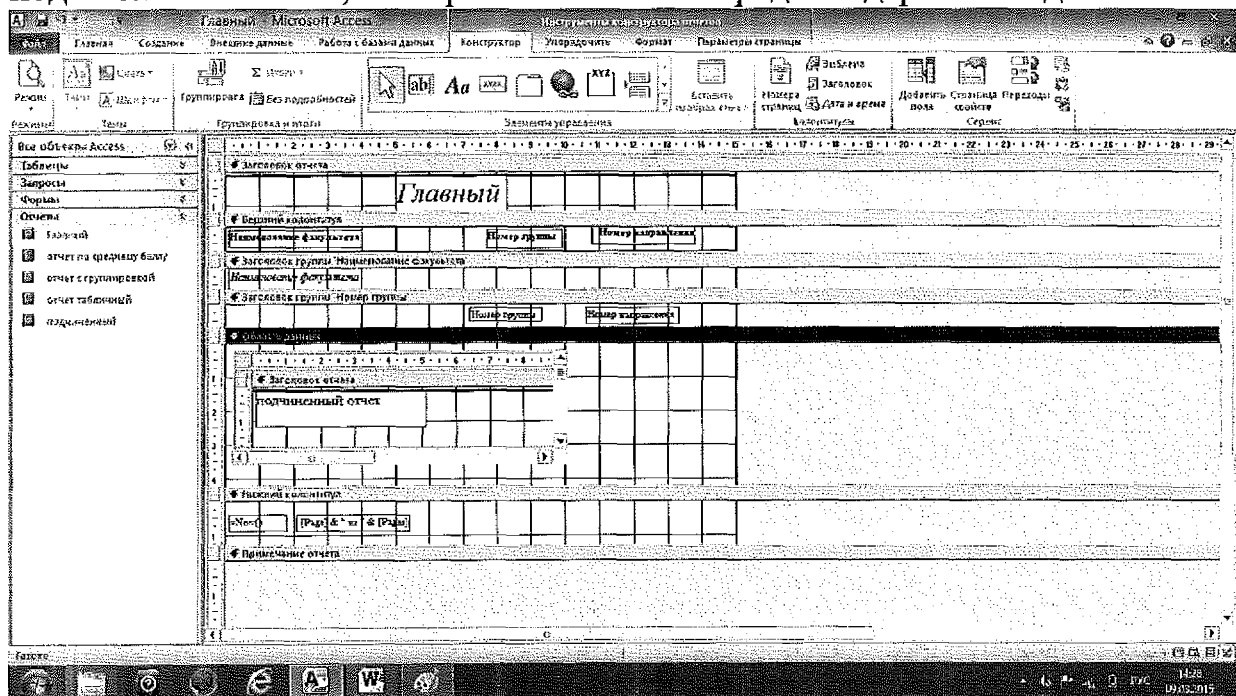


Рис. 54. Главный отчет в режиме конструктора



**Задание 4.** Создать **связанный отчет**, состоящий из главного и подчиненного отчетов. В качестве главного отчета создать отчет на базе запроса по связанным таблицам *Группа-Факультет*. В качестве подчиненного отчета использовать табличный отчет по таблице *Группа*.

**Технология.**

1. Создать с помощью *Мастера отчетов* ступенчатый **Подчиненный отчет** по таблице *Студенты*, сгруппированный по *номерам групп* и упорядоченный по *фамилиям* студентов, включив в него по:

- Номер группы;
- Фамилия, имя, отчество;
- Номер зачетной книжки.

Сохранить отчет, присвоив ему имя **Подчиненный**. Отредактировать отчет в режиме конструктора. Уменьшить ширину столбцов и длину линий обрамления заголовка.

3. Создать с помощью *Мастера отчетов* ступенчатый **главный отчет** по запросу *Группа-Факультет*, упорядоченный по *факультетам* и *номерам групп*, включив в него поля:

- Наименование факультета;
- Номер группы;
- Номер направления подготовки.

Сохранить отчет, присвоив ему имя **Главный**. Отредактировать в режиме конструктора. Уменьшить ширину столбцов и длину линий обрамления заголовка.

5. Включить режим конструктора для **Главного отчета**. Не закрывая окна конструктора отчета *Главный*, перетащить мышью из окна базы данных (вкладка: отчеты) значок отчета **Подчиненный** в область данных главного отчета.

6. Нажать правой кнопкой мыши по полю *Подчиненный* и вызвать окно свойств поля. На вкладке *Данные* обеспечьте связь между подчиненным и основным отчетом по полю *Номер группы*.

7. Просмотреть получившийся **связанный отчет**. Вернуться в режим конструктора и отредактировать отчет.

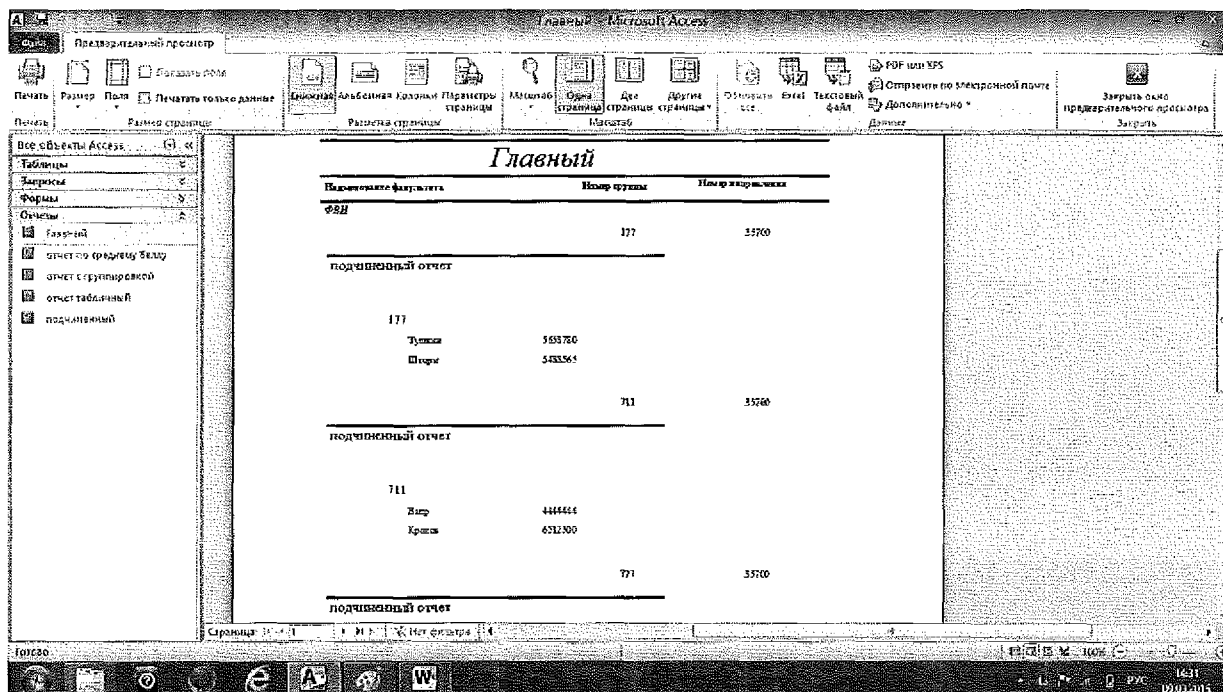


Рис. 55. Связанный отчет

## 2.9 Создание макросов

**Цель работы:** Освоение некоторых возможностей автоматизации управления базой данных. Создание и применение макросов. Создание пользовательского меню.

### Краткие сведения

Макросы удобно использовать для упрощения работы с БД. С помощью макросов можно выполнять определенные действия, например открытие какой либо формы, таблицы или отчета, выход из Access и др.

Макросу присваивается заданное пользователем имя. Если макрос имеет стандартное имя Autoexec, то такой макрос выполняется автоматически сразу после загрузки Access. Макросы состоят из макрокоманд. Макрокоманды записываются в строки первого столбца таблицы. Второй столбец служит для записи примечаний.

Если нажать мышью по ячейке первого столбца макроса, то раскроется список возможных макрокоманд, из которого следует выбрать нужную макрокоманду.

**Задание 1.** Создать форму-меню с кнопками для выполнения ранее составленных запросов, вывода форм и отчетов, а также выхода из Access. Форма-меню должна появляться автоматически сразу после загрузки базы данных. В форме должна быть также кнопка для закрытия базы данных и выхода из Access.

### **Технология.**

1. Создать с помощью *Конструктора* новую форму *Меню*. Появится форма, содержащая только одну область данных. Добавить область заголовка формы в виде надписи. Сократить до нуля область примечаний.

2. Ввести в область заголовка формы надпись *База данных ФГБОУ ВО ТОГУ*. Подобрать оптимально местоположение надписи, ее размер и шрифт. Закрыть и сохранить форму под именем *МЕНЮ*.

3. Создать *Макрос* для автозагрузки формы-меню. Для этого на вкладке *Создание* выбрать команду *Макросы*.

– в окне макроса в столбце *Макрокоманды* развернуть список и выбрать макрокоманду *Открыть форму*.

– в строку *Имя формы* ввести текст: *МЕНЮ*.

– на ленте найти команду *Выполнить макрос*, запустить его и сохранить с именем *Autoexec*.

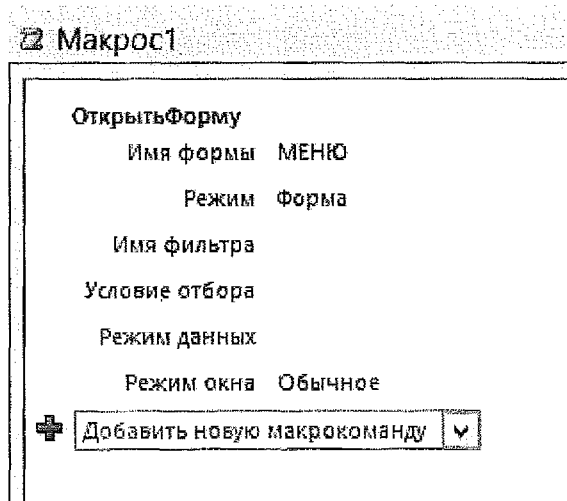


Рис. 56. Вид окна конструктора макросов

4. Создать макрос для выхода из Access. Для этого:

– создать макрос из макрокоманды *Выйти из Access*, выбрав ее из списка;

– выбрать в строке *Параметры* значение *Сохранить все*.

– на ленте найти команду *Выполнить макрос*, запустить его и сохранить с именем *Выход*.

5. Создать макрос для выполнения *Запроса 3 Подсчет коммерческих студентов по группам*. Для этого:

– создать макрос, выбрав макрокоманду *Открыть запрос*;

– в строку *Имя запроса* выбрать название запроса из раскрывающегося списка;

– на ленте найти команду *Выполнить макрос*, запустить его и сохранить с именем *Запрос Подсчет коммерческих студентов по группам*.

6. Аналогичным образом создать макросы с использованием макрокоманда для открытия одной из ранее созданных форм (форма с вкладками) и одного отчета (отчет по среднему баллу).

7. Расположить в ранее созданной форме МЕНЮ (см. Пункт 1, 2) все созданные *макросы* сверху вниз. Для этого нужно открыть форму в режиме конструктора и перетащить мышью все созданные значки макросов (кроме макроса *Autoexec*) из вкладки окна базы данных в окно формы МЕНЮ. Подобрать надлежащий размер и местоположение значков, превратившихся в кнопки меню.



Рис. 57. Вид окна конструктора формы МЕНЮ

8. Проверить работу Меню в режиме формы.

9. Перейти в режим конструктора форм. Отключить вертикальные и горизонтальные полосы прокрутки, область выделения и поле номера записи. Для этого, нажав правую кнопку мыши, вызвать контекстное меню и выбрать *Свойства* для объекта форма. Открыть в окне вкладку *Макет*. Установить значение для свойства *Полосы прокрутки* –

Отсутствуют. Установить для свойств Область выделения и Кнопки навигации значение – Нет.

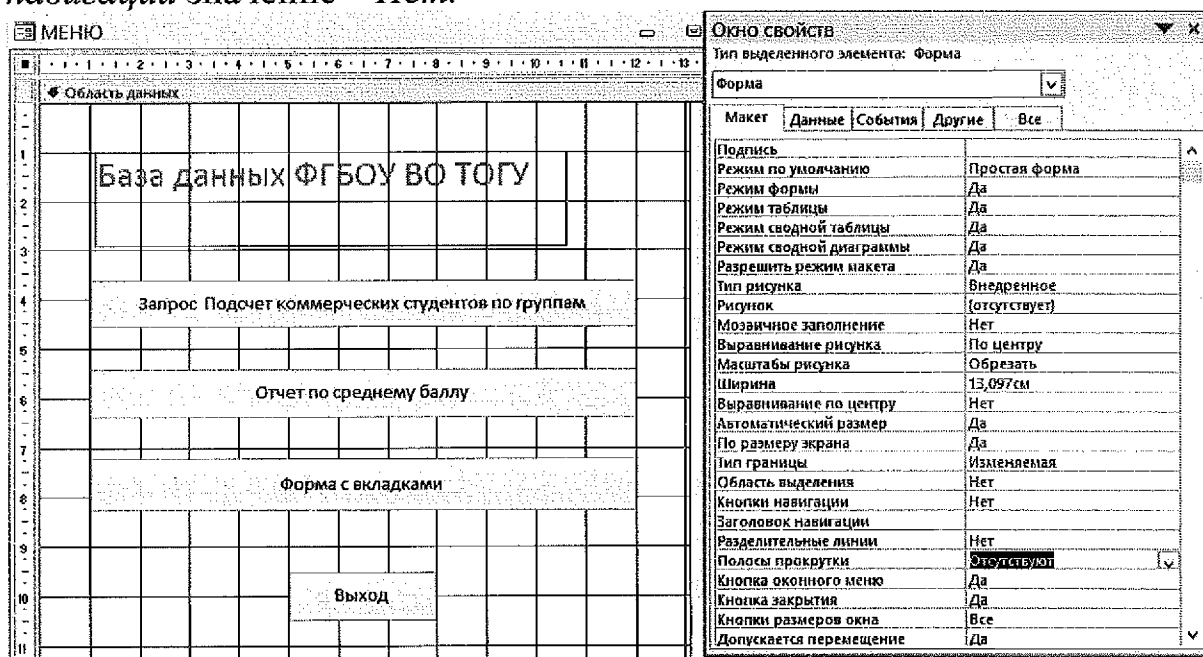


Рис. 58. Окно свойств формы

10. Закрывать базу данных и вновь ее загрузить. Форма МЕНЮ должна автоматически появиться на экране.

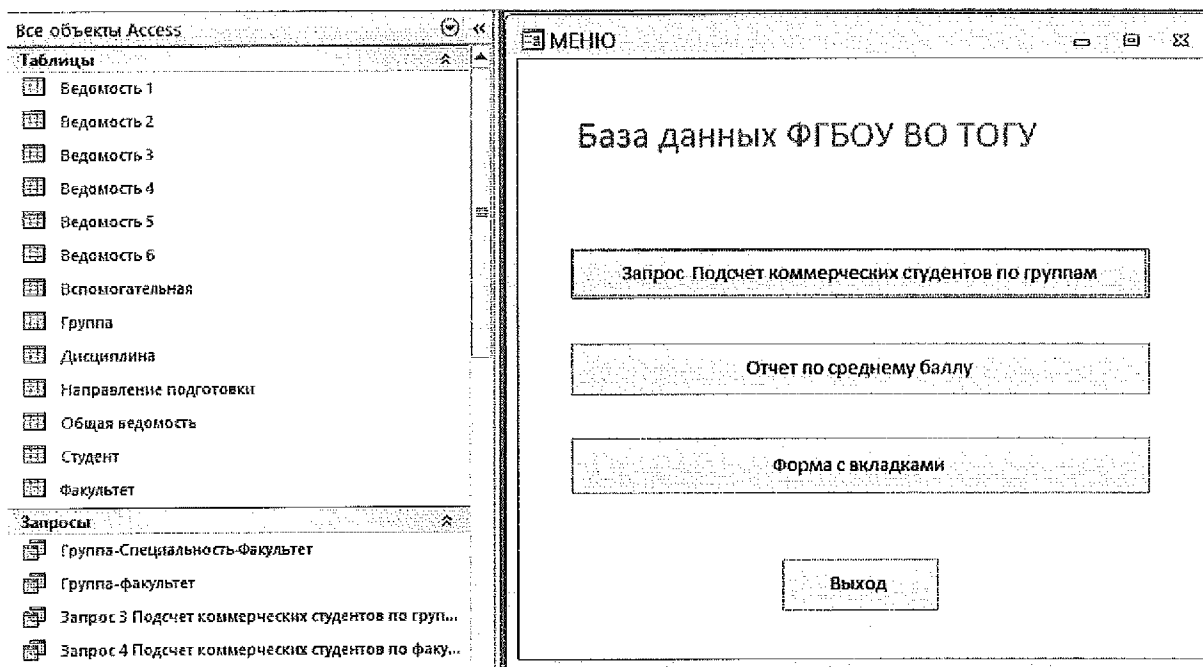


Рис. 59. Окно формы МЕНЮ

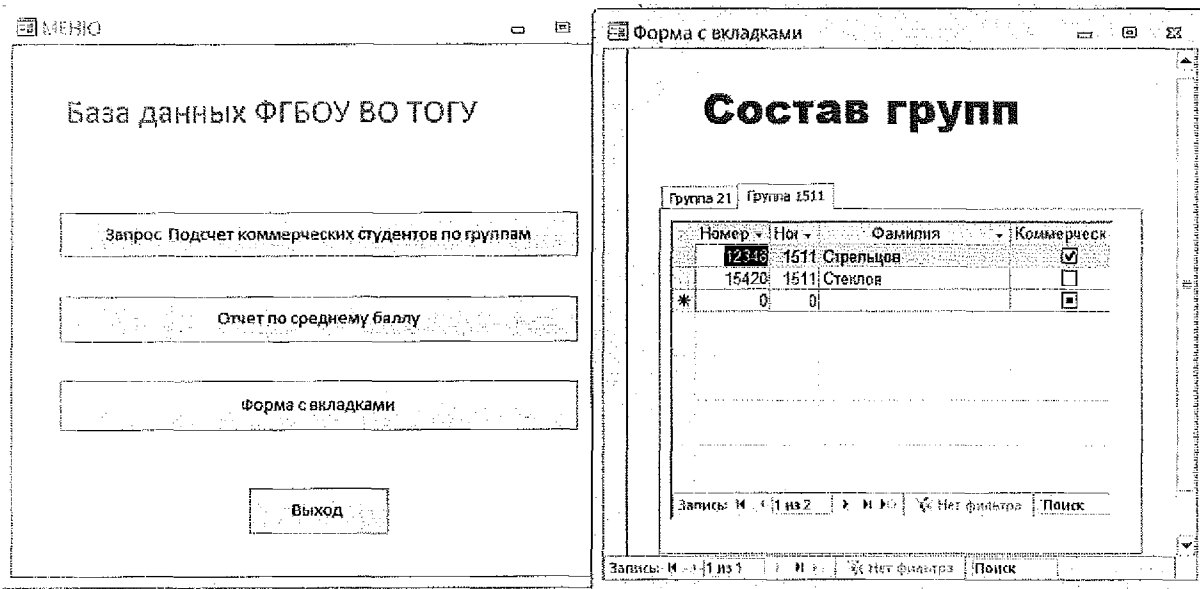


Рис. 60. Вид окна с открытой формой МЕНЮ (после перезагрузки) с выполненным макросом по открытию формы с вкладками

### Создание меню на ленте

1. В главном меню программы выбрать *Файл / Параметры*. На панели слева выбрать *Настройка ленты*. Найти справа внизу кнопку *Создать вкладку*. Переименовать созданную вкладку на МЕНЮ, нажав правую кнопку мыши и в контекстном меню выбрав команду *Переименовать*.
2. Выделив МЕНЮ, выбрать создать группу, назвав ее формы, и создать вторую группу, назвав ее запросы.

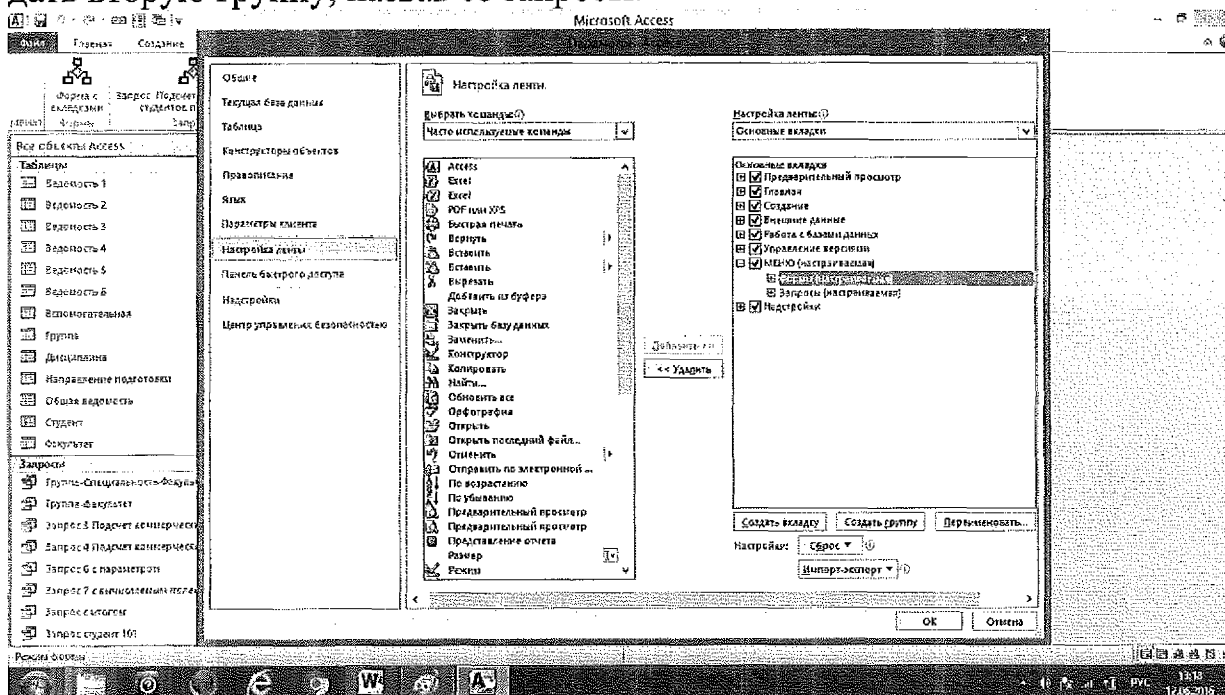


Рис. 61. Создание меню на ленте

3. Выделить группу формы, выбрать команду Макросы, появится окно с созданными макросами. Выделить макрос *Форма с вкладками* и добавить в группу (кнопка *Добавить*). Аналогично в группу запросы добавить макрос *Запрос Подсчет коммерческих студентов по группам*.

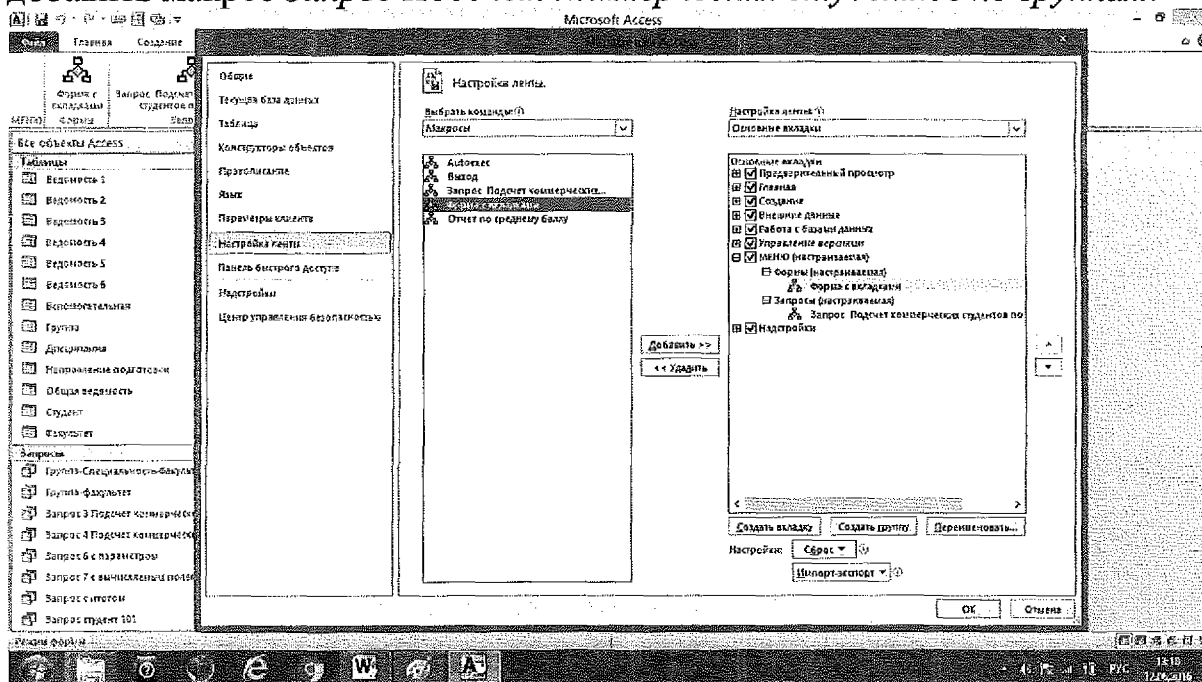


Рис. 62. Добавление макросов на ленту

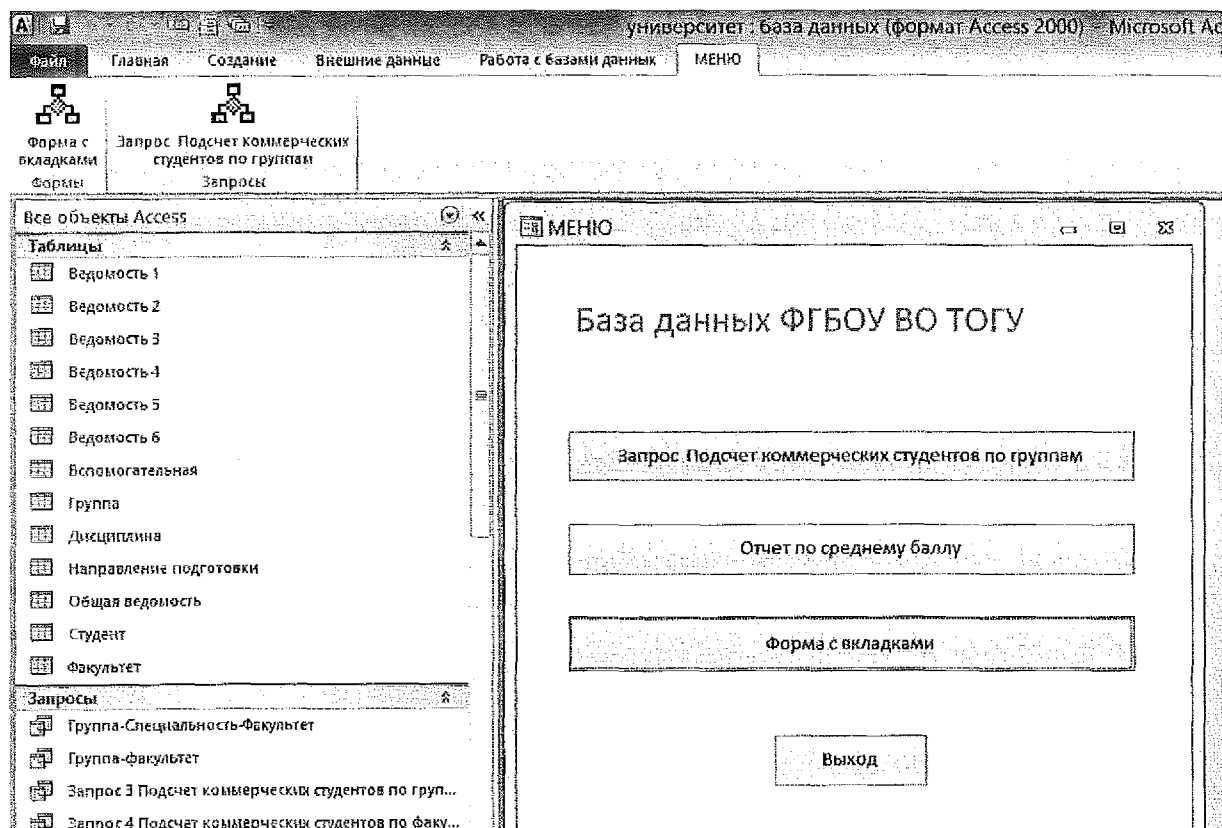


Рис. 63. Вкладка меню на ленте

## 2.10 Фильтры и защита базы данных

### Освоение приемов работы с фильтрами в таблицах

**Задание 1.** Найти студентов, фамилия которых начинается на заданную букву, например на букву «В». Список найденных студентов должен быть упорядочен по алфавиту. Для поиска использовать **фильтр**.

#### Технология:

Открыть таблицу *Студенты* и применить фильтр.

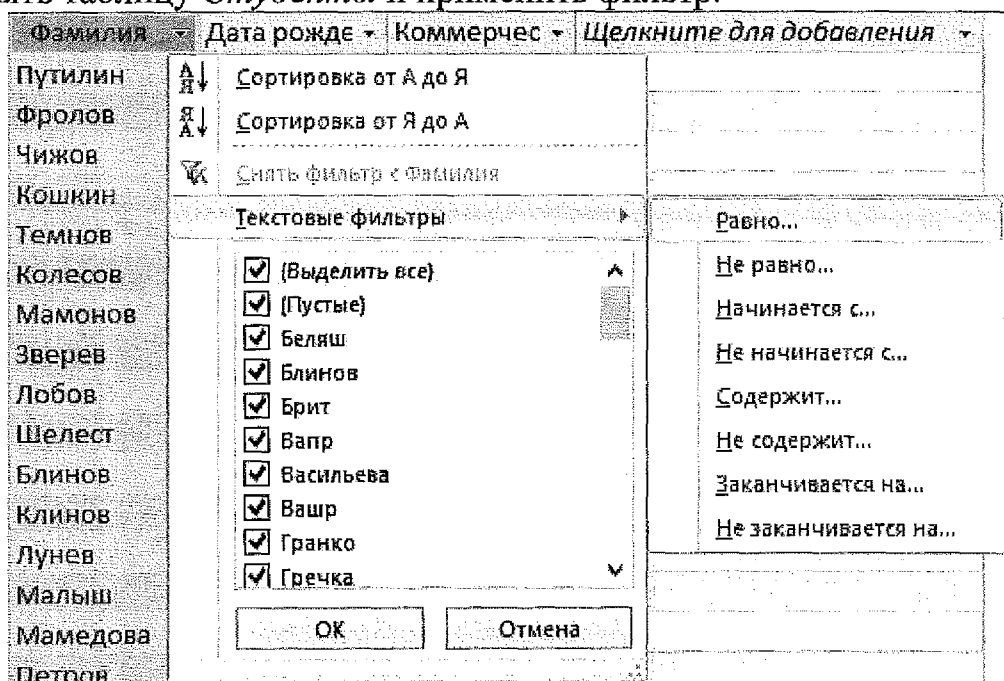


Рис. 64. Применение фильтра

**Задание 2.** Найти студентов, родившихся в заданном году с помощью фильтра.

**Задание 3.** Найти студентов, родившихся в заданном году и обучающихся на коммерческой основе в заданной группе.

#### Защита базы данных.

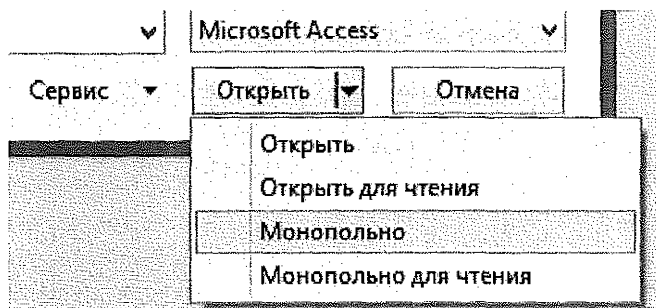
Microsoft Access обеспечивает два традиционных способа защиты базы данных: установка пароля, требуемого при открытии базы данных, и защита на уровне пользователей, которая позволяет ограничить, к какой части базы данных пользователь будет иметь доступ или какую ее часть он сможет изменять. Кроме того, можно предотвратить изменения структуры форм, отчетов и модулей, сохранив базу данных как файл MDE.

#### Установка пароля.

Простейшим способом защиты является установка пароля для открытия базы данных. После того, как пароль установлен, при каждом открытии базы данных будет появляться диалоговое окно, в которое требуется ввести пароль.



Для установки пароля следует открыть базу данных в монопольном режиме (команда **ФАЙЛ / Открыть**, в списке кнопки *Открыть* выбрать режим *Монопольно*) и выполнить команду **ФАЙЛ / Задать пароль базы данных**.



#### Задать пароль базы данных

Использование пароля для ограничения доступа к базе данных. Файлы в формате Microsoft Access 2007 или более поздней версии будут шифроваться.

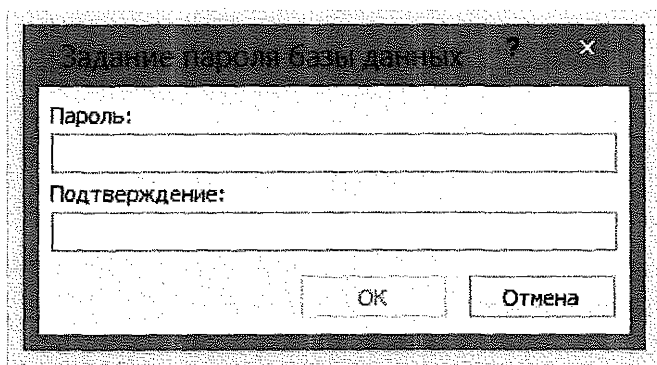


Рис. 65. Окно задания пароля базы данных

Таким образом, были рассмотрены свойства и характеристики интерфейса и инструментов СУБД MS Access и этапы проектирования базы данных «Университет».

### **3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРОВЕДЕНИЮ ЗАНЯТИЙ ПО ДИСЦИПЛИНАМ «БАЗЫ ДАННЫХ» И «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»**

#### **3.1 Цели освоения дисциплин**

Целями освоения дисциплин *«Базы данных»* и *«Информационные системы»* являются: теоретическое и практическое освоение методов и технологий формирования современных баз данных, являющихся основой любой информационной системы, создаваемой в разных сферах человеческой деятельности.

*Задачи дисциплин:* развивать следующие компетенции: знания в области построения баз данных как ядра любой прикладной информационной системы; умения и навыки, требуемые для формирования общекультурных и профессиональные компетенций, реализация которых приводит к созданию основных объектов профессиональной деятельности – баз данных.

#### **3.2 Место дисциплин в структуре ОПОП ВО бакалавриата**

*Цикл, к которому относятся дисциплины*

*«Базы данных»* является дисциплиной базовой части цикла профессиональных дисциплин ФГОС (БЗ.Б.5) и изучается в 7-м семестре бакалаврами направления подготовки «Прикладная математика и информатика».

*«Информационные системы»* является дисциплиной базовой части цикла профессиональных дисциплин (БЗ.Б.ОД.24) и изучается в 6 семестре бакалаврами направления подготовки «Педагогическое образование» по профилю «Математика и информатика».

Дисциплины *«Базы данных»* и *«Информационные системы»* основываются на учебных курсах, входящих в модули следующих дисциплин: «Архитектура компьютера», «Компьютерные сети», «Алгоритмы и алгоритмические языки», читаемых в 1–6-м семестрах.

*Требования к входным знаниям, умениям и компетенциям студентов, необходимым для освоения дисциплин*

Для освоения дисциплин *«Базы данных»* и *«Информационные системы»* студенту необходимы следующие компетенции:

**знать:** основы информатики, назначение и виды алгоритмов и алгоритмических языков, языки и методы программирования, методы структурного и объектного программирования, основы архитектуры и процессов функционирования компьютера, сетей и телекоммуникаций, функциональные и технологические стандарты разработки программных комплексов;

**уметь:** анализировать и оценивать социальную информацию; проводить системный анализ предметной области; выбирать и оценивать архитектуру вычислительных систем, сетей и систем телекоммуникаций и их подсистем; понимать и применять на практике компьютерные технологии для решения различных задач; разрабатывать программные приложения;

**владеть:** навыками программирования в современных средах; навыками тестирования и документирования ПК; навыками решения научных и практических задач.

*Дисциплины, для которых данные дисциплины  
являются предшествующими*

Основные компетенции, полученные при изучении дисциплин «Базы данных» и «Информационные системы» являются необходимыми входными знаниями для учебной и преддипломной практик, предусмотренных ФГОС по направлениям подготовки «Прикладная математика и информатика», «Педагогическое образование».

Изучение дисциплин является базой для дальнейшего освоения студентами курсов по выбору профессионального цикла.

**3.3 Компетенции обучающегося, развиваемые в результате  
освоения дисциплин «Базы данных»  
и «Информационные системы»**

Процесс освоения дисциплин «Базы данных» и «Информационные системы» направлен на развитие следующих компетенций:

– готовность использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования (ПК-11);

– способность руководить учебно-исследовательской деятельностью учащихся (ПК-12);

– готовность применять знания теоретической информатики, фундаментальной и прикладной математики для анализа и синтеза информационных систем и процессов (СКИ-1);

– владение современными формализованными математическими, информационно-логическими и логико-семантическими моделями и методами представления, сбора и обработки информации (СКИ-3);

– способность использовать современные информационные и коммуникационные технологии для создания, формирования и администрирования электронных ресурсов (СКИ-6);

– умение анализировать и проводить квалификационную экспертную оценку качества электронных образовательных ресурсов и программ-

но-технического обеспечения для их внедрения в учебно-образовательный процесс (СКИ-7).

**В результате освоения дисциплин «Базы данных» и «Информационные системы» обучающийся будет (на основе ФГОС) –**

**знать:** основы предметной области: понятие «информационная система», классификации информационных систем, СУБД и их назначение, подходы к проектированию информационных систем, основные операторы языка структурированных запросов SQL, способы защиты информационной системы от несанкционированного пользователя; современные технологии проектирования ИС.

**уметь:** решать задачи предметной области: различать виды информационных систем; осуществлять инфологическое и даталогическое проектирование информационных систем; создавать информационные системы с помощью пакета прикладных программ MS Office (MS Access); использовать операторы языка SQL для создания баз данных в программе SQLITE; осуществлять защиту баз данных в офисных приложениях; решать задачи ЕГЭ по базам данных.

**владеть:** создания информационных систем в сфере образования с применением современных технологий.

### 3.4 Структура и содержание дисциплин

#### Общая трудоемкость дисциплин

Трудоемкость дисциплин «Базы данных» и «Информационные системы» составляет 4 зачетных единицы 144 часа (Табл. 4, 5, 6).

Таблица 4

#### Общая трудоемкость дисциплины

Вид учебной работы	Количество часов (7-й семестр)
Аудиторные занятия	64
Лекции	32
Лабораторные работы (ЛР)	32
Самостоятельная работа	80
Курсовой проект: «Разработка проекта базы данных, включающей не менее 20 записей, с применением современных информационных технологий»	40
Рефераты, индивидуальные задания, глоссарии	40
Вид промежуточной аттестации (зачет, экзамен)	зачет
Общая трудоемкость	144
Зачетные единицы	4

Таблица 5

## Разделы дисциплин и виды занятий

Раздел дисциплины	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
		Лекции	Лаб. раб.	СРС	
Введение в БД	1	2		2	Составление глоссария
Теоретико-графовые модели	2	2	2	2	Составление глоссария
Теоретико-множественные модели данных	3	2	4	4	Индивидуальное задание по реляционной алгебре
Концептуальное (инфологическое) проектирование	4	4	6	4	Индивидуальное задание по проектированию БД методом нормальных форм
Дагалогическое проектирование	5	2	4	4	Реферат
Информационные хранилища данных	6	2	2	4	Реферат
Язык QBE	7-8	2	4		
Язык SQL	9-11	4	4	6	Индивидуальное задание по формированию SQL-запросов к БД
Модификация баз данных. Модели транзакций	12-13	2			
Анализ систем управления БД	14	2			
Разработка приложений с использованием СУБД	15-16	2	4		Индивидуальное задание
Администрирование баз данных. Защита БД	17	2	2		Лабораторная работа
Распределенные базы и банки данных	18	4		10	Реферат, тестирование

Раздел дисциплины	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
Курсовой проект: «Разработка проекта базы данных, включающей не менее 20 записей, с применением современных				44	Курсовой проект
		32	32	80	144 часа, 4 зачетные единицы

Таблица 6

## Содержание разделов дисциплин

Наименование раздела	Содержание
Введение в БД	<p>Определение базы данных и банки данных Архитектура БД, классификация моделей данных. Понятие физической и логической записей, их независимость.</p> <p>Простые и агрегатные типы данных. Основные структуры: массивы, деревья, списки, графы. Операции над данными. Понятие предметной области.</p> <p>Модель предметной области, модель организации данных, модель управления доступом.</p> <p>Декларативный и процедурный способ отображения объектов и отношений. Внутренняя и внешняя схема. Объектно-ориентированный подход.</p> <p>Понятие жизненного цикла БД. Двух- и трехуровневый методы проектирования БД</p>

Наименование раздела	Содержание
Теоретико-графовые модели	<p>Иерархическая модель данных. Принципы организации иерархической модели данных, структуры, используемые для связи данных. Основные преимущества и недостатки иерархической модели</p> <p>Сетевая модель данных. Принципы организации сетевой модели данных, структуры, используемые для связи данных. Основные преимущества и недостатки сетевой модели</p> <p>Объектно-ориентированная модель данных. Принципы организации объектно-ориентированной модели данных, структуры, используемые для связи данных. Основные преимущества и недостатки объектно-ориентированной модели</p>
Теоретико-множественные модели данных	<p>Реляционные модели. Основные определения. Понятие отношения, кортежей и полей. Атрибуты отношения, домены, полное декартово произведение</p> <p>Теоретико-множественные и специальные операции реляционной алгебры. Объединение, пересечение и разность отношений, сцепление отношений, расширенное декартово произведение. Преимущества реляционных систем в современных условиях. Множество образов, операция деления</p> <p>Нормализация отношений. Функциональный и предметный подходы системного анализа предметной области. Корректные схемы БД, логическое проектирование, функциональные зависимости атрибутов отношений</p> <p>Нормальные формы отношений. Алгоритм нормализации отношений</p>
Концептуальное (инфологическое) проектирование	<p>Принципы инфологического (концептуального) проектирования баз данных. Анализ и декомпозиция предметной области. Моделирование локальных представлений</p> <p>Модель «сущность-связь» (ER). Сущности, атрибуты, связи, сущности-связи</p> <p>Модификации концептуальной схемы. Особенности построения ER-диаграмм с использованием среды ERWin, модификация ER-диаграмм</p>

Наименование раздела	Содержание
Даталогическое проектирование	<p>Понятие даталогического проектирования. Определение состава баз данных. Критерии оценки баз данных</p> <p>Проектирование логической модели данных. Отображение концептуальной модели базы данных на выбранную модель данных. Принципы и особенности отображения на реляционную модель. Правила отображения</p> <p>Создание физической модели. Сравнение логической и физической модели баз данных</p>
Информационные хранилища данных	<p>Многомерные модели данных. Особенности построения многомерных моделей данных. Архитектуры хранилищ данных. Связи между данными в информационных хранилищах данных. Организация доступа в хранилищах данных</p>
Язык QBE	<p>Общая характеристика языка QBE. Назначение и функциональность. Процедуры формирования реляционных баз данных. Процедуры управления БД с помощью языка QBE. Формирование запросов к связанным таблицам. Выполнение агрегирующих операторов</p>
Язык SQL	<p>Структура SQL. Язык SQL как стандарт определения данных и манипулирования данными в реляционных моделях. История возникновения языка SQL, современные тенденции его развития. Структура SQL</p> <p>Типы данных. Основные операторы определения данных – SELECT, CREATE, ALTER, DROP. Понятие целостности реляционной модели данных. Основные операторы манипулирования данными – INSERT, DELETE, UPDATE</p> <p>Построение баз данных с помощью SQL. Описание схем отношений, доменов, ограничений, представлений данных</p> <p>Реализация запросов к БД с использованием SQL, создание представлений. Реализация операций реляционной алгебры в SQL</p>
Модификация баз данных. Модели транзакций	<p>Понятие операции транзакции. Основные свойства транзакций – атомарность, согласованность, изолированность и долговечность. Фиксация и откат транзакций</p> <p>Назначение и использование журнала транзакций. Индивидуальные откаты транзакций, восстановление БД после мягкого и жесткого сбоев. Параллельное выполнение транзакций. Захваты и блокировки</p>



Наименование раздела	Содержание
Анализ систем управления БД	Понятие «система управления базами данных». Назначение СУБД, их функциональность. Требования к обеспечению целостности данных, их непротиворечивости и масштабируемости. Типы современных СУБД. Классификация СУБД
Разработка приложений с использованием СУБД	Принципы создания экранных форм в среде СУБД, выбор способов создания экранных форм; подход к созданию интерфейсов; создание страниц доступа; создание отчетов в различных средах СУБД
Администрирование баз данных. Защита БД	<p>Принципы документирования жизненных циклов баз данных</p> <p>Функции и принципы администрирования баз данных. Управление доступом к ресурсам и целостность БД. Защита базы данных. Идентификация пользователей. Проверка и назначение полномочий и представлений данных пользователей. Контроль параллельной обработки. Обслуживание и восстановление базы данных. Источники отказов и сбоев. Резервное копирование данных. Процедуры восстановления</p> <p>Администраторы и конечные пользователи: права и функции. Пример систем управления БД. Основные функции и компоненты. Стандартизация программных интерфейсов</p>
Распределенные базы данных	<p>Классификация БД по способу доступа. Локальные, сетевые и распределенные базы данных. Архитектура «файл-сервер». Сетевые базы данных. Архитектура «клиент/сервер». Распределенные базы данных. Общая модель распределенной системы баз данных</p> <p>Двух и трехуровневая система «клиент-сервер». Распределение данных и запросов. Обработка распределенных данных и запросов</p> <p>Модели сервера баз данных. Многопоточковые и многосерверные архитектуры. Модель сервера приложений</p>

### 3.5 Образовательные технологии

- Исследовательские и проектные методы при выполнении индивидуальных заданий.
- Технология модульного обучения.
- CASE (англ. Computer-Aided Software Engineering)-технологии – набор инструментов и методов программной инженерии для проектирова-

ния программного обеспечения, который помогает обеспечить высокое качество программ, отсутствие ошибок и простоту в обслуживании программных продуктов.

– Технология использования в обучении игровых методов: ролевых, деловых обучающих игр в процессе выполнения курсового проекта.

– Информационно-коммуникационные технологии в процессе тестирования.

– Технологии интерактивного обучения (форумы, чаты) в процессе выполнения курсового проекта.

### **3.6 Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплин**

#### Темы рефератов

1. Роль информационных систем в обеспечении решения задач управления.

2. Основные проблемы информационных систем, сетей и вычислительной техники в России.

3. Общие требования, предъявляемые к современным информационным системам.

4. Жизненный цикл информационных систем.

5. Методы анализа и синтеза структуры информационных систем.

6. Фактографические системы.

7. Принципы построения и оценка уровня безопасности в информационных системах и сетях.

8. Современные тенденции развития стандарта SQL.

9. Современные тенденции развития реляционных моделей данных.

10. Сопоставительный анализ моделей данных.

11. Управление реляционными БД.

12. Целостность данных: ограничения целостности.

13. Распределенные базы данных: преимущества, примеры, проблемы.

14. Хронологические базы данных.

15. Объектно-ориентированные базы данных: преимущества реального сближения двух технологий.

16. Динамический SQL.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплин включают в себя вопросы для самоконтроля и задания для самостоятельной работы, представленные в следующей главе.

### 3.7 Рейтинговая система для определения индивидуальной динамики развития компетенций бакалавра по дисциплинам «Базы данных», «Информационные системы»

Приведем пример таблицы, определяющей рейтинговую систему оценивания достижений бакалавра по дисциплинам «Базы данных», «Информационные системы» (Табл. 7).

Таблица 7

Рейтинговая система				
Наименование дисциплин	Уровень образования	Статус курса в учебном плане	Количество зачетных единиц / кредитов	
«Базы данных», «Информационные системы»	бакалавриат	БЗ.Б.ОД.24 БЗ.Б.5	4	
БАЗОВАЯ ЧАСТЬ				
Тема или задание текущей аттестационной работы	Продукт текущей аттестации	Аудиторная или внеаудит.	Мин. кол-во баллов	Макс. кол-во баллов
Составление индивидуального или группового тематического аннотированного каталога электронных образовательных ресурсов по базам данных	Опубликованный в сети аннотированный каталог	Аудит. и внеаудит.	5	10
Создание объектов баз данных на примере БД «Университет». Обработка данных. Групповая совместная работа с сетевым ресурсом	Таблицы, отчеты, формы, запросы, аналитическая таблица в Google Docs для работы в режиме «онлайн» с описанием перечисленных объектов	Аудит. и внеаудит.	15	25
Создание базы данных предметной области для решения образовательных задач.	База данных, созданная в рамках курсового проекта	Внеаудит.	5	10
Аннотированный обзор и критический анализ ресурсов нового поколения в области баз данных	Реферат	Аудит. и внеаудит.	10	15

Окончание табл. 7

Наименование дисциплин	Уровень образования	Статус курса в учебном плане	Количество зачетных единиц / кредитов	
Сравнительный анализ тестовых заданий (выявление типовых задач), предложенных в пособии в пункте 4.	Сравнительная таблица с ссылками или интеллектуальная карта	Аудит. и внеаудит.	5	10
Анализ правовых аспектов использования информационных систем	Выполнение тестовых заданий, участие в сетевой дискуссии	Аудит.	5	10
Формирование в сети и представление в аудитории электронного портфолио с выполненными заданиями: база данных «Университет», курсовой проект по выбранной предметной области, каталоги, обзор и анализ ресурсов по дисциплине	Электронное портфолио студента или группы	Аудит. и внеаудит.	10	10
Итого: минимум			55	90

## ДОПОЛНИТЕЛЬНАЯ ЧАСТЬ

Тема или задание текущей аттестационной работы	Продукт текущей аттестации	Аудиторная или внеаудит.	Мин. кол-во баллов	Макс кол-во баллов
Создание теста по теме «Базы данных»	Тест	Внеаудит.	5	10
Анализ проблем в сфере внедрения баз данных	Публикация в блоге	Внеаудит.		
Разработка фрагмента занятия с использованием созданной базы данных как электронного образовательного ресурса	Описание фрагмента занятия	Внеаудит.		
Итого: максимум			60	100

При этом используется мониторинг сетевой образовательной деятельности обучающихся, осуществляемый через учет динамики накопления продуктов деятельности в электронном портфолио.

В портфолио включаются следующие рубрики:

– «Ресурсы» – библиография использованных источников по дисциплине, аннотированный список Интернет-ресурсов;

– «БД «Университет» – создание объектов баз данных на примере БД «Университет».

– «Курсовой проект» – разработка базы данных по выбранной предметной области;

– «Отчет и рефлексия» – описание результатов и процесса разработки проекта.

### **3.8 Материально-техническое обеспечение дисциплины**

Лекционные занятия проводятся с применением интерактивной доски для демонстрации этапов создания и проектирования баз данных. Лабораторные занятия проводятся с применением СУБД в компьютерном классе.

Для реализации программы дисциплин «Базы данных», «Информационные системы» используются различные технологии: во время аудиторных лекционных занятий применяются проекторы и ПК для презентации лекций; на лабораторных занятиях – компьютеры и специальное программное обеспечение; каждое рабочее место оборудовано ПК.

Самостоятельная работа (72 часа) выполняется студентами также с использованием ПК в домашних условиях, либо в библиотеке института по специальным заданиям в соответствии с методическими материалами и рейтинговой таблицей; для более глубокого изучения научного материала ряда лекций, имеющих перспективную направленность, студентам предлагаются темы рефератов. Для выполнения самостоятельного курсового проекта представлена тематика предметных областей для разработки баз данных.

## 4 ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

### Вопросы для самоконтроля

1. *Что такое база данных (выберите правильный вариант)?*
  - а) реализованная с помощью компьютера информационная структура (модель), отражающая состояния объектов и их отношения;
  - б) минимальная именованная структурная единица данных;
  - в) совокупность нормализованных отношений, логически взаимосвязанных и отражающих некоторую предметную область;
  - г) универсальное программное средство, предназначенное для обработки информации.
  
2. *Что такое система управления базами данных (выберите правильный вариант)?*
  - а) операции обработки данных;
  - б) реализованная с помощью компьютера информационная структура (модель), отражающая состояния объектов и их отношения;
  - в) совокупность взаимосвязанных структур данных и операций над этими структурами;
  - г) комплекс программных и языковых средств, необходимых для создания и модификации базы данных, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой.
  
3. *Какой из типов моделей данных не поддерживают СУБД (выберите неправильный вариант)?*
  - а) нелинейный;
  - б) иерархический;
  - в) сетевой;
  - г) реляционный.
  
4. *Какие модели баз данных существуют (выберите неправильный ответ)?*
  - а) сетевые;
  - б) иерархические;
  - в) нелинейные;
  - г) реляционные.
  
5. *Какой программный продукт не является СУБД (выберите правильный вариант)?*

- а) OpenOffice.org Calc;
- б) OpenOffice.org Base;
- в) MS Access;
- г) MS FoxPro.

6. *Какую базу данных называют реляционной (выберите правильный вариант)?*

- а) совокупность взаимосвязанных структур данных;
- б) универсальное программное средство, предназначенное для обработки информации;
- в) комплекс программных и языковых средств, необходимых для создания и модификации базы данных, добавления, модификации, удаления, поиска и отбора информации, представления информации на экране и в печатном виде, разграничения прав доступа к информации, выполнения других операций с базой;
- г) совокупность нормализованных таблиц, между которыми могут существовать связи по ключевым значениям.

7. *Что такое поле таблицы (выберите правильный вариант)?*

- а) объекты, которые служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- б) полный набор данных об определенном объекте;
- в) элемент таблицы, который содержит данные определенного рода;
- г) специальная структура, предназначенная только для вывода данных.

8. *Что такое первичный ключ (выберите правильный вариант)?*

- а) главный ключевой элемент (поле), однозначно идентифицирующий строку в таблице;
- б) совокупность взаимосвязанных структур данных;
- в) элементарная единица логической организации данных;
- г) поле, значение которого может повторяться в нескольких записях базы.

9. *Для чего предназначены таблицы (выберите правильный вариант)?*

- а) для хранения данных в виде записей (строк) и полей (столбцов);
- б) только для вывода данных;
- в) для ввода данных;
- г) для извлечения данных из таблиц и предоставления их пользователю в удобном виде.

10. *Для чего предназначены формы (выберите правильный вариант)?*

- а) только для вывода данных;
- б) для ввода данных;

- в) для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- г) для хранения данных в виде записей (строк) и полей (столбцов).

*11. Для чего предназначены запросы (выберите правильный вариант)?*

- а) для хранения данных в виде записей (строк) и полей (столбцов);
- б) для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- в) только для вывода данных;
- г) для ввода данных.

*12. Для чего предназначены отчеты (выберите правильный вариант)?*

- а) для извлечения данных из таблиц и предоставления их пользователю в удобном виде;
- б) только для вывода данных;
- в) только для ввода данных;
- г) для хранения данных в виде записей (строк) и полей (столбцов);

*13. С какого этапа начинается процесс создания базы данных? (выберите правильный вариант)?*

- а) определение связей между таблицами;
- б) усовершенствование структуры базы данных;
- в) определение цели создания базы данных;
- г) определение полей с уникальными значениями в каждой записи.

*14. Для чего организуется связь между таблицами (выберите неправильный вариант)?*

- а) для создания форм;
- б) для создания отчетов;
- в) для внесения данных в таблицы;
- г) для создания запросов.

*15. Каким требованиям должна удовлетворять таблица, чтобы можно было провести ее нормализацию (выберите неправильный вариант)?*

- а) таблица должна содержать нуль или более записей;
- б) все записи таблицы должны иметь одно и то же множество полей, причем одноименные поля должны относиться к одинаковым типам данных;
- в) таблица не должна содержать двух полностью идентичных записей;
- г) все записи таблицы должны иметь одно и тоже множество полей, причем одноименные поля должны относиться к различным типам данных.



16. Сколько нормальных форм существует (выберите правильный вариант)?

- а) одна;
- б) три;
- в) пять;
- г) шесть.

17. Что такое первая нормальная форма (выберите правильный вариант)?

- а) каждый атрибут таблицы атомарен и все строки различны;
- б) все атрибуты таблицы являются простыми и каждый неключевой атрибут функционально полно зависит от ключа;
- в) каждый неключевой атрибут нетранзитивно зависит от первичного ключа;
- г) каждый атрибут таблицы атомарен и все строки одинаковы.

18. Возможна ли сортировка данных в полях таблицы (выберите правильный вариант)?

- а) да;
- б) нет.

*Правильные ответы:* 1. а; 2. г; 3. а; 4. в; 5. а; 6. г; 7. в; 8. а; 9. а; 10.б; 11. б; 12. б; 13. в; 14. в; 15. г; 16. в; 17. а; 18. а.

### **Задания для самостоятельной работы**

1. Создайте базу данных в соответствии с выбранным вариантом.
2. Создайте структуру таблиц в соответствии с заданием в режиме конструктора таблиц.
3. Создайте связи между таблицами в окне конструктора связей, в соответствии с заданием. Обратите внимание, что первичные ключи таблиц подчеркнуты, а внешние ключи имеют те же имена полей, что и первичные ключи в таблицах, с которыми должна быть связь.
4. Заполните таблицы произвольными данными.
5. Создайте ряд запросов на выборку данных из таблиц вашего задания с использованием мастера запросов.
6. Создайте ряд экранных форм для ввода и модификации данных таблиц вашего задания, с использованием мастера форм.
7. Создайте несколько отчетов для представления данных из таблиц вашего задания в печатном виде, с использованием мастера отчетов.

## Варианты заданий

### Вариант 1

**Предметная область ИС:** библиотека

Минимальный список характеристик:

- код книги, автор книги, название, год издания, цена;
- номер читательского билета, ФИО, адрес, телефон читателя;
- код книги, номер читательского билета, дата выдачи книги читателю и дата сдачи книги читателем.

### Вариант 2

**Предметная область ИС:** диспансеризация

Минимальный список характеристик:

- код пациента, ФИО, дата рождения пациента, телефон;
- код врача, ФИО, должность, специализация врача;
- код пациента, код врача, дата посещения, диагноз, поставленный данным врачом данному пациенту.

### Вариант 3

**Предметная область ИС:** кулинария

Минимальный список характеристик:

- код блюда, наименование блюда, цена готового продукта;
- код продукта, наименование продукта, калорийность;
- код блюда, код продукта, количество продукта в данном блюде.

### Вариант 4

**Предметная область ИС:** ремонт автомобилей

Минимальный список характеристик:

- код владельца, ФИО, адрес и телефон владельца автомобиля;
- код мастерской, название, адрес, телефон ремонтной мастерской;
- код автомобиля, марка, номер, цвет автомобиля; код владельца автомобиля;
- код мастерской, код автомобиля, дата начала ремонта, дата окончания ремонта, стоимость ремонта.

### Вариант 5

**Предметная область ИС:** услуги мобильной связи

Минимальный список характеристик:

- код клиента, ФИО, адрес, телефон клиента;
- код оператора, наименование оператора, адрес, телефон;
- код услуги, наименование услуги, код оператора, стоимость;
- код клиента, код услуги, дата подключения услуги, дата отключения услуги.

## Тематика курсовых проектов по предметным областям

### **Вариант 1. Специализированная библиотека**

Рассмотрим специализированную библиотеку, которая располагает книжным фондом определенной тематической направленности. Предполагается, что каждая книга фонда может быть как в одном экземпляре, так и в нескольких, поэтому каждой книге соответствует уникальный инвентарный номер и библиотечный код. Данные о книге содержатся в библиографической карточке, карточки объединяются в каталоги. Существует два вида каталогов: алфавитный и тематический; в алфавитном каталоге карточки отсортированы по фамилии автора, а в тематическом – сначала по темам, а в пределах каждой темы – по фамилии автора.

Библиотека выдает книги читателям во временное пользование. При записи в библиотеку каждому читателю присваивается порядковый номер, ему выдается читательский билет и для него заводится учетная карточка. Учетная карточка кроме данных о читателе в дальнейшем будет содержать информацию о выданных и возвращенных книгах.

Данные, характеризующие работу библиотеки с книгами и читателями, можно сгруппировать следующим образом:

- *книжный фонд* (инвентарный номер книги, библиотечный код книги, отметка о выдаче/возвращении);
- *каталог* (библиотечный код книги, автор, название, издательство, год издания, количество страниц, тема, цена);
- *читатели* (номер читательского билета, фамилия, имя, отчество, домашний адрес, домашний телефон, рабочий телефон);
- *выдача книг* (инвентарный номер книги, номер читательского билета, дата выдачи, запланированная дата возврата).

Создаваемая информационная система предназначена, прежде всего, для ведения данных: о книгах (регистрация новых поступлений, списание литературы), о читателях (регистрация новых читателей, удаление информации о выбывших читателях), а также о перемещении книг между библиотекой и читателями, что должно найти отражение в таблицах «книжный фонд» и «выдача книг». Кроме того, в системе должны быть реализованы возможности просмотра и поиска как среди книг, так и среди читателей.

### **Вариант 2. Издательство**

Издательство – предприятие, занимающееся выпуском разнообразной печатной продукции. Издательство заключает договор с заказчиком (клиентом) на выполнение заказа. Заказчиком может выступать частное лицо или организация. Частное лицо может быть автором издания (или одним из авторов, если их несколько) или представителем автора. Организа-

ция для контактов с издательством также имеет своего представителя — контактную персону.

Заказ может быть оформлен на издание книги, брошюры, рекламного проспекта, буклета, бюллетеня для голосования или какого-либо другого вида издательской продукции. Подготовленные издательством материалы заказчика печатаются в типографиях, где издательство размещает свои заказы.

Информацию о работе издательства можно сгруппировать следующим образом:

- *сведения о заказчиках* (частное лицо или организация, личные данные контактной персоны, адрес, телефон, факс);
- *сведения о заказах* (номер заказа, заказчик, вид печатной продукции, издание, типография, дата приема заказа, отметка о выполнении, дата выполнения заказа);
- *сведения об изданиях* (код издания, автор и название, объем в печатных листах, тираж, номер заказа);
- *сведения об авторах* (личные данные: ФИО, домашний адрес, телефон; дополнительные сведения);
- *сведения о типографиях* (название, адрес, телефон).

### **Вариант 3. Хозрасчетная поликлиника**

Рассмотрим хозрасчетную поликлинику, например, стоматологическую. Поликлиника ведет прием и учет пациентов, учет их посещений (визитов) и учет обслуживания пациентов специалистами (врачами). Существует необходимость в хранении информации обо всех посещениях поликлиники пациентами и о том, на приеме у каких специалистов они находились.

Хранимую информацию о деятельности хозрасчетной поликлиники и ее пациентах можно сгруппировать так:

- *пациент* (номер истории болезни, ФИО пациента, домашний адрес пациента, телефон);
- *специалист* (личный номер специалиста, ФИО специалиста, специальность, домашний адрес, телефон);
- *визиты* (пациент, специалист, визит первый или повторный, дата визита, анамнез, диагноз, лечение, стоимость лекарств, стоимость услуг).

Необходимо обеспечить ввод, хранение и, возможно, редактирование данных. В определенных случаях необходимо выполнять удаление данных. Например, можно удалить информацию обо всех визитах некоторого пациента, если после его последнего визита прошел определенный срок (например, 3 года), а данные о самом пациенте перенести в архив (или также удалить).

Необходимо предусмотреть поиск сведений о пациентах как по фамилии, так и по номеру истории болезни. Так как поликлиника хозрасчет-

ная, за каждое посещение пациенту выписывается счет, который он должен оплатить.

Кроме задач, перечисленных выше, могут быть также решены и другие задачи, например:

- подсчет выручки каждого специалиста за определенный период (день, месяц);
- подсчет выручки поликлиники в целом за определенный период (день, месяц);
- подсчет оплаченной суммы за лекарства за определенный период (день, месяц).

Также могут решаться задачи статистической обработки данных. Например, подсчет количества посещений поликлиники за месяц в целом и по каждой группе специалистов. А затем на основании этих данных можно построить графики посещений поликлиники по месяцам.

#### **Вариант 4. Ателье мод**

Ателье мод выполняет заказы клиентов на индивидуальный пошив одежды. В ателье существует каталог моделей и каталог тканей. По каталогу моделей клиент выбирает модель, а по каталогу тканей – ткань, из которой будет выполнена модель, и заказывает ее пошив в ателье.

Заказ каждого клиента содержит: ФИО клиента, информацию о модели (ее номер из каталога моделей), информацию о ткани (номер из каталога тканей), ФИО закройщика (исполнителя заказа), дату приема заказа, дату примерки, отметку о выполнении заказа, дату выполнения заказа.

В каталоге моделей каждая модель имеет уникальный номер, для каждой модели указывается рекомендуемая ткань, необходимый расход ткани для данной модели с учетом ширины ткани, цена готовой модели, включающая цену ткани и стоимость пошива изделия.

В каталоге тканей каждая ткань имеет уникальный номер, название, а также в каталоге указываются ее ширина и цена за 1 метр.

В ателье может быть еще и склад тканей. В книге учета тканей на складе для каждой ткани указывается общий метраж, который изменяется, если принимается заказ на изготовление модели из данной ткани.

#### **Вариант 5. Оптовый склад**

Склад осуществляет продажу товаров оптом. Любая фирма, занимающаяся продажей товаров в розницу, закупает необходимые ей товары на складе, который служит посредником между производителями и продавцами.

На склад товар поступает от некоторой фирмы-поставщика, в свою очередь склад продает товар фирме-покупателю, заключая с ней сделку о продаже товара.

Деятельность оптового склада характеризуется следующей информацией, которую можно объединить в группы следующим образом:

- *поставщики* (код поставщика, название фирмы-поставщика, адрес, телефон);
- *покупатели* (код покупателя, название фирмы-покупателя, адрес, телефон);
- *товар на складе* (код товара, поставщик, название товара, единицы измерения, количество, цена покупки за единицу товара, цена продажи за единицу товара);
- *сделки о продаже* (код товара, поставщик, покупатель, количество проданного товара).

На основании описанных данных необходимо вести учет поставщиков, покупателей, продаж, движения товара на складе. Кроме того, можно делать выводы о работе склада, спросе на определенные товары, выгоды работы с некоторыми поставщиками и покупателями.

#### **Вариант 6. Торгово-закупочное предприятие**

Торгово-закупочное предприятие имеет склад, содержащий определенные виды товаров, например, продовольственные товары. Предприятие имеет штат сотрудников, являющихся агентами-реализаторами. Предприятие выдает агенту товар, устанавливая цену его продажи. Агент-реализатор оплачивает выданный товар не сразу, а по мере его реализации, оформляя приходные кассовые ордера. С каждой единицы проданного товара агент получает оплату, установленную предприятием.

Данные, характеризующие торгово-закупочное предприятие, могут быть сгруппированы следующим образом:

- *агенты-реализаторы* (код агента, ФИО агента, адрес, телефон);
- *товары* (код товара, название товара, отпускная цена);
- *товары на складе* (код товара, количество);
- *товары у агента* (код агента, код товара, количество);
- *приходный ордер* (код агента, код товара, количество единиц реализованного товара, общая стоимость реализованного товара).

Необходимо вести учет движения товаров как на складе, так и у агентов-реализаторов. Кроме того, предприятие проводит операции: по новым поступлениям товара, по выдаче товара агенту, по расчету с агентом за реализованный товар.

#### **Вариант 7. Автосалон**

Существует некоторая фирма, торгующая автомобилями. Автомобиль выступает в качестве товара и как товар имеет определенные харак-

теристики. Кроме того, на каждый автомобиль имеются исчерпывающие технические данные. Фирма имеет своих клиентов — покупателей автомобилей, сведения о которых хранит в течение определенного времени.

Деятельность фирмы может быть описана данными, сгруппированными следующим образом:

- *товар* (код товара, страна-изготовитель, марка автомобиля, модель, наличие на складе (да, нет, когда будет), цена);
- *технические данные* (код товара, тип кузова, количество дверей, количество мест, тип двигателя, расположение двигателя, рабочий объем двигателя);
- *клиенты* (код товара, ФИО клиента, паспортные данные (серия, номер), домашний адрес, телефон, доставка (да, нет), вид оплаты (перечисление или наличные: кредит или сразу).

Необходимо обеспечить ввод, редактирование и просмотр данных в удобной для пользователя форме.

Предполагается также решение следующих задач:

- выдать информацию о наличии автомобилей определенной марки и модели;
- выдать технические данные заданной модели;
- выдать информацию обо всех проданных моделях некоторой марки, значение которой вводится в качестве параметра;
- посчитать сумму продаж моделей каждой марки и общую сумму продаж;
- выдать полную или частичную информацию о клиентах фирмы;
- выдать списки клиентов и автомобилей по виду оплаты.

Возможны постановка и решение других задач.

### **Вариант 8. Продажа подержанных автомобилей**

Фирма по продаже подержанных автомобилей работает с физическими лицами – клиентами фирмы, имеющими подержанный автомобиль или автомобили и желающими продать их через фирму. Непосредственной продажей автомобилей занимаются сотрудники фирмы – дилеры. На каждый предлагаемый в продажу автомобиль фирма заключает с клиентом договор, содержащий данные о клиенте, необходимые сведения об автомобиле, а также данные о дилере, обслуживающем этот договор.

Данные, характеризующие деятельность фирмы, могут быть сгруппированы следующим образом:

- *клиенты* (код клиента, ФИО, город, адрес, контактный телефон);
- *дилеры* (код дилера, ФИО, фотография, домашний адрес, телефон);

– *договоры* (код договора, код клиента, код дилера, дата заключения договора, марка автомобиля, фото автомобиля, дата выпуска, пробег, дата продажи, цена продажи, размер комиссионных, примечание).

В создаваемой информационной системе необходимо обеспечить ввод и редактирование данных. Кроме того, необходимо выдавать информацию о клиентах и предлагаемых ими автомобилях, а также информацию о деятельности дилеров (с перечислением договоров) и клиентах, которые они обслуживают. Могут быть выполнены разнообразные запросы, например:

- посчитать количество договоров, заключенных с каждым клиентом;
- посчитать количество договоров, обслуживаемых каждым дилером;
- выдать некоторую информацию (например: данные дилера, дата заключения договора, данные клиента, отметка о продаже) обо всех договорах, договорах за некоторый промежуток времени или договорах, удовлетворяющих определенному условию.

### **Вариант 9. Пассажирское автопредприятие**

Муниципальное автопредприятие осуществляет пассажирские перевозки на внутригородских маршрутах.

Автопредприятие имеет парк автобусов, которые работают на определенных маршрутах. Работу автопредприятия обеспечивает персонал предприятия, который можно разделить по категориям занимаемых должностей на администрацию, инженерно-технический персонал и персонал, обслуживающий маршруты (водители, кондукторы). Выезжая на маршрут, водитель автобуса получает маршрутный лист (или путевой лист), содержащий данные об автобусе, маршруте, режиме работы, водителе, кондукторе.

Автопредприятие описывается данными, которые могут быть сгруппированы следующим образом:

- *автобусы* (бортовой номер автобуса, гос. номер автобуса, марка, год выпуска, пробег);
- *маршруты* (номер маршрута, маршрут, протяженность маршрута в км, среднее время одного рейса, плановое количество рейсов за смену);
- *личные данные персонала* (табельный номер, ФИО сотрудника, дата рождения, домашний адрес, домашний телефон, рабочий телефон);
- *учетные данные персонала* (табельный номер, категория, должность, дата приема на работу, номер автобуса (для водителей и кондукторов));
- *маршрутные листы* (номер маршрута, бортовой номер автобуса, дата, количество выполненных рейсов, водитель, кондуктор).

Возможно, что на реальном автопредприятии учитывается большее количество данных, однако исполнителю задания можно ограничиться перечисленными. При желании в учетных данных персонала можно ввести данные для оплаты труда, если предполагается автоматизация начисления



зарплаты. В маршрутных листах можно ввести плановую и фактическую выручки за смену соответственно.

Создаваемая информационная система, прежде всего, должна обеспечивать ввод и редактирование данных в удобной для пользователя форме. Кроме того, можно сформулировать разнообразные запросы как по кадровому составу предприятия, так и по характеристике и техническому обеспечению маршрутов. Например:

- выдать полную или частичную информацию по персоналу;
- выдать полную или частичную информацию по автобусному парку;
- выдать полную или частичную информацию по маршрутам.

В частности, запросы могут быть такими:

- выдать список сотрудников администрации с указанием должности;
- на определенную дату для всех номеров маршрутов выдать информацию о количестве автобусов, обслуживающих каждый маршрут;
- по каждому номеру маршрута и дате (параметры запроса) выдать информацию об автобусах, обслуживающих маршрут: бортовой номер, марка, гос. номер автобуса.

Могут решаться и другие задачи. Например, по итогам работы за месяц посчитать количество рейсов, выполненных каждым автобусом или на каждом маршруте. По итогам работы за месяц посчитать количество смен, отработанных каждым водителем и кондуктором.

### **Вариант 10. Междугородные пассажирские перевозки**

Рассмотрим автовокзал, который занимается обслуживанием и учетом пассажиров на междугородных автобусных маршрутах. На автовокзале имеется расписание движения автобусов, содержащее информацию о маршрутах и рейсах. Кроме того, на автовокзале имеется справочное бюро, в котором можно получить информацию о наличии мест на определенный рейс конкретной даты. И, наконец, на автовокзале есть кассы, в которых пассажир может приобрести билет. Кассы начинают предварительную продажу билетов за определенный промежуток времени до дня отправления автобуса (например, за 10 дней).

Необходимо построить такую базу данных, в которой хранится информация как о технических характеристиках маршрутов, содержащаяся в расписании, так и информация о наличии мест на рейсы, и информация о пассажирах, купивших билеты на определенный рейс.

Ниже предлагается вариант организации информации о рейсах и пассажирах (однако исполнитель задания может предложить собственный вариант организации данных).

Администратор базы данных к началу продажи билетов на рейс, т.е., например, за 10 дней до дня отправления, создает таблицу, соответствующую рейсу и дате отправления. Назовем эту таблицу схемой рейса, в мо-

мент создания схема рейса содержит только номера мест, а в процессе продажи билетов схема будет заполняться информацией о пассажире, которому продан билет на соответствующее место. Причем, в схеме может находиться собственно информация о пассажире (ФИО) или ссылка на нее.

После того как рейс выполнен, таблица со схемой рейса удаляется, но перед этим список пассажиров этого рейса можно отправить в архив, который хранится в течение определенного срока, а информацию о рейсе – в таблицу выполненных рейсов. Если по какой-либо причине рейс отменяется, то информация об этом помещается в таблицу отмененных рейсов. При описанной организации данные можно сгруппировать следующим образом:

- *расписание рейсов* (номер рейса, маршрут, тип автобуса, расстояние в км, дни отправления, время отправления, время прибытия, цена билета);
- *схема рейса* (номер места, информация о пассажире: ФИО);
- *адреса схем рейсов* (номер рейса, дата отправления, ссылка на схему рейса);
- *архив пассажиров* (номер рейса, дата отправления, номер места, фамилия, имя, отчество);
- *выполненные рейсы* (номер рейса, дата отправления, количество проданных мест, полученная сумма);
- *отмененные рейсы* (номер рейса, дата, причина отмены рейса).

В создаваемой системе, прежде всего, необходимо обеспечить ведение данных: организацию таблиц для схем рейсов и ссылок на них, ввод и редактирование данных в таблицах, архивирование данных, удаление таблиц.

Кроме того, в рассматриваемой задаче представляют интерес запросы. Например, следующие:

- наличие свободных мест на рейс;
- количество пассажиров уже выполненного рейса, доходность рейса;
- список всех пассажиров определенного рейса (выполненного или того, на который идет продажа билетов);
- определить, покупал ли билет человек с заданной фамилией и, если покупал, то на какой рейс.

Количественные данные рассматриваемой задачи позволяют также вести их статистический учет, можно для наглядности использовать графическое представление данных. Например, определять количество перевезенных пассажиров и объем перевозок (в денежном выражении) по дням, по месяцам в целом по всем направлениям или по определенному маршруту.

### **Вариант 11. Агентство по продаже авиабилетов**

Агентство занимается продажей авиабилетов на различные рейсы, ведет учет проданных билетов и учет пассажиров, купивших билеты, по-

этому возникает потребность в хранении и обработке данных, сгруппированных следующим образом:

- информация о расписании рейсов (номер рейса, тип самолета, пункт отправления, пункт назначения, дата вылета, время вылета, время полета, цена билета);

- информация о свободных местах на рейс (номер рейса, дата вылета, общее количество мест, количество свободных мест);

- информация о пассажирах, заказавших билет (ФИО, предъявленный документ, его серия и номер, номер рейса, дата вылета).

Особенность данной задачи состоит в том, что информация в базе данных может использоваться как пассажирами (например, для получения сведений о расписании и наличии свободных мест на рейс), так и служащими агентства: кассирами и диспетчерами (администраторами). То есть выделяются три группы пользователей базы данных: пассажир, кассир, диспетчер. В связи с этим возникает проблема разграничения прав доступа этих пользователей к информации базы данных для защиты ее от несанкционированного использования.

Другая особенность рассматриваемой задачи: удаление в архив информации о выполненном рейсе и ввод данных о рейсе с таким же номером на очередную дату, отстоящую от даты удаленного в архив рейса на определенный промежуток, например, на несколько дней вперед.

### **Вариант 12. Гостиница**

Рассмотрим возможную модель организации размещения и учета проживающих в некоторой гостинице граждан (клиентов).

Предполагаемая гостиница располагает номерами с разным уровнем сервиса, комфорта и, соответственно, оплаты. Одной из характеристик номера является его тип. Предположим, что существуют следующие типы номеров: люкс – многокомнатный номер с высоким уровнем сервиса, комфорта и обслуживания; полулюкс – номер меньшей, чем люкс, площади, но с достаточным уровнем сервиса и комфорта; одноместный или двухместный номер с минимальным уровнем сервиса; многоместный номер, также с некоторым уровнем сервиса. Стоимость для номеров типа люкс и полулюкс устанавливается как стоимость всего номера (в сутки), независимо от количества проживающих в номере. Стоимость проживания в одно-, двух- и многоместных номерах устанавливается для одного человека (в сутки). Номера и места в номерах могут бронироваться. При наличии телефона в номере пользование междугородным телефоном оплачивается отдельно по фактическим счетам.

Все прибывающие и размещаемые в гостинице граждане при вселении должны заполнить карточку регистрации. Кроме того, для расчетов с клиентами администрация гостиницы заводит расчетные карточки, кото-

рые содержат данные о размещении клиента, оказанных услугах и всех расчетах с ним. Для номеров люкс и полулюкс расчетная карточка может заводиться только для одного клиента, оплачивающего номер.

Любой номер гостиницы имеет номер, по которому ведется учет проживающих в гостинице. Это свойство номера в последующих описаниях будем называть номером комнаты (независимо от фактического количества комнат в номере).

При выбытии клиента данные о нем сохраняются в архиве.

Можно предположить использование следующих свойств (данных) о номерах гостиницы и ее клиентах, которые можно сгруппировать описанным ниже способом:

– *сведения о номерах люкс и полулюкс* (номер комнаты, тип номера, занят/свободен, количество комнат, этаж, телефон, стоимость номера в сутки, сведения о бронировании, количество фактически проживающих);

– *сведения о прочих номерах* (номер комнаты, тип номера, количество мест, этаж, телефон, стоимость проживания одного человека в сутки, количество свободных мест);

– *карточки регистрации* (номер регистрации клиента, номер комнаты, дата прибытия, ФИО, предъявленный документ, серия и номер документа, дата рождения, пол, домашний адрес, домашний телефон);

– *расчетные карточки* (номер регистрации клиента, номер комнаты, дата и время прибытия, оплата брони, предполагаемая дата убытия, количество оплаченных дней, сумма оплаты, окончательный расчет);

– *архив* (номер регистрации клиента, ФИО, документ, серия и номер, дата рождения, пол, домашний адрес, домашний телефон, номер комнаты, дата прибытия, дата убытия).

По усмотрению исполнителя можно расширить или, наоборот, уменьшить количество свойств (данных) рассматриваемой задачи.

Создаваемая информационная система предназначена для администрации гостиницы, которая на основании информации о номерах занимается размещением клиентов в соответствии с их запросами. При выбытии клиента информация о номере, в котором он проживал, должна обновляться, а информация о клиенте должна удаляться из рабочих таблиц (карточки регистрации клиентов и карточки учета) и помещаться в архивную таблицу.

Кроме перечисленных задач ведения данных в системе могут решаться задачи поиска, например, поиск номера или места в номере в соответствии с некоторыми критериями поиска. Другая задача поиска – это поиск клиента, проживающего в гостинице в данный момент или проживавшего в ней ранее.

Кроме того, в рассматриваемой задаче можно организовать статистическую обработку данных; например, учет количества проживающих в гостинице по определенным периодам.

## Перечень заданий по реляционной алгебре для организации самостоятельной работы студентов

**Задание 1.** Составить и заполнить таблицы (3, 4, 5, 6, 9, 11, 13) данными, полученными в результате действия операций реляционной алгебры.

**Пример 1.** Пусть даны два отношения  $A$  и  $B$  с информацией о сотрудниках.

*Таблица 1 Отношение A*

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

*Таблица 2 Отношение B*

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Пушников	2500
4	Сидоров	3000

Объединение отношений  $A$  и  $B$  будет иметь вид:

*Таблица 3 Отношение A UNION B*

Пересечение имеет вид:

*Таблица 4 Отношение A INTERSECT B*

Вычитание имеет вид:

*Таблица 5 Отношение A MINUS B*

*Таблица 6 Отношение B MINUS A*

**Пример 2.** Пусть даны два отношения  $A$  и  $B$  с информацией о поставщиках и деталях.

*Таблица 7 Отношение A (Поставщики)*

Номер поставщика	Наименование поставщика
1	Иванов
2	Петров
3	Сидоров

Таблица 8 Отношение B (Детали)

Номер детали	Наименование детали
1	Болт
2	Гайка
3	Винт

Декартово произведение отношений A и B будет иметь вид:

Таблица 9 Отношение A TIMES B

**Пример 3.** Пусть дано отношение A с информацией о сотрудниках.

Таблица 10 Отношение A

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Результат выборки  $A \text{ WHERE Зарплата} < 3000$  будет иметь вид:

Таблица 11 Отношение A WHERE Зарплата < 3000

**Пример 4.** Пусть дано отношение A с информацией о поставщиках, включающих наименование и месторасположение.

Таблица 12 Отношение A (Поставщики)

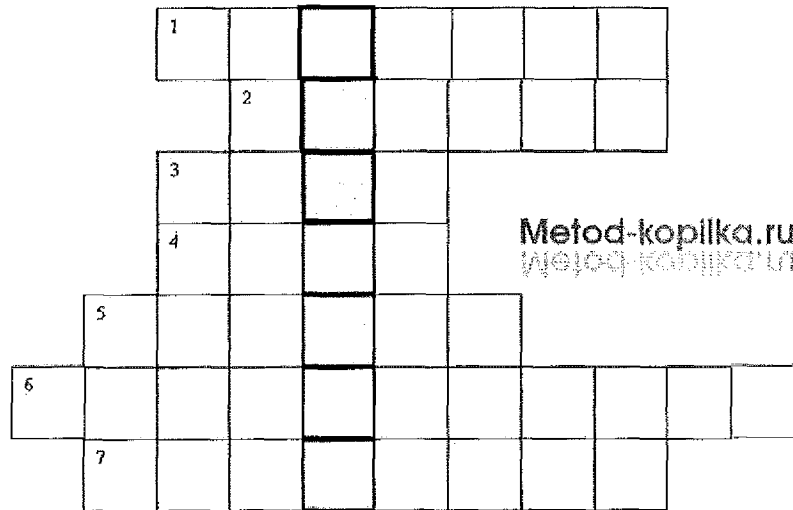
Номер поставщика	Наименование поставщика	Город поставщика
1	Иванов	Уфа
2	Петров	Москва
3	Сидоров	Москва
4	Сидоров	Челябинск

Проекция  $A[\text{Город поставщика}]$  будет иметь вид:

Таблица 13 Отношение A[Город поставщика]

## Кроссворд по теме «Базы данных»

Перенесите кроссворд в тетрадь и разгадайте его.



1. Структура БД, в которой связи между данными свободные.
2. Средство извлечения данных из БД.
3. Набор программ, с помощью которых производится работа с базой данных.
4. Столбец в табличной БД.
5. Строка в табличной БД.
6. Структура БД, в которой данные строятся по принципу взаимосвязанных таблиц.
7. Структура, в которой одно данное является главным, все нижележащие – подчиненными.

Сформулируйте определение получившемуся по вертикали понятию с точки зрения баз данных.

*Ответы:* сетевая, запрос, СУБД, поле, запись, реляционная, иерархия.  
Таблица – основной объект реляционной БД.

### Тестовые задания

Продолжите предложения.

1. Информационная система – это...

2. База данных – это...

3. Перечислите виды информационных систем по способу организации:

4. Приведите классификацию баз данных.

5. Перечислите основные объекты баз данных:

6. Что является основным объектом в реляционной базе данных?

7. Приведите примеры СУБД.

### Тест № 1

1. База данных – это:

- произвольный набор информации;
- специальным образом организованная и хранящаяся на внешнем носителе совокупность взаимосвязанных данных о некотором объекте;
- совокупность программ для хранения и обработки больших массивов информации;
- интерфейс, поддерживающий наполнение и манипулирование данными;
- компьютерная программа, позволяющая в некоторой предметной области делать выводы, сопоставимые с выводами человека-эксперта.

2. Система управления базами данных представляет собой программный продукт, входящий в состав:

- системного программного обеспечения;



- операционной системы;
- систем программирования;
- уникального программного обеспечения;
- прикладного программного обеспечения.

3. *Примером иерархической базы данных является:*

- каталог файлов, хранимых на диске;
- расписание поездов;
- электронная таблица;
- экспертная система;
- страница классного журнала.

4. *В записи файла реляционной базы данных (БД) может содержаться:*

- только логические величины;
- только текстовая информация;
- исключительно однородная информация (данные только одного типа);
- исключительно числовая информация;
- неоднородная информация (данные разных типов).

5. *Предположим, что некоторая база данных содержит поля ФАМИЛИЯ, ГОД РОЖДЕНИЯ, ДОХОД. При поиске по условию ГОД РОЖДЕНИЯ > 1958 AND ДОХОД < 3500 будут найдены фамилии лиц:*

- имеющих доход менее 3500, или тех, кто родился в 1958 году и позже;
- имеющих доход менее 3500 и родившихся в 1958 году и позже;
- имеющих доход менее 3500 и тех, кто родился в 1958 году;
- имеющих доход менее 3500 и родившихся в 1959 году и позже;
- имеющих доход не менее 3500 и старше тех, кто родился в 1958 году.

6. *Предположим, что некоторая база данных описывается следующим перечнем записей:*

- 1 Иванов, 1956, 2400
- 2 Сидоров, 1957, 5300
- 3 Петров, 1956, 3600
- 4 Козлов, 1952, 1200

Какие из записей этой БД поменяются местами при сортировке по возрастанию, произведенной по первому полю:

- 1 и 3;
- 2 и 3;
- 2 и 4;
- 3 и 4;
- 1 и 4?

7. Сетевая база данных предполагает такую организацию данных, при которой:

- помимо вертикальных иерархических связей (между данными) существуют и горизонтальные;
- связи между данными отражаются в виде совокупности нескольких таблиц;
- связи между данными отражаются в виде таблицы;
- между ними имеются исключительно вертикальные связи;
- связи между данными описываются в виде дерева.

8. Структура файла реляционной базы данных (БД) меняется:

- при добавлении одной или нескольких записей;
- при удалении диапазона записей;
- при изменении любой записи;
- при уничтожении всех записей;
- при удалении любого поля.

Ключ к тесту: 1-2, 2-5, 3-1, 4-5, 5-4, 6-3, 7-1, 8-5.

## Тест № 2

1. В иерархической базе данных совокупность данных и связей между ними описывается:

- a) таблицей;
- b) сетевой схемой;
- c) древовидной структурой;
- d) совокупностью таблиц.

2. В реляционной базе данных информация организована в виде

- a) сети;
- b) иерархической структуры;
- c) файла;

- d) дерева;
- e) прямоугольной таблицы.

3. *Основным элементом реляционной БД является:*

- a) поле,    b) форма,    c) таблица,    d) запись.

4. *База данных представлена в табличной форме. Запись образует*

- a) поле в таблице;
- b) имя поля;
- c) строку в таблице;
- d) ячейку.

5. *В поле реляционной базы данных могут быть записаны:*

- a) только номера записей;
- b) как числовые, так и текстовые данные одновременно;
- c) данные только одного типа;
- d) только время создания записей.

6. *В число основных функций СУБД не входит:*

- a) определение того, какая именной информация (о чем) будет храниться в базе данных;
- b) создание структуры файла базы данных;
- c) первичный ввод, пополнение и редактирование данных;
- d) поиск и сортировка данных.

7. *База данных – это...*

- a) специальным образом организованная и хранящаяся на внешнем носителе совокупность взаимосвязанных данных о некотором объекте;
- b) совокупность программ для хранения и обработки больших массивов информации;
- c) интерфейс, поддерживающий наполнение и манипулирование данными.

8. *Примером иерархической базы данных является:*

- a) страница классного журнала;
- b) каталог файлов, хранимых на диске;
- c) расписание поездов;
- d) электронная таблица.

9. *Наиболее точным аналогом реляционной БД может служить:*

- a) неупорядоченное множество данных;
- b) вектор;

- с) генеалогическое дерево;
- д) двумерная таблица.

*10. Структура реляционной БД меняется при удалении:*

- а) одного из полей;
- б) одной записи;
- с) нескольких записей;
- д) всех записей.

*11. Система управления базами данных (СУБД) – это:*

- а) программная система, поддерживающая наполнение и манипулирование данными в файлах баз данных;
- б) набор программ, обеспечивающий работу всех аппаратных устройств компьютера и доступ пользователя к ним;
- с) прикладная программа для обработки текстов и различных документов;
- д) оболочка операционной системы, позволяющая более комфортно работать с файлами.

### **Тест № 3**

*1. Модель базы данных может быть:*

- а) иерархическая;
- б) сетевая;
- в) системная;
- г) реляционная.

*2. Объектом действий в базе данных являются:*

- а) поле;
- б) формула;
- в) запись;
- г) символ.

*3. СУБД – это программное средство для:*

- а) обеспечения работы с таблицами чисел;
- б) управления большими информационными массивами;
- в) хранения файлов;
- г) создания и редактирования текстов.

*4. База данных – это...*

- а) набор взаимосвязанных модулей, обеспечивающих автоматизацию многих видов деятельности;
- б) таблица, позволяющая хранить и обрабатывать информацию;

- в) интегрированная совокупность данных, предназначенная для хранения и функционального использования;
- г) прикладная программа для обработки информации пользователя.

5. *Объект базы данных, который создается для пользователя при просмотре, вводе и редактировании данных – это...*

- а) таблица;
- б) форма;
- в) запрос;
- г) отчет.

6. *Результат обращения пользователя к СУБД для поиска данных, добавления, удаления и обновления записей – это...*

- а) таблица;
- б) форма;
- в) запрос;
- г) отчет.

7. *Над записями базы данных можно проводить следующие операции:*

- а) сортировку;
- б) выборку;
- в) индексирование;
- г) удаление.

9. *Существуют следующие типы запросов:*

- а) запрос-выборка;
- б) запрос на добавление записей;
- в) запрос на объединение записей;
- г) запрос на удаление записей.

10. *Отчет, создаваемый на базе запроса, позволяет:*

- а) представить данные в удобной для чтения и анализа форме;
- б) сгруппировать записи с вычислением итоговых и средних значений;
- в) включить в отчет и напечатать графические объекты;
- г) проводить сортировку и выборку записей.

#### **Тест № 4** **Вариант 1**

1. *Базы данных – это:*

- А) информационные структуры, хранящиеся во внешней памяти ЭВМ;

Б) программные средства, позволяющие организовывать информацию в виде таблиц;

В) программные средства обрабатывающие табличные данные;

Г) программные средства, осуществляющие поиск информации.

2. В реляционной БД информация организована в виде

А) файла;

Б) дерева;

В) прямоугольной таблицы;

Г) иерархической структуры.

3. БД содержит информацию об учениках школы: фамилия, класс, балл за тест, балл за практическое задание, общее количество баллов. Какого типа должно быть поле **ОБЩЕЕ КОЛИЧЕСТВО БАЛЛОВ**?

А) символьное;

Б) логическое;

В) числовое;

Г) дата.

Реляционная БД задана таблицей:

№	ФИО	ПОЛ	ВОЗРАСТ	КЛУБ	СПОРТ
1	Панько Л.П.	жен	22	Спарта	футбол
2	Арбузов А.А.	муж	20	Динамо	лыжи
3	Жиганова П.Н.	жен	19	Ротор	футбол
4	Седов О.Л.	муж	19	Спарта	биатлон
5	Багаев С.И.	муж	23	Звезда	лыжи

4. Какие записи будут выбраны по условию:  
*возраст > 20?*

А) 1,2,4;

Б) 3;

В) 1,2,5;

Г) 1,5.

5. Какие записи будут выбраны по условию:  
*спорт = лыжи И пол = муж или возраст < 20?*

А) 2,5;

Б) 2,3,4,5;

В) 1,3,5;

Г) 1,2,3.

6. Какие записи будут выбраны по условию:  
*спорт = лыжи или не пол = муж?*

- А) 2,5;
- Б) 2,3,4,5;
- В) 1,3,5;
- Г) 1,2,3,5.

7. Определить типы всех полей в данной таблице.

### Вариант 2

1. Модель содержит информации...

- а) столько же, сколько и моделируемый объект;
- б) меньше, чем моделируемый объект;
- в) больше, чем моделируемый объект;
- г) не содержит информации.

2. Главным ключом в базах данных называется:

- А) все записи БД;
- Б) поле значение, которого не повторяется у разных записей;
- В) поле значение, которого повторяется у разных записей;
- Г) поле логического типа.

3. Система управления базами данных – это...

- А) комплекс аппаратно-программных средств для работы с БД;
  - Б) организованная совокупность данных предназначенных для длительного хранения во внешней памяти ЭВМ;
  - В) БД с табличной формой организации данных;
  - Г) программное обеспечение, предназначенное для работы с БД.
- Реляционная БД задана таблицей:

№	Ф.И.О.	ПОЛ	ВОЗРАСТ	КЛУБ	СПОРТ
1	Панько Л.П.	жен	22	Спарта	футбол
2	Арбузов А.А.	муж	20	Динамо	лыжи
3	Жиганова П.Н.	жен	19	Ротор	футбол
4	Седов О.Л.	муж	19	Спарта	биатлон
5	Багаев С.И.	муж	23	Звезда	лыжи

4. Какие записи будут выбраны по условию:  
*возраст <= 20?*

- А) 1,2;
- Б) 2,3,4;
- В) 4;
- Г) 3,4.

5. Какие записи будут выбраны по условию:  
*спорт = лыжи И пол = муж или возраст > 20?*

- А) 2,5;
- Б) 2,3,4,5;
- В) 1,2,5;
- Г) 1,2,3.

6. Какие записи будут выбраны по условию:  
*возраст > 19 и не пол = жен?*

- А) 2,5;
- Б) 2,3,4,5;
- В) 1,2,5;
- Г) 1,2,3.

7. Определить типы всех полей в данной таблице.

### Тест № 5

1. Представлена база данных «Волшебные страны»

СТРАНА	НАСЕЛЕНИЕ	ПЛОЩАДЬ
НАРНИЯ	48	46,9
БУЯН	156	47,7
ОЗ	158	95,3
ГУСТОЙ ЛЕС	99	53,5
ЗАЗЕРКАЛЬЕ	511	76,2

После проведения сортировки сведения о ГУСТОМ ЛЕСЕ переместились на две строки вверх. Сортировка проводилась в порядке:

- 1) возрастания по полю СТРАНА;
- 2) убывания по полю ПЛОЩАДЬ;
- 3) возрастания по полю ПЛОЩАДЬ;
- 4) убывания по полю СТРАНА;
- 5) возрастания по полю НАСЕЛЕНИЕ.

2. Представлена база данных «Классы школы»

Класс	Кол. учеников	Староста
9а	27	Колесник
10а	26	Андреев
8б	30	Чебаев
11а	18	Раков
10б	24	Крупинский



А) после сортировки в порядке возрастания по полю «Класс» сведения о 9а переместятся на:

- 1) 4 строки вниз;
- 2) 1 строку вниз;
- 3) 3 строки вниз;
- 4) 2 строки вниз;
- 5) 0 строк.

Б) после сортировки в порядке возрастания по полю «Класс» сведения о 8б переместятся на:

- 1) 1 строку вверх;
- 2) 1 строку вниз;
- 3) 2 строки вверх;
- 4) 2 строки вниз;
- 5) 0 строк.

3. Запросу *Серия = Для чайников* или *Год\_издания > = 1996* в базе данных отвечают записи

№	Автор	Серия	Наименование	Год издания	Кол стр
1	Уолш Р.	Для начинающих	Windows 95	1996	128
2	Султанов И.	Для пользователей	Энциклопедия Delphi	1997	300
3	Кирсанов Д.	Для чайников	Word 7.0	1996	236
4	Визе М.	Компьютер для носорога	Access 2.0	1994	255

- 1) только 3;
- 2) только 1,2,3;
- 3) только 1,3;
- 4) только 1;
- 5) только 2,3.

4. Запрос к базе данных с полями *Фамилия*, *Год\_рождения*, *Класс*, *Оценка* для вывода списка учеников 10 класса, 1981 года рождения, имеющих оценки 4 или 5, содержит выражение:

- 1)  $Класс > 10$  и  $Оценка = 4$  и  $Год\_рождения = 1981$ ;
- 2)  $Класс = 10$  или  $Оценка > 4$  или  $Год\_рождения = 1981$ ;
- 3)  $Оценка \geq 4$  и  $Год\_рождения = 1981$  и  $Класс = 10$ ;
- 4)  $Оценка \geq 4$  и  $Год\_рождения > 1980$  и  $Класс = 10$ ;
- 5)  $Класс = 10$  и  $Оценка > 4$  или  $Год\_рождения = 1981$ .

5. Представлена база данных «Телефонный справочник»

Фамилия И.О.	Телефон
Иванов И.И.	234-56-98
Иванова А.П.	235-60-07
Кедров А.К.	435-88-78
Иванов А.К.	568-98-00
Иванников П.П.	384-15-15

А) После проведения сортировки по полю ФИО в порядке возрастания запись, содержащая номер телефона 568-98-00, переместится на

- 1) 1 строку вверх;
- 2) 2 строки вверх;
- 3) 1 строку вниз;
- 4) не переместится;
- 5) 3 строки вверх.

Б) После проведения сортировки по полю ФИО в порядке убывания запись, содержащая сведения о телефоне 384-15-15, переместится на

- 1) 1 строку вверх;
- 2) 2 строки вверх;
- 3) 3 строки вверх;
- 4) 4 строки вверх;
- 5) не переместится.

6. По какому полю базы данных упорядочены записи?

№	Фамилия	Улица	Дом	Квартира	№ телефона
1	Иванов	Советская	42	15	258-36-19
2	Петров	Пушкина	15/2	366	366-56-98
3	Сидоров	Гоголя	35	25	255-41-88
4	Кузьмин	Гафури	69	38	564-89-71

- 1) фамилия;
- 2) улица;
- 3) № телефона.

7. Запросу (Физика = 5 или Информатика = 5) и Математика = 5 и Первая буква (Фамилия) = «И» в базе данных отвечают записи

№	Фамилия	Имя	Математика	Физика	Информатика
1	Ипатов	Егор	4	4	5
2	Киреев	Алексей	5	5	5
3	Илюхин	Сергей	5	4	4
4	Ивашкина	Татьяна	5	5	4

- 1) 1,3,4;
- 2) 4;
- 3) 3,4;
- 4) 2,4;
- 5) 2,3.

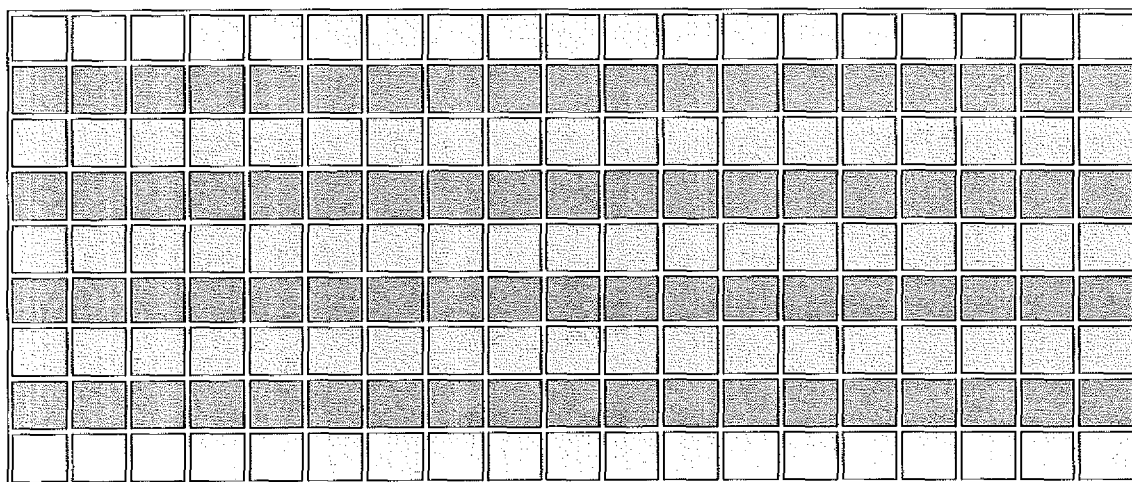
8. База данных «Зоопарк» содержит в полях **Животное**, **Тип**, **Количество** сведения о наименовании животного, типа его рациона (возможны два типа: П – плотоядное или Т – травоядное) и количество пищи в день. Для получения информации о животных, съедающих в день от 5 до 10 кг мясных продуктов, запрос имеет вид:

- 1) Животное = (Тип = П или Количество = 5);
- 2) Тип <> Т и  $10 \leq \text{Количество} \leq 5$ ;
- 3) Тип = П и ( $\text{Количество} \geq 5$  и  $\text{Количество} \leq 10$ );
- 4) Тип <> Т и  $10 \leq \text{Количество} < 5$ ;
- 5) Тип = П и ( $\text{Количество} \geq 5$  и  $\text{Количество} \geq 10$ ).

### Венгерский кроссворд

Венгерский кроссворд представляет собой поле из клеток, в которые уже вписаны буквы ответов. В цепочке клеток, составляющих каждый ответ, соседние клетки должны соприкасаться сторонами. Слова-ответы не пересекаются и не имеют общих клеток с другими словами. После разгадывания венгерского кроссворда на поле остаются «лишние» буквы — из них (по порядку или анаграммой) составляется общий ответ на головоломку.

*Венгерский кроссворд по теме  
«Базы данных. Модели и таблицы»*



- 1, 2. Графические информационные модели (5, 6) (количество букв в слове).
3. Информационная модель – прямоугольная (7).
4. Числовая прямоугольная таблица (7).

5. Матрица из нулей и единиц называется ... матрицей (8).
6. База ... – организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ и постоянного применения (6).
7. БД, в которой содержатся краткие сведения об описываемых объектах, представленные в строго определенном формате (16).
8. База данных с табличной формой организации (11).
9. Строка таблицы (6).
10. Столбец таблицы (4).
- 11, 12, 13. Тип поля (8, 10, 4).
14. Система управления базами данных (4).
15. Таблица, содержащая интересующие пользователя сведения, извлеченные из базы данных (7).
16. Знак отношения (5).
17. Логическая операция (9).
18. Значение логического выражения (6).
19. Логическая операция (8).
20. Значение логического выражения (4).
21. Режим работы, в котором определяется структура БД (8).
22. Процесс упорядочения записей в таблице по возрастанию или убыванию значений какого-нибудь поля (10).
23. Логические выражения – условия ... в командах СУДБ (6).

*Ответы:*

- 1, 2. Карта, чертеж.
3. Таблица.
4. Матрица.
5. Двоичной.
6. Данных.
7. Фактографическая.
8. Реляционная.
9. Запись.
10. Поле.
- 11, 12, 13. Числовой, символьный, дата.
14. СУБД.
15. Справка.
16. Равно.
17. Умножение.
18. Истина.
19. Сложение.
20. Ложь.
21. Создание.
22. Сортировка.

23. Поиска.

Ключевое слово: модель.

### Чайнворд по теме «Базы данных»

Найдите 13 терминов по теме, первые буквы слов пропущены, слова записаны по вертикали и горизонтали. Дайте характеристику зашифрованным терминам. Первые буквы – т, з, д, с, у, п, к, и, т, з, о, ф, о.

	Н	И	К			Л	Т
Н	Ь	Л	А	У	Н	И	Е
О	Е			Б	А		Ч
	Е	А	А	Д	Л	О	Т
Л	О	Б	П	И	Е		
Ю	В	Л	А	С	Р	О	С
Ч	Е	И	Ц	Ь	М	Ы	О
	Б	Ъ			М		Р
Т	К	Е	И	П	Я	А	П

### Правильная таблица

у	Н	И	К	с	д	Л	Т
Н	Ь	Л	А	У	Н	И	Е
О	Е	т	з	Б	А	п	Ч
к	Е	А	А	Д	Л	О	Т
Л	О	Б	П	И	Е	ф	о
Ю	В	Л	А	С	Р	О	С
Ч	Е	И	Ц	Ь	М	Ы	О
о	Б	Ъ	т	п	М	з	Р
Т	К	Е	И	П	Я	А	П

Ответы: уникальное, таблица, запись, СУБД, длина, поле, отчет, формы, ключ, объект, тип, имя, запрос.

### Работа с программой командной строки sqlite.exe

**Задание 1.** В программе sqlite создать таблицу tbl с названием полей: id, name, mark и создать перечисленные ниже запросы к базе данных.

id	name	mark
1	Иванов	4
2	Петров	5
3	Васильев	4

Создание таблицы:

```
create table tbl (id integer, name varchar(30), mark integer);
```

Добавление записей в таблицу:

```
insert into tbl (id, name, mark) values (1, 'Иванов', 4);
```

```
insert into tbl (id, name, mark) values (2, 'Петров', 5);
```

```
insert into tbl (id, name, mark) values (2, 'Петров', 5);
```

Просмотр всей таблицы:

```
Select * from tbl;
```

Сортировка по полю имя:

```
Select * from tbl order by name;
```

Определение количества записей:

```
Select count(*) from tbl;
```

Среднее арифметическое оценок:

```
Select avg(mark) from tbl;
```

Количество 4:

```
Select count(mark) from tbl where mark=4;
```

Удаление записей, где оценка 5:

```
Delete from tbl where mark=5;
```

```
Select * from tbl;
```

Увеличить на 1 оценку:

```
Update tbl set mark=mark+1;
```

```
Select * from tbl;
```

### **Программа командной строки sqlite.exe**

Программа командной строки «sqlite.exe» предназначена для создания и работы с базой данных, используя язык запросов SQL и специальные команды. Вся работа, а именно ввод команд и получение результатов выполнения запросов происходит в консольном окне WINDOWS.

Определим специальные команды управления программы командной строки «sqlite.exe». Все специальные команды начинаются с точки, если команда введена без ведущей точки она будет интерпретирована как SQL запрос. Ниже представлен список всех специальных команд и дано краткое их описание.

Основные команды работы с `sqlite.exe`

Команда	Краткое описание
<code>.databases</code>	Вывод списка имен всех подключенных в текущем сеансе баз данных и соответствующих им файлов.
<code>.dump ?TABLES?:</code>	Дамп всех SQL инструкций использованных в создании БД или отдельной таблицы в текстовом формате
<code>.exit</code>	Выход из программы
<code>.explain ON OFF</code>	Управляет режимом вывода виртуальных машинных команд. Используется при выполнении SQL запроса EXPLAIN.
<code>.header(s) ON OFF</code>	ВКЛ   ВЫКЛ показ заголовков столбцов
<code>.import FILE TABLE</code>	Импорт данных из файла FILE в таблицу TABLE
<code>.indices TABLE</code>	Показывает имена всех индексов таблицы
<code>.mode MODE</code>	Установка режима вывода: <code>line(s)</code> , <code>column(s)</code> , <code>insert</code> , <code>list</code> , <code>html</code> и других
<code>.nullvalue STRING</code>	Напечатает строку STRING вместо NULL данных при выводе SQL запроса SELECT
<code>.output FILENAME</code>	Послать весь вывод в файл FILENAME
<code>.output stdout</code>	Послать весь вывод на экран
<code>.prompt MAIN COTINUE</code>	Изменить стандартную строку подсказки
<code>.quit</code>	Выход из программы
<code>.read FILENAME</code>	Выполнение SQL инструкций из файла FILENAME
<code>.schema ?TABLE?</code>	Покажет текст SQL инструкции CREATE для всех таблиц или указанной таблицы
<code>.separator STRING</code>	Изменить строку разделитель колонок, используется при выводе SQL запроса SELECT и команды <code>.import</code>
<code>.show</code>	Показать значения установленных переменных
<code>.tables ?PATTERN?</code>	Вывод списка имен таблиц БД (возможно по шаблону)
<code>.timeout MS</code>	Блокирование открытия таблиц на число миллисекунд MS
<code>.width NUM NUM :</code>	Установка ширины столбцов при выводе в режиме <code>column</code>

Подробно рассмотрим некоторые из этих команд. Предварительно создадим базу данных, на которой мы и рассмотрим специальные коман-

ды. Из командной строки запустите консольную программу «sqlite3.exe» введя следующее:

```
sqlite3 exam.db
SQLite version 3.1.3
Enter ".help" for instructions
```

где exam.db имя создаваемой базы данных.

Создадим таблицу tbl1, с помощью SQL инструкции CREATE TABLE.

```
sqlite> CREATE TABLE tbl1 (one INTEGER PRIMARY KEY
AUTOINCREMENT, two VARCHAR(50), three REAL);
```

Вставим несколько записей в данную таблицу. Отметим, что в инструкции создания таблицы для каждого поля мы задали тип данных для его значений, этот тип данных игнорируется программой «SQLite». Система управления базой данных "SQLite" довольно свободно относится к определению типов данных, что не характерно для стандарта языка SQL. Типизация данных в «SQLite» основывается на так называемом принципе «typeless», означающий игнорирование информации о типе данных в определении столбцов таблицы при ее создании с помощью инструкции CREATE TABLE. И все же лучше включать имена типов столбцов при объявлении таблиц, это повысит переносимость вашей базы данных. Имеется одно исключение из принципа «typeless» – это столбец типа INTEGER PRIMARY KEY (INTEGER не INT; INT PRIMARY KEY - удовлетворяет принципу «typeless»). Столбец с типом INTEGER PRIMARY KEY должен содержать 32-х битное целое число, попытка записать данные любого другого типа в столбец этого типа приведет к ошибке. Этот тип обычно используется перед спецификатором столбца AUTOINCREMENT, столбец с типом INTEGER PRIMARY KEY AUTOINCREMENT является ключевым столбцом таблицы, его значения генерируются автоматически.

```
sqlite> INSERT INTO tbl1 VALUES(NULL, 'hello', 3.10);
sqlite> INSERT INTO tbl1 VALUES(NULL, 'is', 5.34);
sqlite> INSERT INTO tbl1 VALUES(NULL, 'from', NULL);
```

Создадим, пока пустую таблицу tbl2 и индекс по ее первому полю «ikey».

```
sqlite> CREATE TABLE tbl2 (ikey INTEGER PRIMARY KEY, nm
VARCHAR(50));
sqlite> CREATE UNIQUE INDEX itbl2 ON tbl2(ikey);
```



**sqlite>**

### **Команда .dump**

Позволит получить полный дамп (листинг) всех SQL инструкций использованных при создании базы данных или некоторой конкретной таблицы.

Синтаксис:

```
.dump ?TABLE?
```

где TABLE – необязательное имя таблицы, для которой вы хотите получить дамп SQL инструкций.

Пример:

```
sqlite> .dump
```

```
BEGIN TRANSACTION;  
CREATE TABLE tbl1 (one integer primary key autoincrement, two  
varchar(50),  
three real);  
INSERT INTO "tbl1" VALUES(1, 'hello', 3.1);  
INSERT INTO "tbl1" VALUES(2, 'is', 5.34);  
INSERT INTO "tbl1" VALUES(3, 'from', NULL);  
DELETE FROM sqlite_sequence;  
INSERT INTO "sqlite_sequence" VALUES('tbl1', 3);  
CREATE TABLE tbl2 (ikey integer integer primary key, nm VARCHAR(20));  
CREATE UNIQUE INDEX itbl2 on tbl2(ikey);  
COMMIT;  
sqlite>
```

Эту команду можно использовать для экспорта базы данных созданную в SQLite в другую программу работы с базами данных. Для этого необходимо сохранить полный дамп SQL инструкций в файле.

```
sqlite> .output exam.sql --перенаправить вывод в файл
```

```
sqlite> .dump
```

### **Команда .header(s)**

Включает / выключает вывод названий заголовков столбцов.

Синтаксис:

```
.header(s) ON | OFF
```

где

- ON – включить вывод имен заголовков столбцов
- OFF – отключить вывод имен заголовков столбцов

Пример:

```
sqlite> .header ON
sqlite> SELECT * FROM tbl1;
one|two|three
1|hello|3.1
2|is|5.34
3|from|
sqlite> .header OFF
sqlite> SELECT * FROM tbl1;
1|hello|3.1
2|is|5.34
3|from|
sqlite>
```

### Команда .import

С помощью данной команды имеется возможность импортировать данные из текстового файла. Данные в текстовом файле должны быть разделены строкой-разделителем, эта строка разделитель определяется командой .separator. По умолчанию символом-разделителем является символ вертикальной черты "|". Данная команда введена в версии 3.

Синтаксис:

```
.import FILE TABLE
```

где

- FILE – имя входного тестового файла с данными
- TABLE – имя таблицы куда вставляются данные

Пример:

Подготовим следующие данные в файле **exam.txt**

```
1;Roy
2;Ted
3;Sam
4;Bob
```

Вставим эти данные в таблицу **tbl2**, для этого выполним следующие команды:

**sqlite> .separator ';' – установим символ-разделитель в точку с запятой**

**sqlite> .import exam.txt tbl2 – импорт данных**

**sqlite> SELECT \* FROM tbl2;**

**1; Roy**

**2; Ted**

**3; Sam**

**4; Bob**

### **Команда .mode**

Данная команда изменяет формат вывода таблицы SQL запросом SELECT. «SQLite» может вывести результат запроса в следующих форматах: csv, column, html, insert, line, list, tabs, tcl. Для версии 2.8 поддерживается только следующие форматы: line(s), column(s), insert, list, html. Рассмотрим их по порядку.

– Режим csv. В данном формате столбцы записи таблицы разделяются запятыми.

Пример:

**sqlite> .mode csv**

**sqlite> SELECT \* FROM tbl1;**

**1,"hello",3.1**

**2,"is",5.34**

**3,"from",**

**sqlite>**

Режим удобен для экспорта таблицы в сторонние программы, например данный формат, понимает Excel.

Режим column. В этом режиме вывод выровнен по левой границе столбцов записей таблицы.

Пример:

**sqlite> .mode columns**

**sqlite> SELECT \* FROM tbl1;**

**1      hello      3.1**

**2      is          5.34**

**3      from**

**sqlite>**

Смотрите также команду .width для задания ширины столбцов.

Режим html. Вывод в этом режиме выполнен в стиле определения таблицы языка html. Удобен для включения таблицы в страничку сайта.

Пример:

```
sqlite> .mode html
sqlite> SELECT * FROM tbl2;
```

```
<TR><TD>1</TD>
</TD>Roy
</TR>
<TR><TD>2</TD>
</TD>Ted
</TR>
<TR><TD>3</TD>
</TD>Sam
</TR>
<TR><TD>4</TD>
</TD>Bob
</TR>
sqlite>
```

Режим insert. В данном режиме генерируются SQL запросы INSERT для таблицы, имя которой указано третьим параметром в команде.

Синтаксис:

```
.mode insert TABLE
```

где TABLE – имя таблицы, для которой генерируются инструкции INSERT

Пример:

```
sqlite> .mode insert tbl3
sqlite> SELECT * FROM tbl1;
INSERT INTO tbl3 VALUES(1,'hello',3.1);
INSERT INTO tbl3 VALUES(2,'is',5.34);
INSERT INTO tbl3 VALUES(3,'from',NULL);
sqlite>
```

Режим line. Здесь каждый столбец выводится в отдельной строке в формате имя\_столбца = значение. Записи разделяются пустой строкой.

Пример:

```
sqlite> .mode line
sqlite> SELECT * FROM tbl1;
one = 1
two = hello
three = 3.1
```

```
one = 2
two = is
three = 5.34
```

```
one = 3
two = from
three =
sqlite>
```

Режим list. Является режимом, установленным по умолчанию. Каждая запись таблицы выводится в отдельной строке с разделением столбцов символом-разделителем. Символ-разделитель определяется командой .separator, и по умолчанию равен прямой черте '|'.  
Пример:

```
sqlite> .mode list
sqlite> SELECT * FROM tbl1;
1|hello|3.1
2|is|5.34
3|from|
sqlite>
```

Режим tabs. В данном режиме столбцы записей таблицы отделяются друг от друга символом табуляции.

```
Пример:
sqlite> .mode tabs
sqlite> SELECT * FROM tbl1;
1  hello  3.1
2  is     5.34
3  from
sqlite>
```

Режим tcl. Вывод выполнен в формате элементов списка языка TCL.

```
Пример:
sqlite> .mode tcl
sqlite> SELECT * FROM tbl2;
"1"  " Roy \r"
"2"  " Ted\r"
"3"  " Sam\r"
"4"  " Bob\r"
sqlite>
```

## Команда .output

Данная команда перенаправляет вывод в файл или обратно на экран.

Синтаксис:

```
.output FILENAME
```

```
.output stdout
```

где,

- FILENAME – имя файла, куда направлен вывод
- stdout – направит вывод на экран

## Команда .read

Читает и выполняет SQL инструкции из файла.

Синтаксис:

```
.read FILENAME
```

где FILENAME - имя файла содержащего SQL инструкции

Пример:

Создайте текстовый файл, содержащий следующие SQL запросы:

```
BEGIN TRANSACTION;  
CREATE TABLE tbl3 (name VARCHAR(30), lastname VARCHAR(50),  
num INT);  
INSERT INTO tbl3 VALUES ('Jim', 'Green', 1653);  
INSERT INTO tbl3 VALUES ('Ann', 'Folkin', 1891);  
INSERT INTO tbl3 VALUES ('Fredy', 'Donovan', 2345);  
COMMIT;
```

Назовем созданный файл exam.sql, и выполним следующие команды:

```
sqlite> .read exam.sql  
sqlite> SELECT * FROM tbl3;  
Jim|Green|1653  
Ann|Folkin|1891  
Fredy|Donovan|2345  
sqlite>
```

## Команда .schema

Данная команда выдаст все SQL инструкции CREATE TABLE и CREATE INDEX использованные для создания таблиц и индексов текущей базы данных. Возможно указание имени конкретной таблицы, для которой необходимо вывести SQL инструкцию ее создания.

Синтаксис:

```
.schema ?TABLE?
```

где, ?TABLE ? - необязательное имя таблицы

Пример:

```
sqlite> .schema
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE tbl1 (one integer primary key autoincrement, two
varchar(50), three real);
CREATE TABLE tbl2 (ikey integer integer primary key, nm
VARCHAR(20));
CREATE TABLE tbl3 (name VARCHAR(30), lastname VARCHAR(50),
num INT);
CREATE UNIQUE INDEX itbl2 on tbl2(ikey);
sqlite>
```

### Команда .tables

Команда выдаст список названий таблиц созданных в текущей базе данных. Возможно, использовать определение шаблона, для выдачи имен таблиц удовлетворяющему шаблону. Синтаксис шаблона аналогичен определению шаблона в предложении LIKE.

Синтаксис:

```
.tables ?PATTERN?
```

где ?PATTERN? – необязательное определение шаблона поиска

Пример:

```
sqlite> .tables
sqlite_sequence tbl1      tbl2      tbl3
sqlite> .tables %tb% 'с шаблоном
tbl1 tbl2 tbl3
sqlite>
```

### Команда .width

Используется для установки ширины столбцов, при формате вывода установленном в column.

Синтаксис:

```
.width NUM NUM :
```

где NUM - ширина соответствующего столбца в символах.

Пример:

```
sqlite> .mode column
sqlite> .width 3 10 7
```

```
sqlite> SELECT * FROM tbl1;  
1 hello 3.1  
2 is 5.34  
3 from  
sqlite>
```

### Вопросы к зачету

1. Информационные системы. Структура и классификация информационных систем.
2. Базы данных. Классификация. Модели представления данных.
3. Системы управления базами данных. Примеры СУБД. Классификация. Состав языковых средств.
4. Архитектура СУБД.
5. Типовая организация СУБД. Основные функции СУБД.
6. Реляционная модель данных, основная терминология.
7. Администрирование БД. Состав специалистов, обслуживающих БД. Функции системного администратора.
8. Основы теории проектирования БД. Этапы проектирования реляционных БД.
9. Инфологическое моделирование данных. ER-диаграммы.
10. Даталогическое проектирование. Переход от инфологической к реляционной модели.
11. Нормализация отношений. Нормальные формы. Пример нормализации.
12. Язык реляционных баз данных SQL: основные характеристики.
13. Организация SQL-запросов к БД.
14. Процесс прохождения пользовательского запроса.
15. Обеспечение безопасности баз данных. Защита информации в БД. Проверка полномочий.
16. Базисные средства манипулирования реляционными БД. Реляционная алгебра.
17. СУБД типа «клиент-сервер». Распределенные БД.



## ЗАКЛЮЧЕНИЕ

В настоящее время информацию рассматривают как один из основных ресурсов развития общества, а информационные системы и технологии как средство повышения производительности и эффективности работы людей.

Информационные системы и базы данных стали неотъемлемой частью нашей повседневной жизни: покупка в супермаркете, расчеты с использованием кредитной карты, заказ путевки в туристическом агентстве, заказ книг в местной библиотеке, оформление страхового полиса, работа в Интернет, обучение в университете и пр. Наиболее широко информационные системы используются в производственной, управленческой и финансовой деятельности.

В широком смысле информационной системой можно назвать любую организационно-упорядоченную совокупность документов.

Информационную систему можно рассматривать как компьютеризированную систему, обеспечивающую автоматизированный сбор, хранение, поиск, обработку и передачу значительных объемов информации.

*Информационная система* – совокупность тем или иным образом структурированных данных и комплекса аппаратно-программных средств для хранения данных и манипулирования ими.

Учебно-методическое пособие, раскрывающее структуру любой информационной системы, содержит в себе не только теоретический материал по разработке и проектированию баз данных, но и методические указания к проведению занятий по дисциплинам «Базы данных», «Информационные системы» для бакалавров.

Данное пособие может быть использовано в процессе изучения других модулей, курсов вариативной части базового и профессионального циклов дисциплин: «Информационные ресурсы», «Информационные технологии в образовании», «Современные информационные технологии», «Интернет-технологии. Электронные образовательные ресурсы», «Основы математической обработки информации» и др.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Грофф, Дж. SQL: полное руководство / Дж. Грофф, П. Вайнберг; пер. с англ. – К. : BHV, 2001.
2. Дейт, К. Введение в системы баз данных / К. Дейт. – М.: Вильямс, 2000.
3. Диго, С. М. Базы данных / С. М. Диго. – М. : Финансы, 2005.
4. Информатика: учебник / под ред. проф. Н. В. Макаровой. – М. : Финансы и статистика, 1997. – 678 с.
5. Информационные системы и технологии в экономике: учебник / Т. П. Барановская, В. И. Лойко, М. И. Семенов и др. / под ред. В. И. Лойко. – 2-е изд., доп. и перераб. – М. : Финансы и статистика, 2005. – 416 с.
6. Информационные системы: учебно-методические материалы [Электронный ресурс]. – Режим доступа: [http://old.miemp.ru/student/metod/fu/i\\_sis.pdf](http://old.miemp.ru/student/metod/fu/i_sis.pdf) (дата обращения: 25.09.2016).
7. Карпова, Т. С. Базы данных. Модели, разработка, реализация / Т. С. Карпова. – СПб. : Питер, 2001.
8. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. – М. : Вильямс, 2003.
9. Кренке, Д. Теория и практика построения баз данных / Д. Кренке – СПб. : Питер, 2003. – 800 с.
10. Кузин, А. В. Базы данных: учебное пособие для студентов высш. учеб. заведений / А. В. Кузин, С. В. Левонисова. – 2-е изд., стер. – М. : Академия, 2008. – 320 с.
11. Кузнецов, С. Д. SQL. Язык реляционных баз данных / С. Д. Кузнецов. – М. : Майор, 2001. – 192 с.
12. Ледовских, И. А. Программа и материалы элективного курса для учащихся 10–11 классов «Теория и практика построения реляционных баз данных [Электронный ресурс]. – Режим доступа: <http://www.school-collection.edu.ru/catalog/res/930fb3ea-5271-f6f1.../view/> (дата обращения: 1.09.2016).
13. Советов, Б. Я. Базы данных. Теория и практика / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовский. – М. : Высшая школа, 2005. – 464 с.
14. Харрингтон, Джен Л. Проектирование реляционных баз данных / Джен Л. Харрингтон. – М. : Лори, 2006. – 230 с.
15. Хомоненко А. Д. Базы данных / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев. – СПб. : Корона-принт, 2004. – 736 с.
16. Элективные курсы «Информатика для учащихся 10–11 классов». [Электронный ресурс]. – Режим доступа: <http://rudocs.exdat.com/docs/index-9036.html?page=2> (дата обращения: 1.09.2016).

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
1 ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ .....	5
1.1 Базы данных и системы управления базами данных.....	5
1.2 Классификация баз данных .....	7
1.3 Объекты базы данных.....	16
1.4 Проектирование реляционной базы данных .....	17
2 ОСНОВЫ РАБОТЫ С MS ACCESS .....	21
2.1 Этапы проектирования базы данных в MS Access .....	21
2.2 Основные принципы работы MS Access.....	21
2.3 Создание базы данных в MS Access.....	22
2.4 Создание новой таблицы .....	24
2.5 Связи между таблицами .....	28
2.6 Создание запросов.....	31
2.7 Создание форм.....	46
2.8 Создание отчетов.....	60
2.9 Создание макросов .....	74
2.10 Фильтры и защита баз данных.....	80
3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРОВЕДЕНИЮ ЗАНЯТИЙ ПО ДИСЦИПЛИНАМ «БАЗЫ ДАННЫХ» И «ИНФОРМАЦИОННЫЕ СИСТЕМЫ» .....	82
3.1 Цели освоения дисциплин.....	82
3.2 Место дисциплин в структуре ОПОП ВО бакалавриата.....	82
3.3 Компетенции обучающегося, развиваемые в результате освоения дисциплин «Базы данных» и «Информационные системы» .....	83
3.4 Структура и содержание дисциплин .....	84
3.5 Образовательные технологии .....	89
3.6 Учебно-методическое обеспечение самостоятельной работы студентов и оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплин .....	90
3.7 Рейтинговая система для определения индивидуальной динамики развития компетенций бакалавра по дисциплинам «Базы данных», «Информационные системы».....	91
3.8 Материально-техническое обеспечение дисциплины.....	93
4 ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....	94
ЗАКЛЮЧЕНИЕ.....	137
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	138

Учебное издание

**Ледовских Ирина Анатольевна**  
**Шулика Надежда Анатольевна**  
**Табачук Наталья Петровна**

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ КАК СРЕДСТВО ПОИСКА  
И ХРАНЕНИЯ ДАННЫХ**

Учебно-методическое пособие

Отпечатано с авторского оригинал-макета

Дизайнер обложки *Е. И. Саморядова*

Подписано в печать 6.04.2017

Бумага писчая.

Гарнитура «Таймс». Печать цифровая.

Формат 60×84 1/16. Усл. печ. л. 8,3

Тираж 100 экз. Заказ 112.

Издательство Тихоокеанского государственного университета  
680035, Хабаровск, ул. Тихоокеанская, 136.  
Отдел оперативной полиграфии издательства  
Тихоокеанского государственного университета  
680035, Хабаровск, ул. Тихоокеанская, 136.