

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»  
В САНКТ-ПЕТЕРБУРГЕ

---

Серия: «Управление образованием»

МАГИСТЕРСКАЯ ПРОГРАММА  
«УПРАВЛЕНИЕ ОБРАЗОВАНИЕМ»

Направление 081100.68  
«Государственное и муниципальное управление»

*Л. А. Керов*

**ИНФОРМАЦИОННО-  
КОММУНИКАЦИОННЫЕ  
ТЕХНОЛОГИИ В УПРАВЛЕНЧЕСКОЙ  
ПРАКТИКЕ**

**Часть 2. Базы данных**

*Учебное пособие*



Санкт-Петербург  
2012

УДК 681.3.07  
ББК 32.973.26-0.18.2.75  
К36

Рецензенты:

**В. А. Костин**, доцент кафедры информатики СПбГУ;

**С. В. Афанасьева**, доцент кафедры бизнес-информатики Национального исследовательского университета «Высшая школа экономики» в Санкт-Петербурге

**К36 Керов Л. А.** Информационно-коммуникационные технологии в управленческой практике. Часть 2. Базы данных. Учебное пособие [Текст] / Л.А. Керов ; Санкт-Петербургский филиал Нац. исслед. ун-та «Высшая школа экономики». — СПб.: Отдел оперативной полиграфии НИУ ВШЭ — Санкт-Петербург, 2012. — 74 с. — 70 экз.

Учебное пособие посвящено созданию реляционных баз данных и их использованию в управленческой практике образовательных учреждений. В качестве инструментальной системы применяется система управления базами данных Microsoft Office Access 2007.

В учебном пособии рассматриваются вопросы концептуального проектирования базы данных на основе анализа бизнес-правил работы образовательного учреждения, а также вопросы физической реализации базы данных. Детально рассматриваются различные аспекты создания и связывания таблиц базы данных. Особое внимание уделяется извлечению данных из таблиц с помощью запросов и фильтров, созданию форм и отчетов.

Материал учебного пособия апробирован автором в курсах, которые он преподавал студентам математико-механического факультета Санкт-Петербургского государственного университета и студентам Национального исследовательского университета – Высшей школы экономики.

Для студентов, обучающихся по направлению 081100.68 «Государственное и муниципальное управление» подготовки магистра, для магистерской программы «Управление образованием»

*Рекомендовано к печати*

*Учебно-методическим советом НИУ ВШЭ в Санкт-Петербурге*

© Керов Л., 2012

© НИУ ВШЭ — Санкт-Петербург, 2012

## ОГЛАВЛЕНИЕ

<b>ГЛАВА 1. СОЗДАНИЕ БАЗЫ ДАННЫХ В ACCESS 2007</b> .....	5
<b>1.1. Понятие базы данных</b> .....	5
<b>1.2. Концептуальная модель базы данных</b> .....	5
<b>1.3. Создание базы данных в Access 2007</b> .....	7
<b>1.4. Создание таблицы STUDENTS</b> .....	7
1.4.1. Создание структуры таблицы .....	7
1.4.2. Задание ключевого поля .....	9
1.4.3. Переход в режим таблицы .....	10
1.4.4. Заполнение таблицы данными .....	11
1.4.5. Параметры поля .....	11
<b>1.5. Создание таблицы INSTRUCTORS</b> .....	13
1.5.1. Создание структуры таблицы .....	13
1.5.2. Заполнение поля объекта OLE .....	14
1.5.3. Заполнение поля «Вложение» .....	14
<b>1.6. Создание таблицы COURSES</b> .....	16
<b>1.7. Создание таблицы REGISTRATION</b> .....	17
<b>1.8. Создание таблицы TESTS</b> .....	18
1.8.1. Создание структуры таблицы .....	18
1.8.2. Подготовка данных для тестов .....	19
1.8.3. Заполнение таблицы данными .....	20
<b>1.9. Создание таблицы RESULTS</b> .....	20
<b>1.10. Связывание таблиц базы данных</b> .....	21
1.10.1. Открытие схемы данных .....	21
1.10.2. Добавление новой связи .....	22
1.10.3. Добавление остальных связей .....	24
1.10.4. Отображение подчиненных таблиц .....	25
<b>ГЛАВА 2. ЗАПРОСЫ И ФИЛЬТРЫ В ACCESS 2007</b> .....	26
<b>2.1. МАСТЕР ЗАПРОСОВ</b> .....	26
2.1.1. Запуск Мастера .....	26
2.1.2. Выбор таблиц и полей .....	27
2.1.3. Выбор типа отчета .....	28
2.1.4. Указание имени запроса .....	28
2.1.5. Результат запроса .....	29
2.1.6. Редактирование запроса .....	29

<b>2.2. КОНСТРУКТОР ЗАПРОСОВ</b> .....	31
2.2.1. Запуск Конструктора .....	31
2.2.2. Конструирование запроса.....	32
<b>2.3. ИТОГОВЫЙ ЗАПРОС</b> .....	33
2.3.1. Создание запроса .....	33
2.3.2. Добавление групповой операции .....	34
2.3.3. Выполнение статистической операции .....	35
<b>2.4. ФИЛЬТРЫ</b> .....	36
2.2.1. Установка фильтра .....	36
2.2.2. Преобразование фильтра в запрос.....	38
<b>ГЛАВА 3. ФОРМЫ И ОТЧЕТЫ В ACCESS 2007</b> .....	39
<b>3.1. ОДНОТАБЛИЧНАЯ ФОРМА</b> .....	39
3.1.1. Создание формы .....	39
3.1.2. Редактирование формы.....	39
3.1.3. Изменение заголовка формы.....	41
3.1.4. Изменение размеров элемента.....	42
3.1.5. Редактирование надписей .....	43
3.1.6. Режим формы.....	43
<b>3.2. МАСТЕР ФОРМ</b> .....	44
3.2.1. Запуск Мастера форм .....	44
3.2.2. Выбор внешнего вида формы.....	46
3.2.3. Выбор стиля формы.....	46
3.2.4. Задание имени формы .....	47
<b>3.3. МАСТЕР ОТЧЕТОВ</b> .....	48
3.3.1. Запуск Мастера отчетов.....	48
3.3.2. Вид представления данных.....	49
3.3.3. Выбор уровней группировки .....	49
3.3.4. Порядок сортировки записей.....	50
3.3.5. Выбор вида макета.....	50
3.3.6. Выбор стиля.....	51
3.3.7. Задание имени отчета.....	51
3.3.8. Редактирование отчета.....	52
3.3.9. Окончательный вид отчета .....	53
<b>СПИСОК ЛИТЕРАТУРЫ</b> .....	54
<b>ГЛОССАРИЙ</b> .....	55
<b>ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ</b> .....	70

# ГЛАВА 1

## СОЗДАНИЕ БАЗЫ ДАННЫХ В ACCESS 2007

### 1.1. Понятие базы данных

Работа любой организации основывается на использовании множества различных **данных**:

- текстовых
- числовых
- графических и др.

Эти данные необходимы для выполнения управленческих и аналитических функций организации.

Наиболее удобно и эффективно хранить и обрабатывать указанные данные с помощью компьютера. Чтобы это было возможно, данные должны быть специальным образом подготовлены; например, представлены в виде таблиц. Такая специальным образом подготовленная совокупность данных, хранящаяся в файлах внешней памяти компьютера, называется **базой данных** (БД).

Если данные в БД хранятся в виде таблиц, то такая БД называется **реляционной**. Теория реляционных баз данных была предложена и разработана в работах британского ученого в области математики и компьютерных наук Эдгара Кодда [7].

Функционирование базы данных обеспечивается **системой управления базами данных** (СУБД), примером которой может служить **Microsoft Office Access 2007**. Эта система предназначена для управления реляционными БД.

### 1.2. Концептуальная модель базы данных

Приступая к разработке реляционной базы данных, нужно каким-то образом решить, какие таблицы должны в ней храниться. Начинать можно с формулирования правил, которыми руководствуется организация

в своей работе. Такие правила принято называть **бизнес-правилами** (business rules).

Если разрабатывается база данных для университета, то бизнес-правила могут быть сформулированы в виде следующих предложений:

- Каждый **студент** имеет зачетную книжку, в которой указаны его полное имя, номер студенческого билета и направление обучения.
- В зачетной книжке выполняется **регистрация** пройденных учебных **курсов** с указанием названия курса, **преподавателя**, итоговой оценки и даты получения оценки.
- Итоговый контроль знаний проводится в письменном виде с использованием **тестов**, которые разрабатываются для каждого курса.
- **Результаты** прохождения тестов используются для расчета итоговых оценок по курсам.

Существительные, содержащиеся в бизнес-правилах, обозначают некоторые определенные **сущности**, каждая из которых может быть представлена соответствующей таблицей (рис. 1):

- Студенты (STUDENTS);
- Преподаватели (INSTRUCTORS);
- Учебные курсы (COURSES);
- Регистрация (REGISTRATION);
- Тесты (TESTS);
- Результаты (RESULTS).

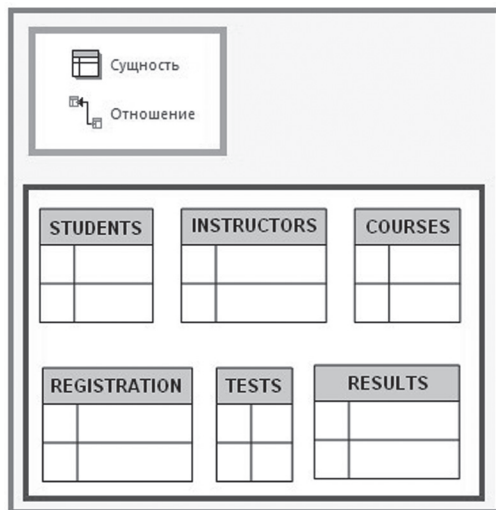


Рис. 1. Сущности концептуальной модели

Таблица состоит из строк и столбцов. Каждая строка таблицы, называемая также **записью**, содержит набор данных, характеризующих конкретный **экземпляр сущности**. Каждый столбец таблицы, называемый также **полем**, содержит данные, которые являются значениями некоторого атрибута экземпляров сущности.

Совокупность сущностей и бизнес-правил, определяющих контекст использования сущностей и отношения между ними, можно рассматривать в качестве **концептуальной модели базы данных**. **Концептуальная модель** (conceptual model) — это определённое множество понятий и связей между ними, являющихся смысловой структурой рассматриваемой предметной области.

Теория концептуальных моделей баз данных, основанных на анализе логики предметной области организации и получившая название **ER-моделей**, была предложена и развита в работах американского ученого в области прикладной математики и компьютерных наук Питера Чена [6].

### 1.3. Создание базы данных в Access 2007

Для создания новой базы данных в СУБД **Microsoft Office Access 2007** нужно выполнить следующие действия (рис. 2):

- 1) В левом верхнем углу главного окна **Access** щелкнуть кнопку **Office**, и в открывшемся меню выбрать команду **Создать**.
- 2) В ответ на это действия появляется раздел **Новая база данных**, в котором в поле **Имя файла** нужно ввести имя файла базы данных и щелкнуть кнопку **Создать**.

### 1.4. Создание таблицы STUDENTS

Создание таблицы подразделяется на два этапа:

- 1) Создание структуры таблицы в режиме **Конструктор**
- 2) Заполнение таблицы данными в **Режиме таблица**

#### 1.4.1. Создание структуры таблицы

Для определения структуры новой таблицы нужно выполнить следующие действия (рис. 3):

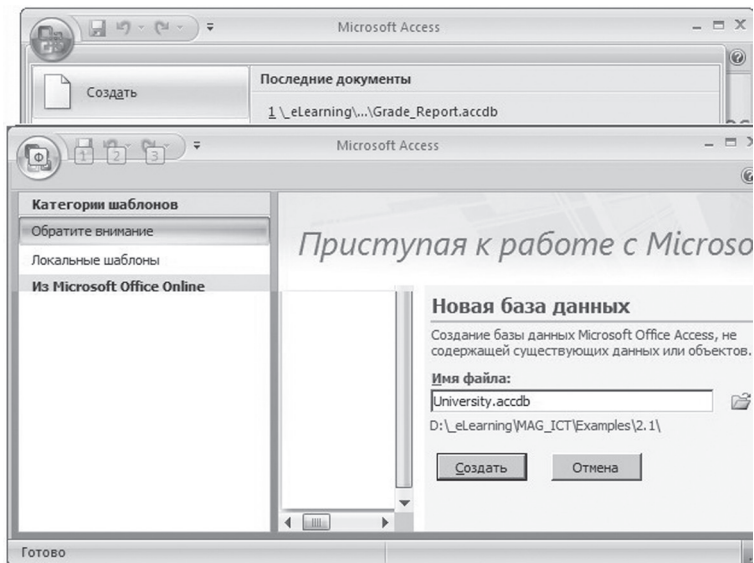


Рис. 2. Создание новой базы данных

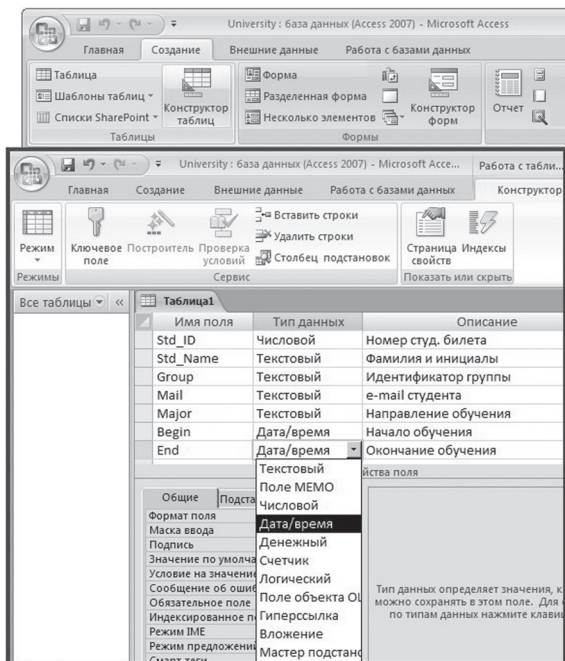


Рис. 3. Определение структуры новой таблицы



- 1) Раскрыть вкладку **Создание**, затем в группе **Таблицы** щелкнуть кнопку **Конструктор таблиц**. В ответ на это действие откроется окно конструктора таблиц, а на ленте появится новая вкладка **Конструктор**
- 2) В столбец **Имя поля** следует ввести название столбца создаваемой таблицы
- 3) В столбце **Тип данных** нужно щелкнуть по кнопку со стрелкой для раскрытия списка типов данных, и выбрать тип данных для данных в создаваемом столбце таблицы
- 4) В столбце **Описание** можно ввести описание создаваемого поля таблицы

#### Типы данных СУБД Access:

Тип	Описание
Текстовый	Текст до 255 символов
Поле МЕМО	Текст до 65535 символов
Числовой	Числа различных форматов
Дата/время	Дата и/или время
Денежный	Денежные значения различных форматов
Счетчик	Счетчик с автоматическим увеличением на 1 для каждой новой записи
Логический	Да/Нет
Поле объекта OLE	Поле для вставки рисунков и объектов других типов
Гиперссылка	Ссылка на файл данного или другого приложения, или на Web-страницу
Вложение	Поле для вложения одного или нескольких файлов произвольных форматов

#### 1.4.2. Задание ключевого поля

**Первичный** ключ (основной идентификатор) – это один или несколько столбцов таблицы, совокупность значений в которых уникальна для каждой строки данных в этой таблице. Первичный ключ называется **простым**, если он состоит из одного столбца; и называется **составным**, если он состоит из нескольких столбцов. Для таблицы **Students** первичным ключом может быть назначен столбец **Stud\_ID**.

Для создания ключевого поля таблицы нужно выполнить следующие действия (рис. 4):

- 1) Щелчком мыши выделить строку поля
- 2) Щелкнуть по кнопке **Ключевое поле**

На этой строке появится значок ключа, показывающий особый статус поля.

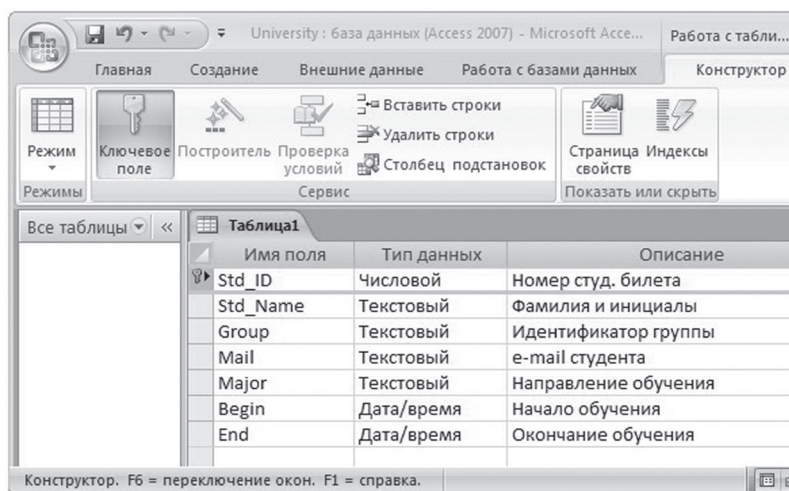


Рис. 4. Создание ключевого поля таблицы

### 1.4.3. Переход в режим таблицы

Заполнение таблицы данными выполняется в **Режиме таблицы**. Для перехода из одного режима в другой предназначена кнопка **Режим**, которая отображается на нескольких вкладках ленты (рис. 5).

Для перехода в **Режим таблицы** нужно выполнить следующие действия:

- 1) Щелкнуть кнопку **Режим** и выбрать команду **Режим таблицы**
- 2) В ответ на это действие откроется диалоговое окно с предложением сохранить таблицу и кнопками **Да** и **Нет**
- 3) Щелкнуть по кнопке **Да**. В ответ на это действие откроется диалоговое окно для ввода имени таблицы. Введем имя таблицы и щелкнем кнопку **OK**

В результате выполненных действий будет создана новая пустая таблица с именем **STUDENTS**, в которую можно вводить данные. Слева от окна таблицы располагается **область переходов**, предназначенная для навигации по объектам базы данных.

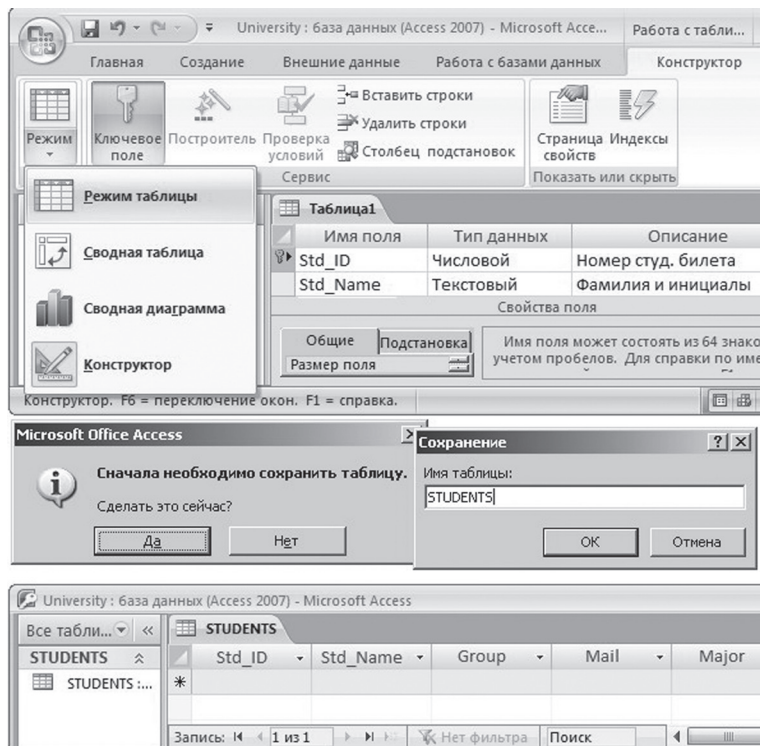


Рис. 5. Переход в режим таблицы

#### 1.4.4. Заполнение таблицы данными

Для создания строки таблицы нужно ввести данные в очередное поле и нажать клавишу **Enter** или **Tab** (рис. 6).

#### 1.4.5. Параметры поля

В нижней части окна конструктора на вкладках **Общие** можно увидеть значения свойств поля таблицы, заданные по умолчанию. Настройка значений свойств полей таблицы позволяет оптимизировать функционирование базы данных.

В качестве примера выполним настройку значений некоторых свойств поля **Begin** таблицы **STUDENTS** (рис. 7):

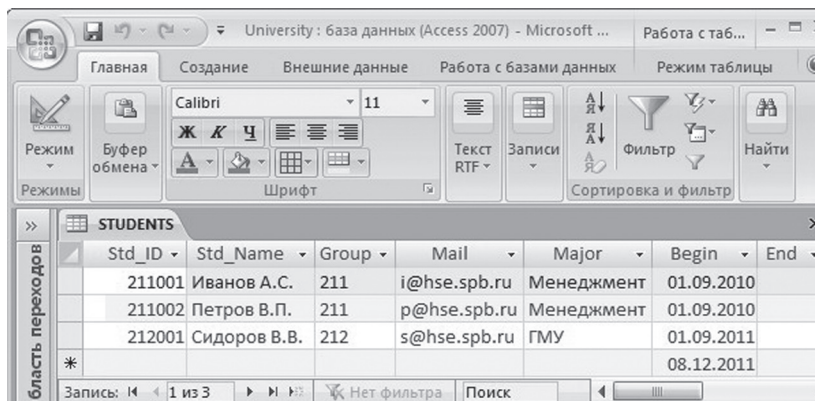


Рис. 6. Заполнение таблицы данными

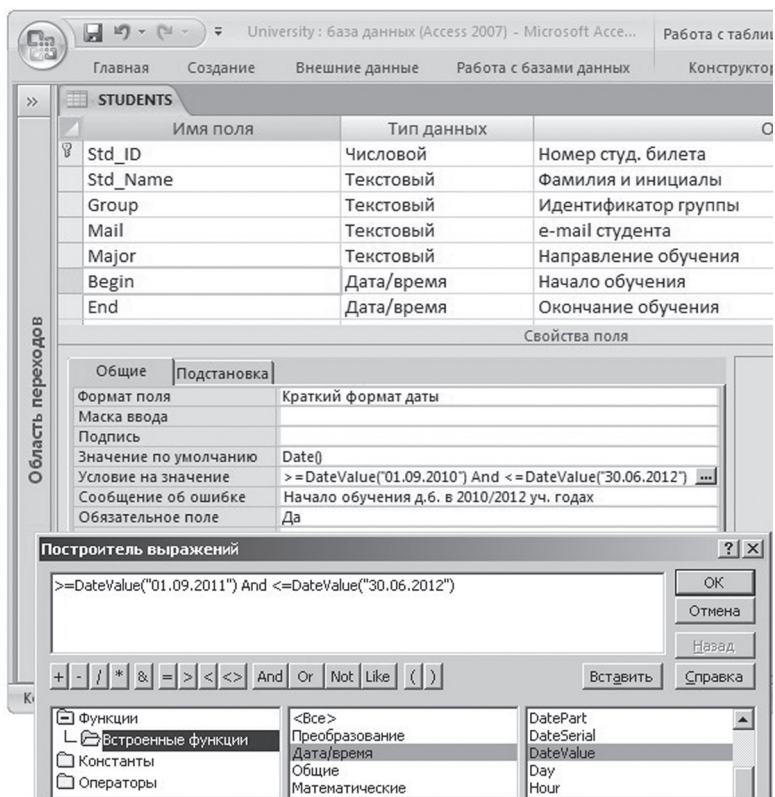


Рис. 7. Настройка значений свойств поля

- В качестве **формата поля** выберем **Краткий формат даты**, не содержащий показания времени, т.к. для фиксации начала обучения это не важно
- В качестве **значения по умолчанию** укажем функцию **Date()**, которая возвращает текущую дату, соответствующую показаниям календаря компьютера
- В качестве **условия на значение** укажем ограничение на возможные значения даты начала обучения. Предположим, что обучение должно начинаться в 2010/2012 учебных годах. Для построения такого ограничения щелкнем по кнопке с многоточием в поле **Условие на значение**.
- Откроется диалоговое окно **Построитель выражений**. Верхнее поле построителя содержит конструируемое выражение. Его можно изменять как путем непосредственного редактирования текста, так и с помощью кнопок и списков построителя.
- Необходимо, чтобы дата регистрации была равна или позднее 1 сентября 2010 года. Поэтому введем оператор «>=».
- Для указания даты воспользуемся встроенной функцией **DataValue**, преобразующей строковую константу в дату в формате Access. Щелкнем по значку папки **Функции** в левом списке построителя, затем щелкнем по значку папки **Встроенные функции**.
- В среднем списке выберем категорию **Дата/время**
- Двойным щелчком по имени функции **DataValue** добавим ее в поле выражения. В качестве ее аргумента укажем строковое изображение нужной даты.
- Добавим логическую операцию **And** и ограничение на правую границу даты регистрации. В заключение щелкнем кнопку **OK**.
- В поле **сообщения об ошибке** введем текст сообщения.
- В поле **Обязательно поле** укажем **Да**.
- Щелкнем по кнопке закрытия окна **Конструктора** и щелкнем по кнопке сохранения базы данных.

## 1.5. СОЗДАНИЕ ТАБЛИЦЫ INSTRUCTORS

### 1.5.1. Создание структуры таблицы

Используя рассмотренные выше приемы работы, создадим структуру таблицы **INSTRUCTORS** (рис. 8).

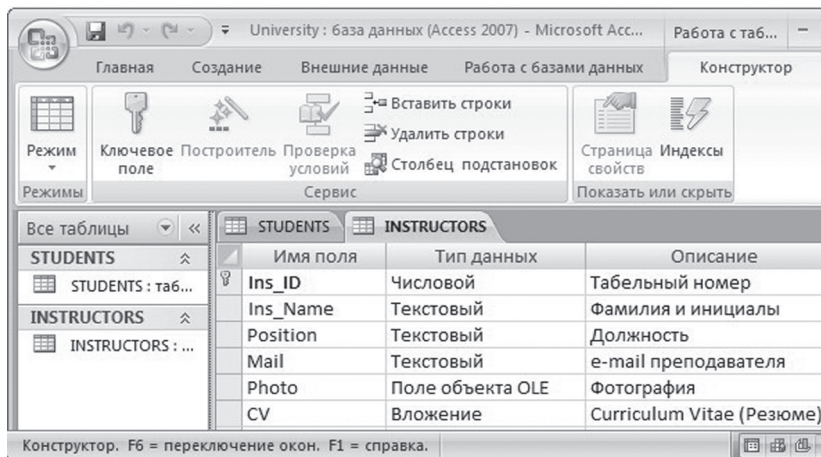


Рис. 8. Создание структуры таблицы INSTRUCTORS

### 1.5.2. Заполнение поля объекта ole

Заполнение таблицы данными выполняется в **Режиме таблицы**. Значения в поля с типами **Текстовый** и **Числовой** вводятся с клавиатуры.

Для заполнения поля **Photo**, которое имеет тип **Поле объекта OLE**, выполним следующие действия (рис. 9):

- 1) Откроем в графическом редакторе нужный графический файл и скопируем его в **Буфер обмена**
- 2) Щелкнем по полю **Photo** правой кнопкой мыши, и в контекстном меню выберем команду **Вставить**
- 3) В поле **Photo** будет вставлен объект типа **Bitmap Image**

### 1.5.3. Заполнение поля «вложение»

Для заполнения поля **CV**, которое имеет тип **Вложение**, нужно выполнить следующие действия (рис. 10):

- 1) Щелкнуть по полю правой кнопкой мыши, и в контекстном меню выбрать команду **Управление вложениями**
- 2) На экран будет выведено диалоговое окно **Вложения**, в котором следует щелкнуть кнопку **Добавить**, и выбрать файл для вложения
- 3) В заключение щелкнуть кнопку **ОК**
- 4) Заполним вторую строку таблицы, оставив незаполненным поле вложения.

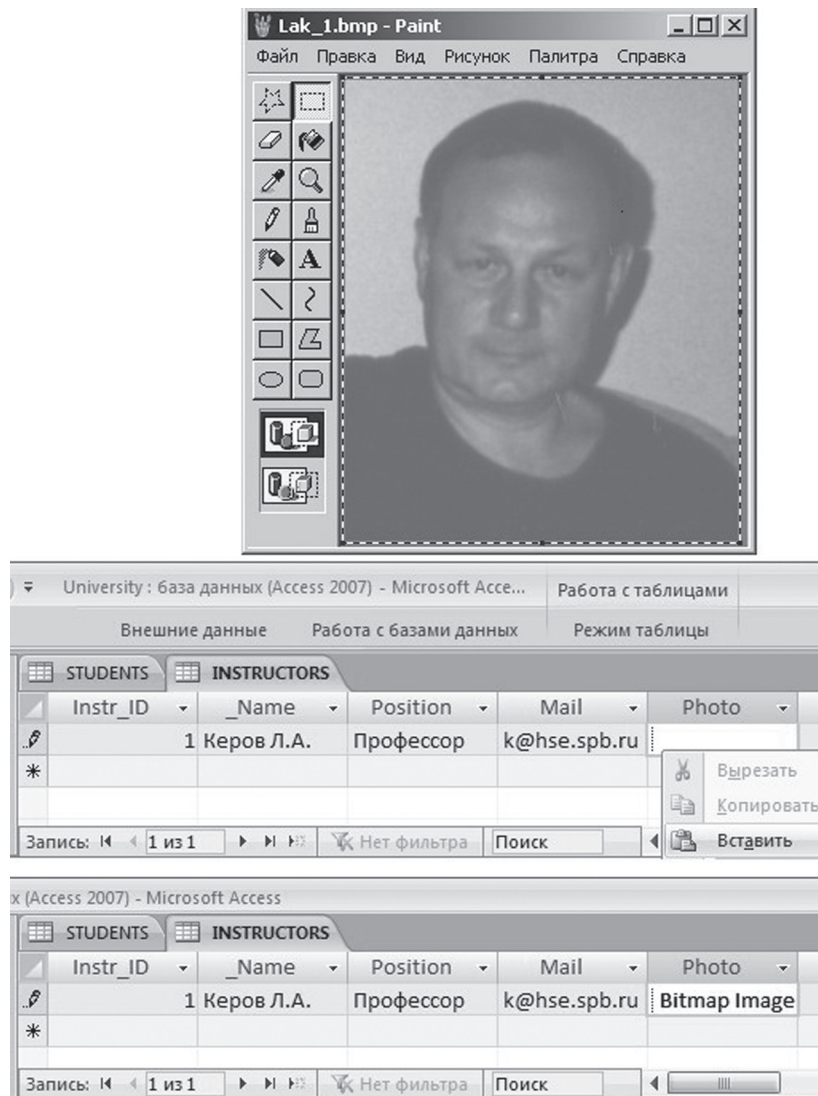


Рис. 9. Заполнение поля Photo

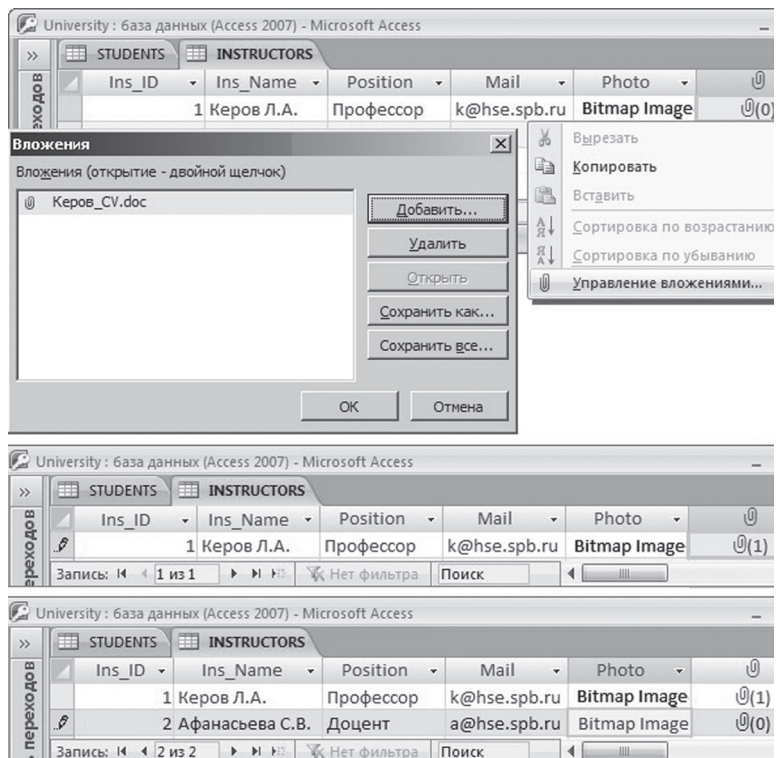


Рис. 10. Заполнение поля CV

## 1.6. СОЗДАНИЕ ТАБЛИЦЫ COURSES

Используя рассмотренные выше приемы работы, создадим и заполним данными таблицу **COURSES** (рис. 11).

В таблицу **COURSES**, помимо полей, которые характеризуют сам учебный курс, добавлено поле **Ins\_ID**. Это поле является первичным ключом в таблице **INSTRUCTORS**. Оно добавлено для связи таблицы **INSTRUCTORS** с таблицей **COURSES**.

Поле таблицы называется **внешним ключом** этой таблицы, если оно является первичным ключом другой таблицы. Таким образом, **Ins\_ID** – это внешний ключ таблицы **COURSES**.



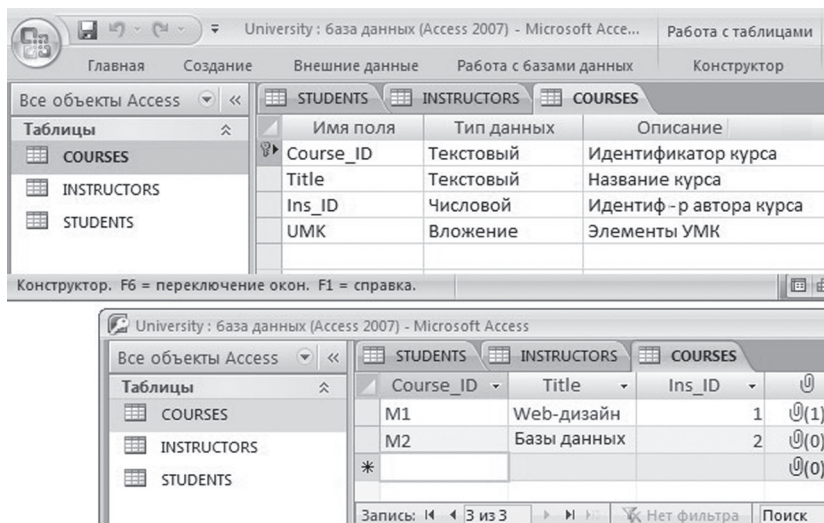


Рис. 11. Создание таблицы COURSES

Поле **UMK** предназначено для вложения в него различных элементов учебно-методического комплекса для дисциплины, например: файла учебной программы, файла учебного пособия и т.п.

## 1.7. СОЗДАНИЕ ТАБЛИЦЫ REGISTRATION

Используя рассмотренные выше приемы работы, создадим и заполним данными таблицу **REGISTRATION** (рис. 12).

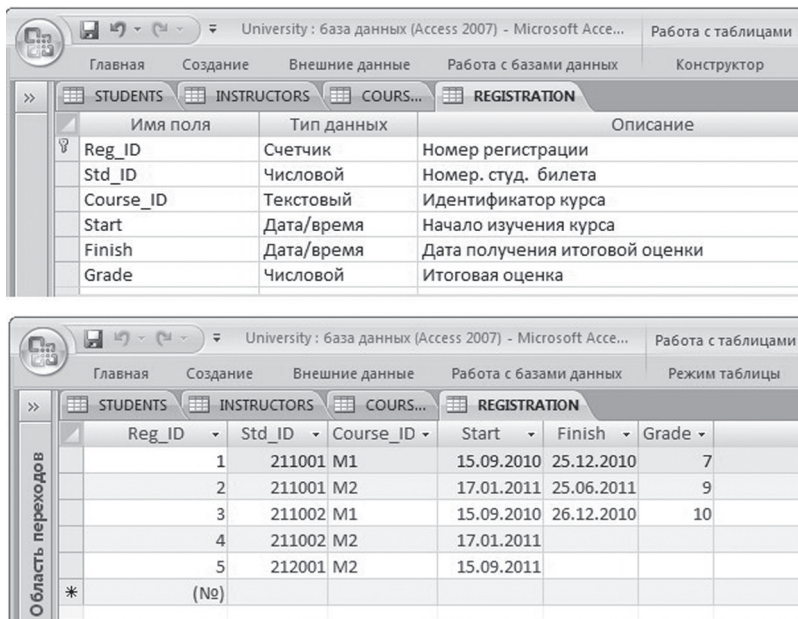


Рис. 12. Создание таблицы REGISTRATION

## 1.8. СОЗДАНИЕ ТАБЛИЦЫ TESTS

### 1.8.1. Создание структуры таблицы

Используя рассмотренные выше приемы работы, создадим структуры таблицы **TESTS** (рис. 13).

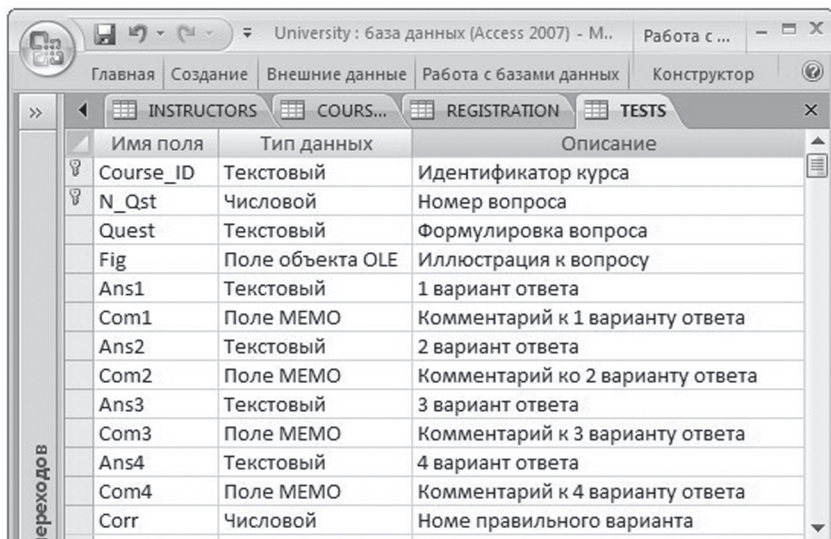


Рис. 13. Создание структуры таблицы TESTS

### 1.8.2. Подготовка данных для тестов

Подготовим данные для тестов в соответствии со структурой таблицы **TESTS**. Данные для тестов по курсу «**М1. Web-дизайн**» представим в виде следующей таблицы:

Вопрос	Варианты ответов	Комментарии
Каким образом группируются элементы формы типа "radio"?	<ul style="list-style-type: none"> <li>По имени name</li> <li>По значению type</li> <li>По значению id</li> <li>Группируются только смежные элементы</li> </ul>	<ul style="list-style-type: none"> <li>Правильно. Все элементы группы переключателей должны иметь одинаковое значение параметра name</li> <li>Неправильно. Этот атрибут задает тип элемента управления</li> <li>Неправильно. Этот атрибут позволяет идентифицировать переключатель при обращении к нему из сценария</li> <li>Неправильно. Смежность элементов никак не влияет на их группировку</li> </ul>

Вопрос	Варианты ответов	Комментарии
Необходимо расположить две таблицы рядом по горизонтали. Какие варианты справятся с этой задачей?	<ul style="list-style-type: none"> <li>• <code>&lt;table&gt; ... &lt;/table&gt;</code> <code>&lt;table&gt; ... &lt;/table&gt;</code></li> <li>• <code>&lt;table align=»left»&gt;</code> <code>&lt;/table&gt;</code> <code>&lt;table&gt; ... &lt;/table&gt;</code></li> <li>• <code>&lt;table&gt; ... &lt;/table&gt;</code> <code>&lt;table align=»left»&gt;</code> ... <code>&lt;/table&gt;</code></li> <li>• Создать одну таблицу из двух ячеек, и в каждую поместить по таблице</li> </ul>	<ul style="list-style-type: none"> <li>• Неправильно. Каждая таблица начинается с новой строки</li> <li>• Неправильно. Каждая таблица начинается с новой строки</li> <li>• Неправильно. Каждая таблица начинается с новой строки</li> <li>• Правильно. С помощью внешней таблицы можно задать двумерную структуризацию данных</li> </ul>
Какие теги создают маркированный список?	<ul style="list-style-type: none"> <li>• <code>&lt;ol&gt;</code></li> <li>• <code>&lt;dl&gt;</code></li> <li>• <code>&lt;ul&gt;</code></li> <li>• <code>&lt;ml&gt;</code></li> </ul>	<ul style="list-style-type: none"> <li>• Неправильно. Этот тег используется в нумерованном списке (ordered list)</li> <li>• Неправильно. Этот тег используется в списке определений (definition list)</li> <li>• Правильно. Этот тег используется в маркированном списке (unnumbered list)</li> <li>• Неправильно. Такого тега не существует</li> </ul>

### 1.8.3. Заполнение таблицы данными

Используя **Буфер обмена**, перенесем предварительно подготовленные данные в таблицу **TESTS** (рис. 14).

TESTS												
C	N	Quest	Fig	Ans1	Com1	Ans2	Com2	Ans3	Com3	Ans4	Com4	C
M1	1	Каким об		По име	Правиль	По знач	Неправ	По знач	Неправ	Группи	Неправ	1
M1	2	Необход		<code>&lt;table&gt;</code>	Неправ	<code>&lt;table :</code>	Неправ	<code>&lt;table&gt;</code>	Неправ	Создат	Правил	4
M1	3	Какие те		<code>&lt;ol&gt;</code>	Неправ	<code>&lt;dl&gt;</code>	Неправ	<code>&lt;ul&gt;</code>	Правиль	<code>&lt;ml&gt;</code>	Неправ	3

Рис. 14. Заполнение данными таблицы TESTS

### 1.9. СОЗДАНИЕ ТАБЛИЦЫ RESULTS

Используя рассмотренные выше приемы работы, создадим и заполним данными таблицу **RESULTS** (рис. 15).

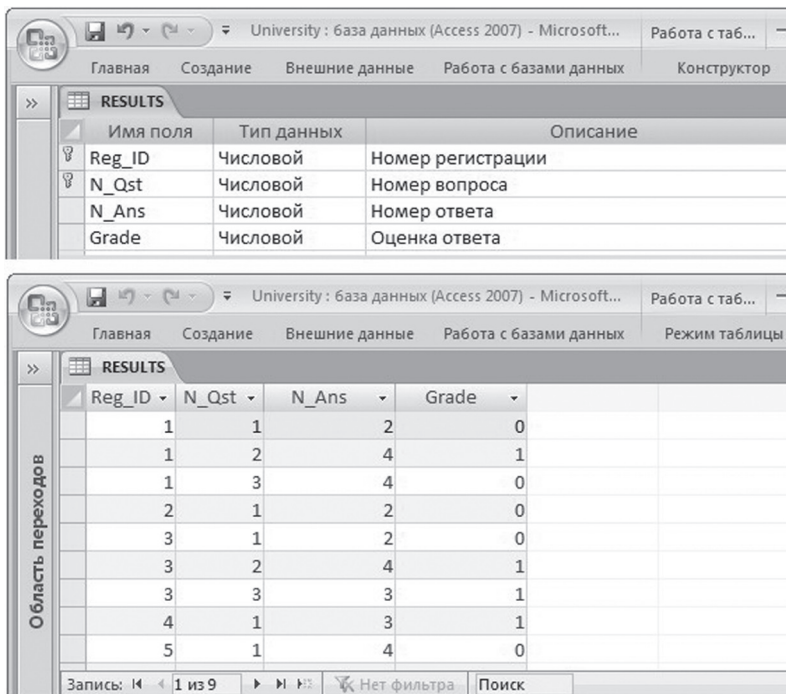


Рис. 15. Создание таблицы RESULTS

## 1.10. СВЯЗЫВАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ

### 1.10.1. Открытие схемы данных

Для открытия **Схемы данных** нужно выполнить следующие действия (рис. 16):

- 1) Раскрыть вкладку **Работа с базами данных**
- 2) В группе **Показать или скрыть** выбрать команду **Схема данных**
- 3) Откроется диалоговое окно **Добавление таблицы**, в котором следует выбрать нужные таблицы и щелкнуть кнопку **Добавить**
- 4) В ответ на это действие откроется объект **Схема данных**. Изображения таблиц на схеме данных можно перемещать: следует поместить курсор мыши на заголовок таблицы, нажать левую кнопку и, удерживая кнопку нажатой, переместить курсор мыши.

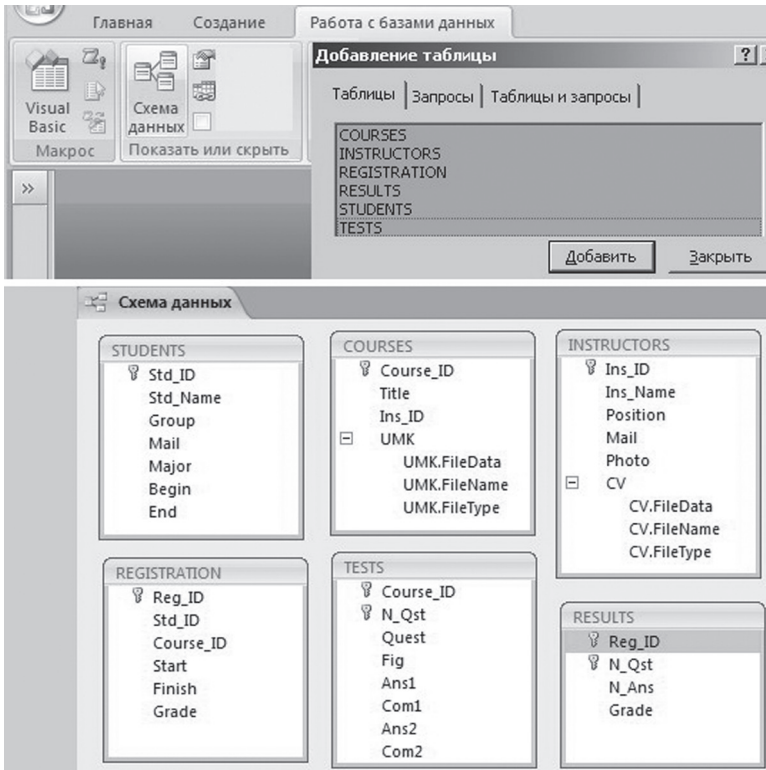


Рис. 16. Открытие схемы данных

### 1.10.2. Добавление новой связи

Для добавления связи между таблицами, нужно выполнить следующие действия (рис. 17):

- 1) Закрыть вкладки для всех таблиц
- 2) Поместить в таблице (например, **STUDENT**) курсор мыши на поле для связи (например, **Std\_ID**), нажать левую кнопку и, удерживая кнопку нажатой, переместить курсор мыши на одноименное поле в другой таблице (например, **Registration**)
- 3) Откроется диалоговое окно **Изменение связей**, в котором следует установить флажки **Обеспечение целостности данных**, **Каскадное обновление связанных полей**, **Каскадное удаление связанных полей**
- 4) Щелкнуть кнопку **Создать**

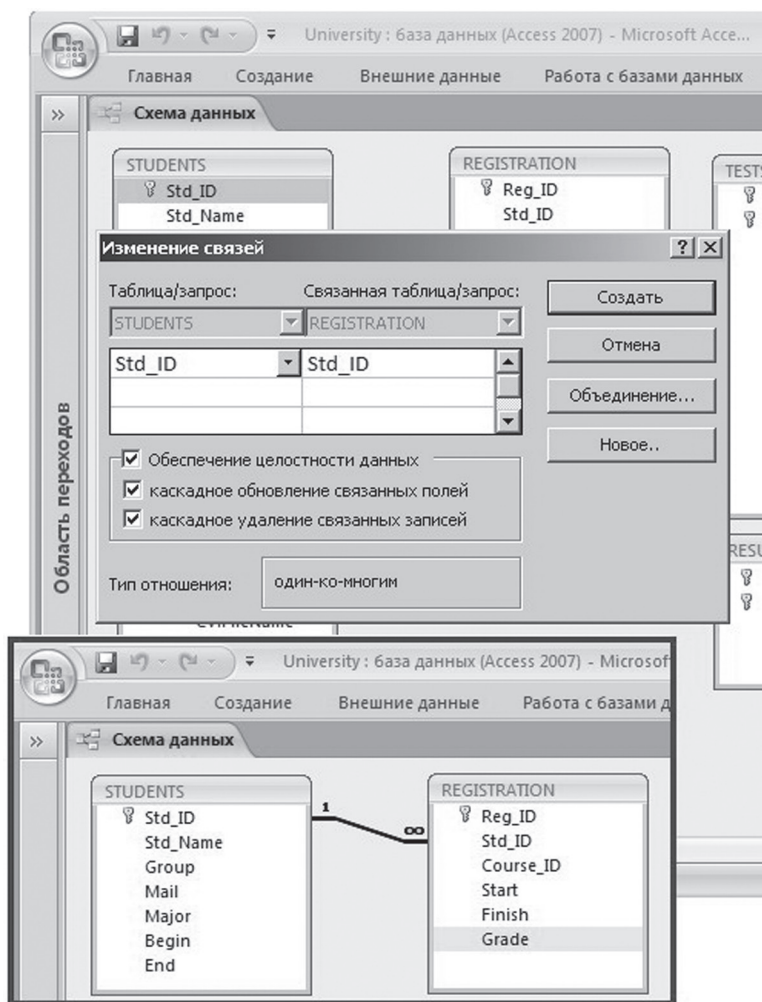


Рис. 17. Добавление связи между таблицами

В результате выполненных действий будет создана новая связь между таблицами, в которой первичный ключ главной таблицы (сторона «один») связан с внешним ключом другой таблицы (сторона «многие»).

Если установлен флажок **Обеспечение целостности данных**, то Access запретит:

- Удалять запись в главной таблице, если в подчиненной таблице имеется связанная с ней запись
- Добавлять в подчиненную таблицу запись с таким значением внешнего ключа, для которого отсутствует такое же значение первичного ключа в главной таблице

Если установлен флажок **Каскадное обновление связанных полей**, то при изменении значения первичного ключа в главной таблице будет автоматически выполняться изменение аналогичного внешнего ключа во всех связанных таблицах.

Если установлен флажок **Каскадное удаление связанных полей**, то при удалении записи в главной таблице будут автоматически удалены все записи в связанных таблицах с аналогичными значениями внешних ключей.

### 1.10.3. Добавление остальных связей

Используя рассмотренные выше приемы работы, создадим и остальные связи между таблицами (рис. 18).

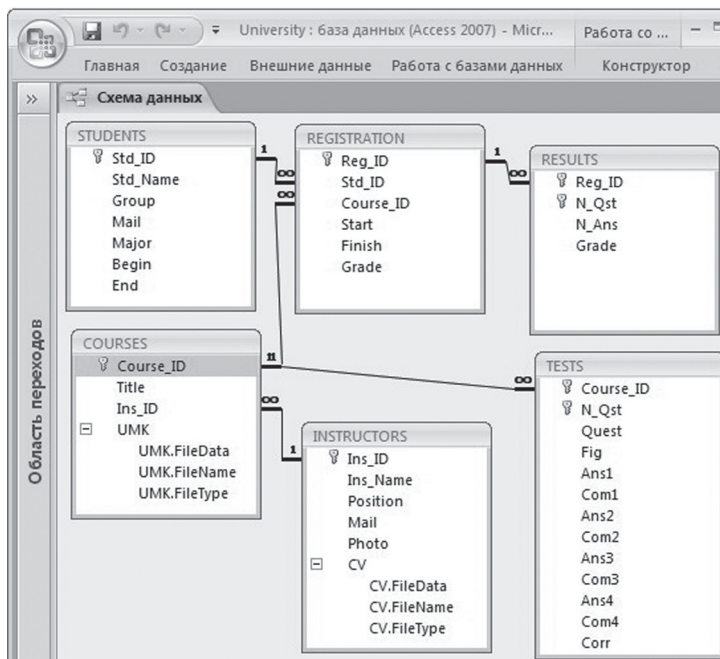


Рис. 18. Схема данных



### 1.10.4. Отображение подчиненных таблиц

В режиме Таблицы могут одновременно отображаться несколько связанных таблиц. На схеме данных видно, что у таблицы **STUDENTS** есть подчиненная таблица **REGISTRATION**.

У главной таблицы слева от первого столбца отображается квадратик со знаком «плюс» (+). Если щелкнуть по нему левой кнопкой мыши, то откроется часть подчиненной таблицы, содержащая только те записи, которые связаны с записью в главной таблице. При этом знак «плюс» (+) заменяется на знак «минус» (-). Если щелкнуть по знаку «минус» (-), то подчиненная таблица закроется (Рис.19).

Для связанных таблиц автоматически выполняется контроль целостности данных, поэтому при создании базы данных удобнее сначала разработать структуру таблиц и выполнить их связывание, а уже потом заполнять таблицы данными.

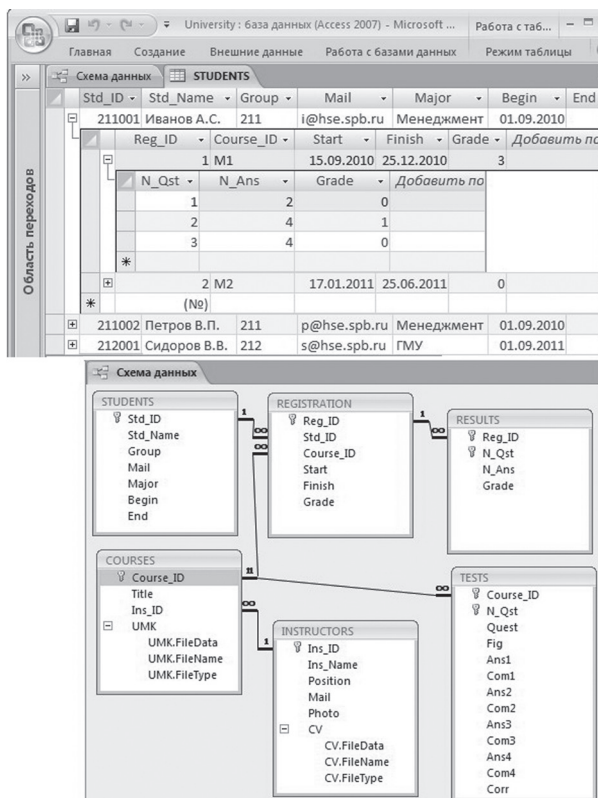


Рис. 19. Отображение подчиненных таблиц

## ГЛАВА 2. ЗАПРОСЫ И ФИЛЬТРЫ В ACCESS 2007

### 2.1. МАСТЕР ЗАПРОСОВ

**Запрос** является инструментом поиска данных в БД и их структурирования. Запрос может использовать одну или несколько таблиц, выбирает из них часть данных и формирует на их основе виртуальную таблицу.

#### 2.1.1. Запуск мастера

Для создания простого запроса с помощью **Мастера** выполним следующие действия (рис. 20):

- 1) Раскроем вкладку **Создание** и щелкнем в группе **Другие** кнопку **Мастер запросов**
- 2) Появится диалоговое окно **Новый запрос**, в котором выберем из предлагаемого списка элемент **Простой запрос** и щелкнем кнопку **ОК**

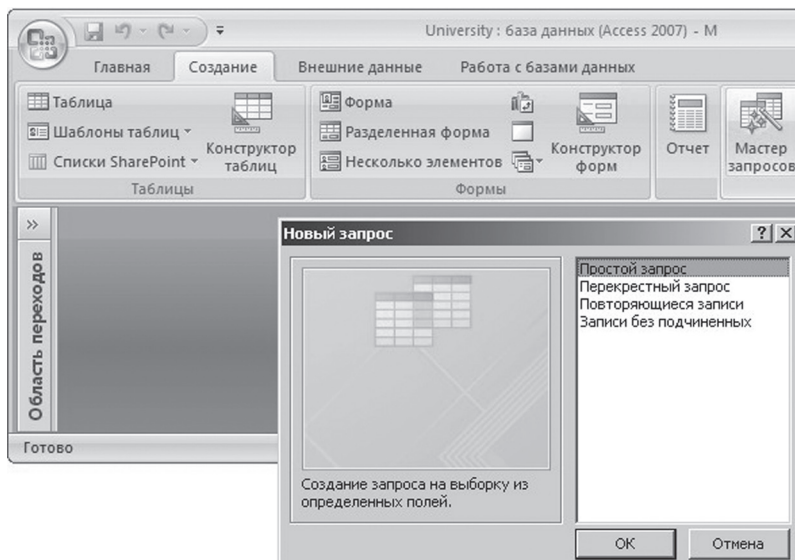


Рис. 20. Запуск мастера запросов

### 2.1.2. Выбор таблиц и полей

Допустим, что требуется получить таблицу успеваемости студентов, в которой должны быть следующие столбцы:

- Фамилия и инициалы студента
- Номер его студенческого билета
- Идентификатор и название изучаемого курса
- Итоговая оценка

Во втором диалоговом окне **Мастера запросов** выберем таблицы и поля, которые должны быть показаны в результате запроса, затем щелкнуть кнопку **Далее** (рис. 21).

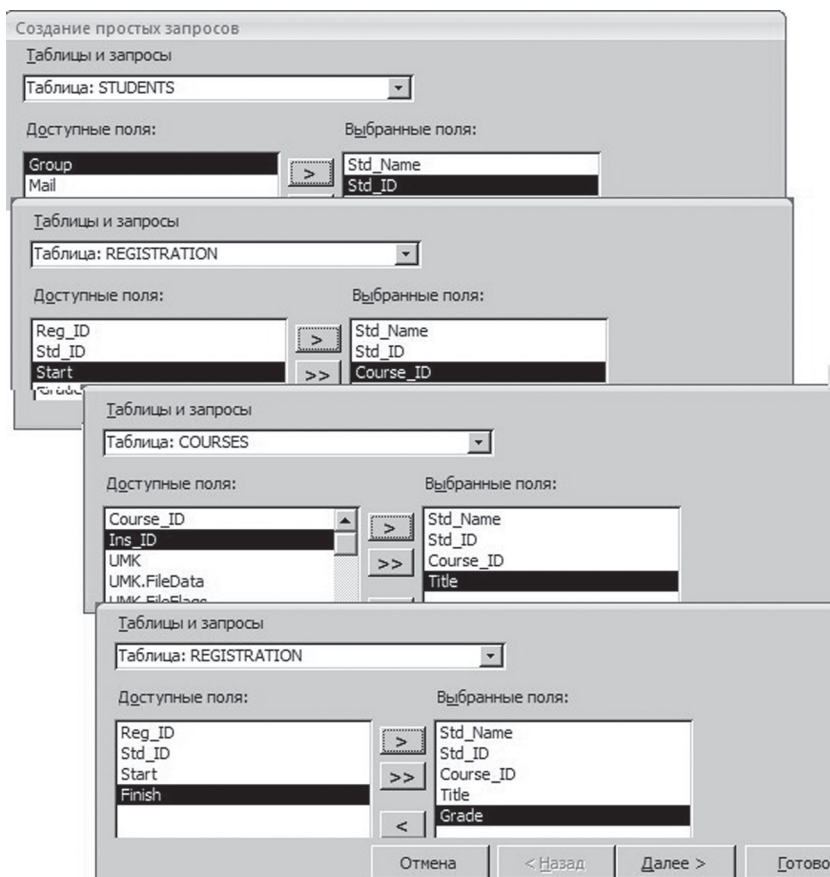


Рис. 21. Выбор таблиц и полей

### 2.1.3. Выбор типа отчета

В третьем диалоговом окне **Мастера запросов** выберем подробный тип отчета, затем щелкнем кнопку **Далее** (рис. 22).

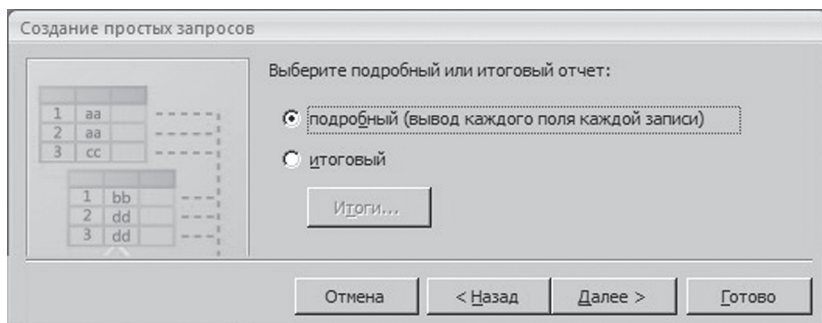


Рис.22. Выбор типа отчета

### 2.1.4. Указание имени запроса

В четвертом диалоговом окне **Мастера запросов** укажем имя запроса, затем щелкнем кнопку **Готово** (рис. 23).

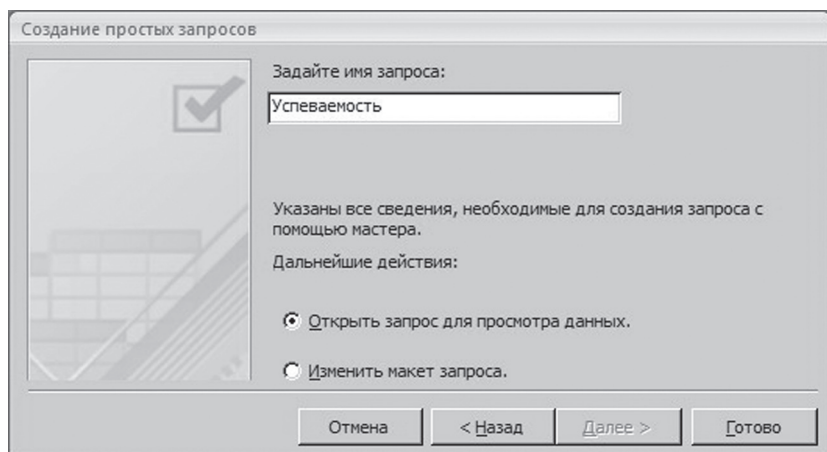


Рис. 23. Указание имени запроса

### 2.1.5. Результат запроса

В результате выполненных действий на экране будет отображен объект, соответствующий результату запроса и представляющий собой виртуальную таблицу (рис. 24). Этот объект сохраняется в файле базы данных. Чтобы в дальнейшем посмотреть результаты запроса, достаточно открыть его в режиме таблицы.

Std_Name	Std_ID	Course_ID	Title	Grade
Иванов А.С.	211001	M1	Web-дизайн	3
Иванов А.С.	211001	M2	Базы данных	0
Петров В.П.	211002	M1	Web-дизайн	7
Петров В.П.	211002	M2	Базы данных	10
Сидоров В.В.	212001	M2	Базы данных	0
*				

Рис. 24. Результат запроса

### 2.1.6. Редактирование запроса

Поскольку результат запроса отображается в виде таблицы, то возникает желание не только просматривать, но и изменять данные в этой таблице. Поскольку данная таблица является виртуальной, то на самом деле редактирование полей в запросе означает редактирование полей в таблицах, на основе которых строился этот запрос.

Однако такое редактирование не всегда возможно. Является ли запрос обновляемым, отображается визуально: если в конце его записей содержится пустая строка, помеченная знаком «звездочка» (\*) в области выделения записей слева, то в эту строку можно вводить данные, которые создадут новую запись. Если такая строка отсутствует, то добавлять записи в запрос и изменять поля запроса нельзя.

Предположим, что нужно скрыть столбец **Course\_ID** (рис. 25):

- Щелчком правой кнопкой мыши по этому столбцу
- Из контекстного меню выберем команду **Скрыть столбцы**

Если в дальнейшем нужно будет показать этот столбец:

- Щелчком правой кнопкой мыши по любому столбцу
- Из контекстного меню выберем команду **Отобразить столбцы**
- В появившемся диалоговом окне **Отображение столбцов** установим флажок того столбца, который должен быть показан

Заметим, что указанные операции можно выполнять с любой таблицей, а не только с запросами.

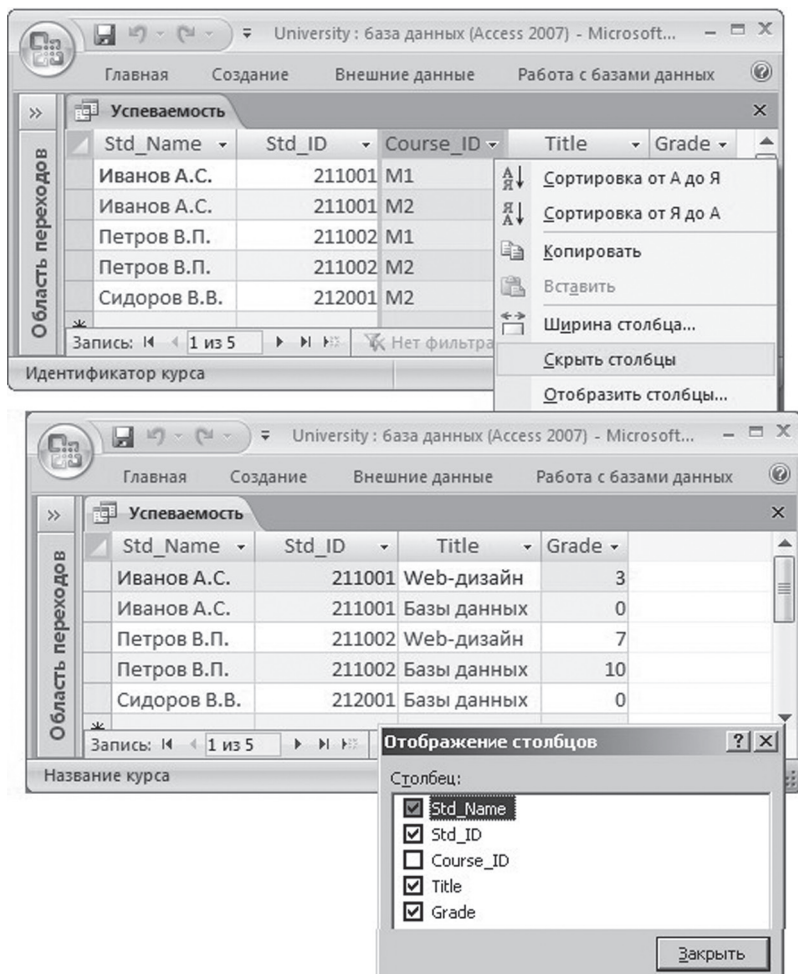


Рис. 25. Редактирование запроса

## 2.2. КОНСТРУКТОР ЗАПРОСОВ

### 2.2.1. Запуск конструктора

Если при построении запроса нужны более сложные условия выборки данных из таблиц, чем продемонстрированные выше, то использует-ся **Конструктор запросов** (рис. 26):

- 1) На вкладке **Создание** в группе **Другие** выберем команду **Конструктор запросов**
- 2) Появится диалоговое окно **Добавление таблицы**, в котором выберем нужные таблицы и щелкнем кнопку **Добавить**

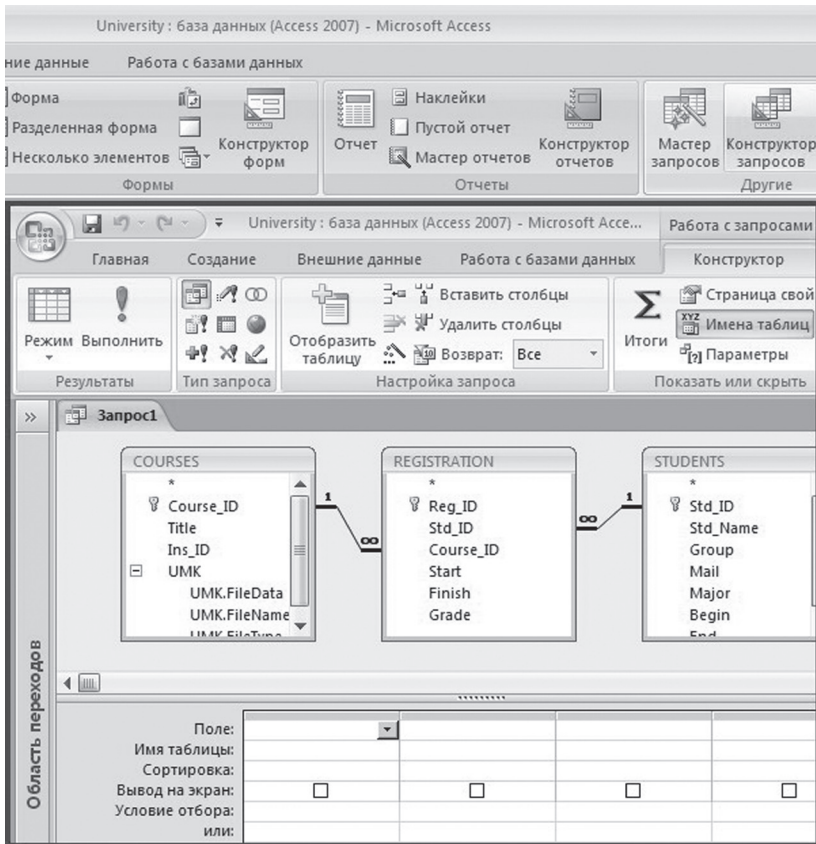


Рис. 26. Запуск конструктора запросов

В результате выполненных действий **Access** переходит в режим **Конструктора запросов**; при этом рабочая область окна разделяется на две области:

- В верхней области отображаются добавленные таблицы и связи между ними
- В нижней области отображается бланк для конструирования запроса

### 2.2.2. Конструирование запроса

Предположим, что в результате запроса нужно получить таблицу «хвостистов», т.е. студентов с неудовлетворительными итоговыми оценками (рис. 27):

The screenshot shows the Microsoft Access 2007 interface in the 'Конструктор' (Design) view for a query named 'Запрос1'. The design grid is as follows:

Поле:	Std_Name	Std_ID	Course_ID	Title	Grade
Имя таблицы:	STUDENTS	STUDENTS	REGISTRATION	COURSES	REGISTRATION
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					<= 3
или:					

Below the design grid, a preview window titled 'Хвостисты' displays the following data:

Std_Name	Std_ID	Title	Grade
Иванов А.С.	211001	Web-дизайн	3
Иванов А.С.	211001	Базы данных	0
Сидоров В.В.	212001	Базы данных	0

Рис. 27. Конструирование запроса



- Щелкнем ячейку в строке **Имя таблицы** и выберем таблицу из списка таблиц, затем щелкнем ячейку в строке **Поле** и выберем поле из списка полей. Выберем те же поля в тех же таблицах, как и при построении запроса **Успеваемость**
- В строке **Вывод на экран** снимем флажок у поля **Course\_ID**
- В строке **Условие отбора** для поля **Grade** укажем условие отбора неуспевающих студентов
- На вкладке **Конструктор** в группе **Результаты** выберем команду **Выполнить**

Полученную в итоге виртуальную таблицу сохраним под именем **Хвостисты**.

## 2.3. Итоговый запрос

### 2.3.1. Создание запроса

Запросы позволяют не только выбирать записи из таблиц, но и вычислять различные статистические параметры. Например, можно подсчитать среднее значение оценок студентов. Сначала с помощью Конструктора запросов сконструируем запрос (рис. 28):

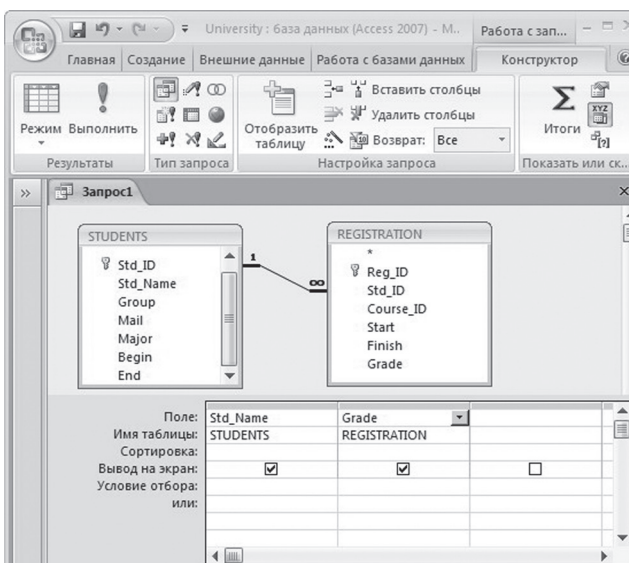


Рис. 28. Конструирование запроса

- 1) На вкладке **Создание** в группе **Другие** выберем команду **Конструктор запросов**
- 2) Появится диалоговое окно **Добавление таблицы**, в котором выберем таблицы **STUDENTS** и **REGISTRATION** и щелкнем кнопку **Добавить**
- 3) В бланке запроса выберем из таблицы **STUDENTS** поле **Std\_Name**, из таблицы **REGISTRATION** – поле **Grade**

### 2.3.2. Добавление групповой операции

На вкладке **Конструктор** в группе **Показать или скрыть** выберем команду **Итоги**. На бланке запроса появится дополнительная строка **Групповая операция** (рис. 29).

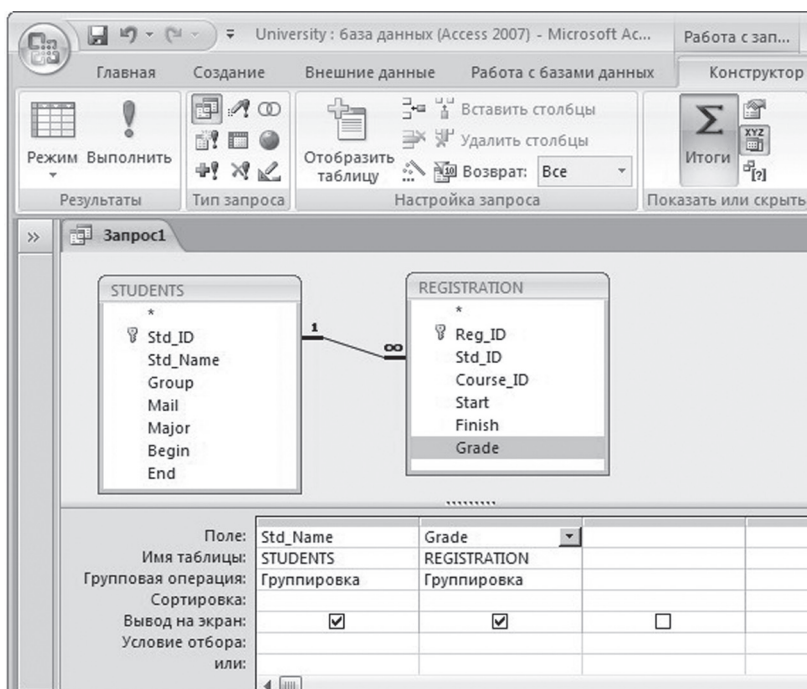


Рис. 29. Добавление групповой операции

### 2.3.3. Выполнение статистической операции

Щелкнем по кнопке со стрелкой в ячейке **Группировка** в столбце **Grade**, чтобы раскрыть список статистических операций:

Операция	Описание
Группировка	Поле, определяющее группу записей, по которой вычисляются статистические параметры
Sum	Сумма значений поля по всем записям
Avg	Среднее значение поля
Min	Минимальное значение поля
Max	Максимальное значение поля
Count	Количество записей, соответствующее поле которых не содержит величины Null
StDev	Стандартное отклонение величины поля от среднего
Var	Вариация значений поля
First	Первое значение в группе записей, по которой вычисляются статистические параметры
Last	Последнее значение в группе записей, по которой вычисляются статистические параметры
Выражение	Вычисляемое поле, значение которого рассчитывается по сложной формуле
Условие	Режим задания условия отбора для поля, по которому не выполняется группировка. Access автоматически делает такое поле скрытым

Выберем операцию **Avg**. На вкладке **Конструктор** в группе **Результаты** выберем команду **Выполнить** (рис. 30).

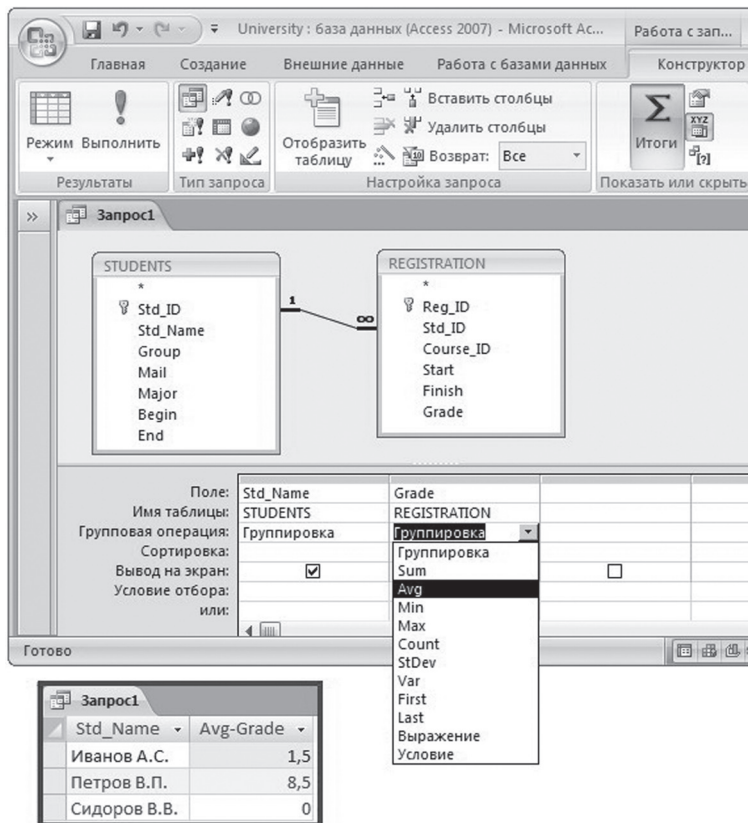


Рис. 30. Выполнение статистической операции

## 2.4. ФИЛЬТРЫ

Фильтры, как и запросы, предназначены для отбора определенных записей из базы данных.

### 2.2.1. Установка фильтра

Предположим, что мы хотим получить список хорошо и отлично успевающих студентов (рис. 31):

- 1) Откроем запрос **Успеваемость** и выделим в нем столбец **Grade**
- 2) На вкладке **Главная** в группе **Сортировка и фильтр** выберем команду **Фильтр**
- 3) На экран будет выведено диалоговое окно, в котором снимем все флажки, кроме оценок 7 и 10
- 4) Щелкнем кнопку **ОК**
- 5) На экран будет выведен отфильтрованный запрос

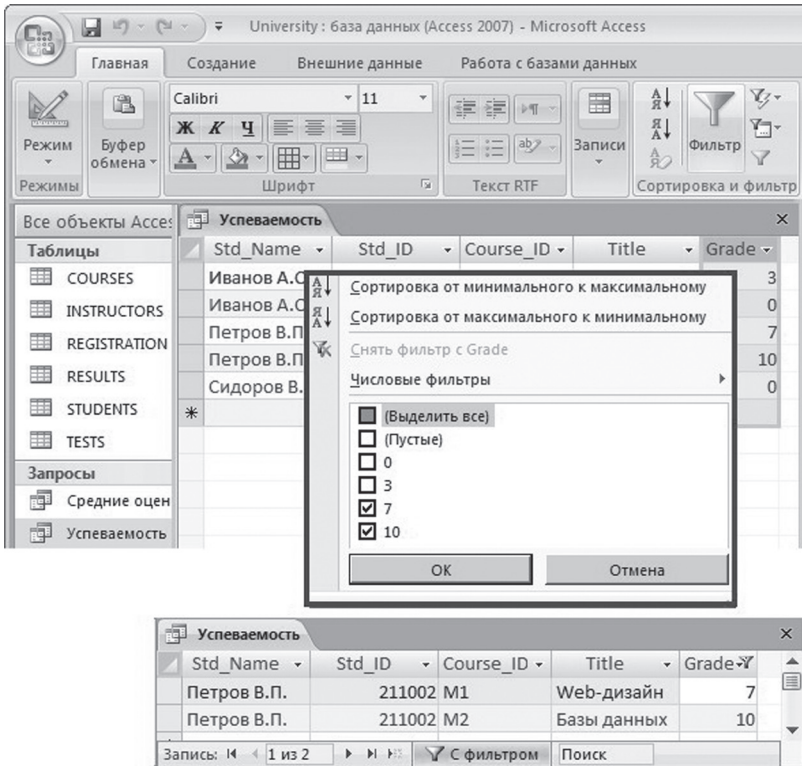


Рис. 31. Установка фильтра

### 2.2.2. Преобразование фильтра в запрос

Фильтр не является объектом базы данных, а сохраняется вместе с таблицей. Назначение нового фильтра для таблицы автоматически удаляет предыдущий фильтр. Если выполняемую фильтром операцию отбора записей нужно выполнять многократно, то фильтр можно превратить в запрос (рис. 32):

- 1) На вкладке **Главная** в группе **Сортировка и фильтр** щелкнем по кнопке со стрелкой **Параметры расширенного фильтра**
- 2) Появится список команд, в котором выберем элемент **Расширенный фильтр**
- 3) На панели быстрого доступа щелкнем по кнопке **Сохранить**
- 4) В открывшемся диалоговом окне **Сохранение в виде запроса** введем имя запроса и щелкнем кнопку **ОК**

В результате выполненных действий будет создан новый запрос.

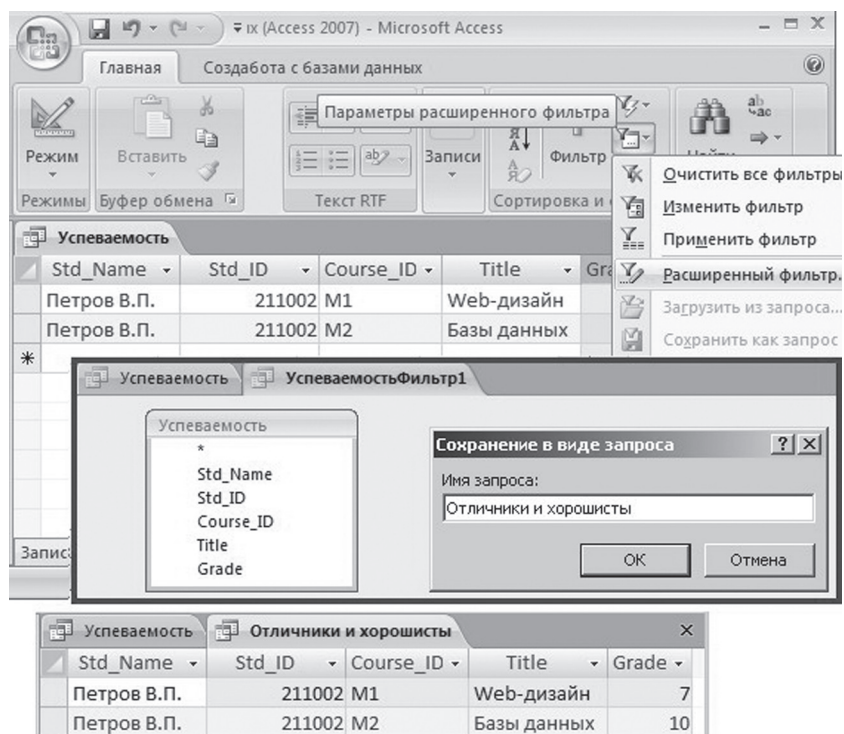


Рис. 32. Преобразование фильтра в запрос

## ГЛАВА 3 ФОРМЫ И ОТЧЕТЫ В ACCESS 2007

### 3.1. ОДНОТАБЛИЧНАЯ ФОРМА

#### 3.1.1. Создание формы

Самым простым и быстрым способом создания формы является ее автоматическое создание на основе выбранной таблицы или запроса (рис. 33):

- 1) Выберем в области переходов таблицу или запрос, например, таблицу **INSTRUCTORS**
- 2) На вкладке **Создание** в группе **Формы** выберем команду **Форма**

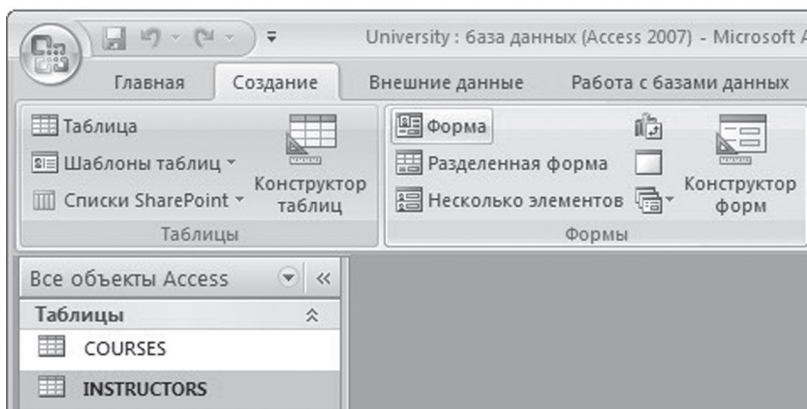


Рис. 33. Выбор команды «Форма»

#### 3.1.2. Редактирование формы

В результате выполненных действий будет создана новая форма, открытая в **Режиме макета**. Этот режим позволяет выполнять редактирование формы, например:

- Изменять размеры элементов формы с помощью перетаскивания мышкой их границ
- Перемещать элементы формы

- Удалять элементы формы
- Изменять надписи и т.п.

Для выполнения действий редактирования формы на ленте появляются две дополнительные вкладки:

- Формат
- Упорядочить

Поскольку таблица INSTRUCTORS связана с таблицей COURSES, на форму также добавлена эта связанная таблица (рис. 34).

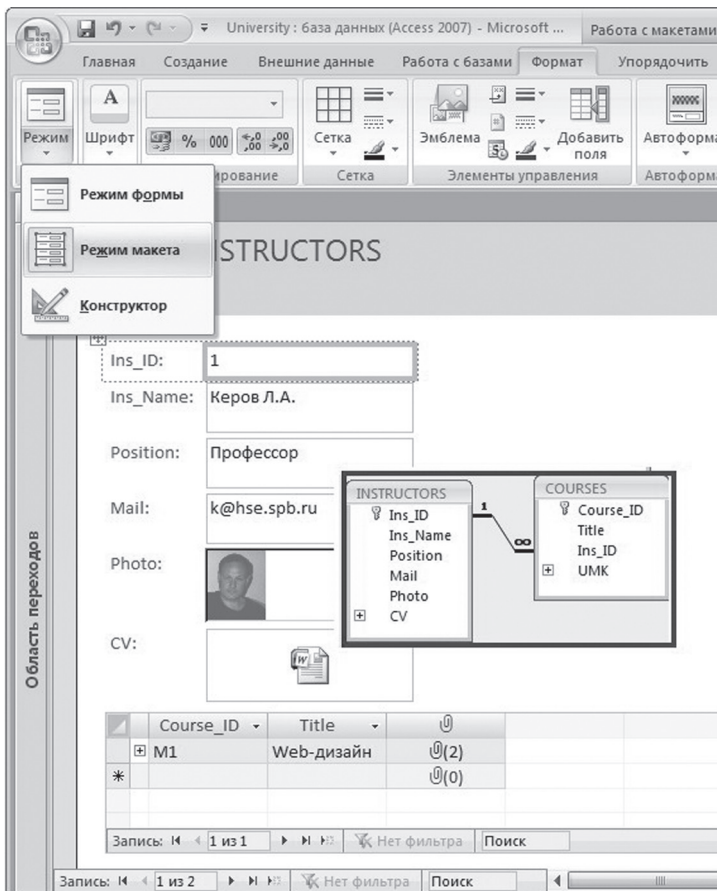


Рис. 34. Редактирование формы



### 3.1.3. Изменение заголовка формы

Для изменения заголовка формы нужно выполнить следующие действия:

- 1) На вкладке **Формат** в группе **Элементы управления** выбрать команду **Заголовок**
- 2) С помощью клавиатуры отредактировать заголовок формы  
Сохраним форму под именем **Список преподавателей** (рис. 35).

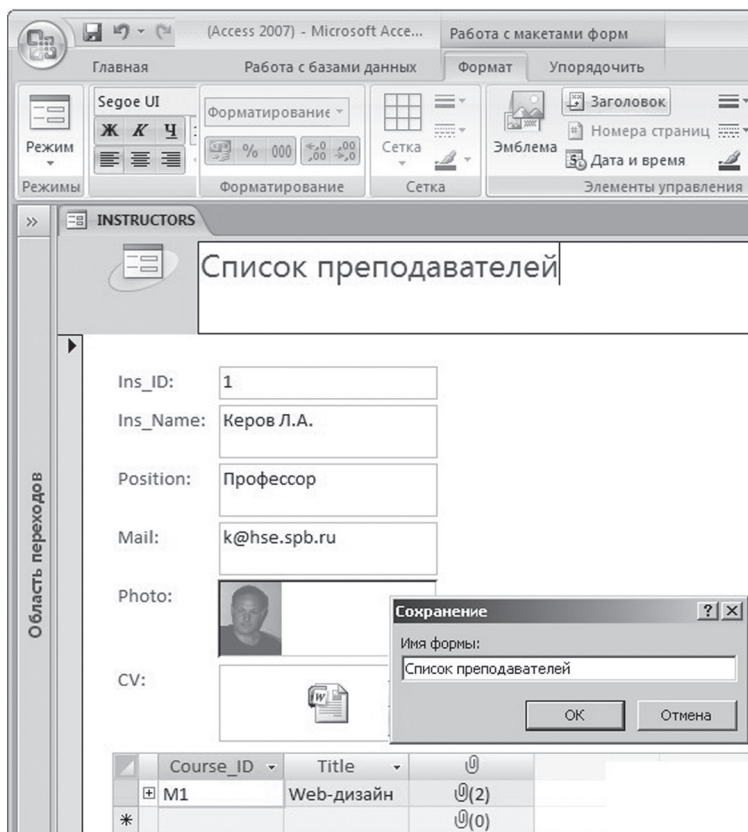


Рис. 35. Изменение заголовка формы

### 3.1.4. Изменение размеров элемента

Для изменения размеров формы нужно выполнить следующие действия (рис. 36):

- 1) На вкладке **Упорядочить** в группе **Макет элемента управления** выбрать команду **В столбик**
- 2) С помощью мыши переместить элемент формы и изменить его размеры

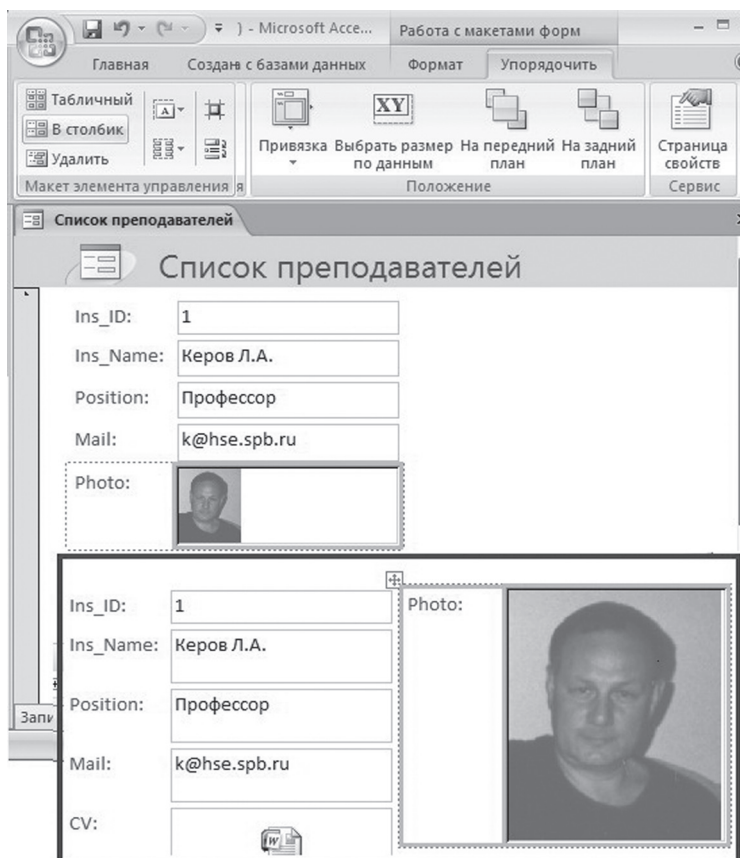


Рис. 36. Изменение размеров элемента

### 3.1.5. Редактирование надписей

Удалим надпись **Photo**, а остальные надписи напишем по-русски (рис. 37).

Course_ID	Title	Link Icon
M1	Web-дизайн	(2)
*		(0)

Рис. 37. Редактирование надписей

### 3.1.6. Режим формы

В **Режиме формы** выполняется просмотр и редактирование данных формы. Если какие-то данные на форме изменены, то при переходе к предыдущей или следующей записи формы эти изменения автоматически вносятся в таблицу, на основе которой построена эта форма (рис. 38).

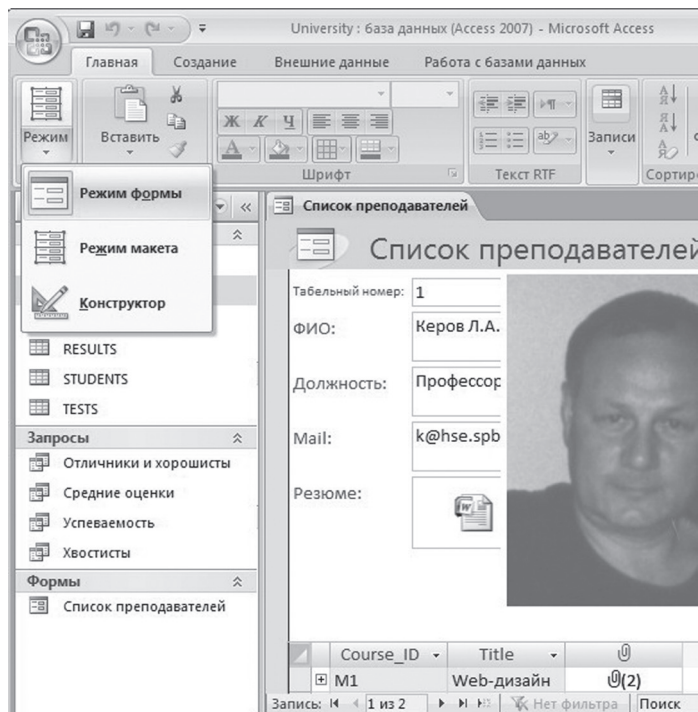


Рис. 38. Режим формы

## 3.2. МАСТЕР ФОРМ

### 3.2.1. Запуск мастера форм

Для запуска **Мастера форм** нужно выполнить следующие действия (рис. 39):

- 1) На вкладке **Создание** в группе **Формы** щелкнем по кнопке **Другие формы**, и в раскрывшемся списке выберем команду **Мастер форм**
- 2) В первом диалоговом окне **Мастера форм** раскроем список **Таблицы и запросы** и выберем нужную таблицу

3) Поля таблицы, которые должны отображаться на форме, перенесем из списка **Доступные поля** в список **Выбранные поля** с помощью кнопки со стрелкой «>»

4) Щелкнем по кнопке **Далее**

Следует отметить, что Мастер форм позволяет размещать на форме поля из нескольких таблиц или запросов. Для этого дополнительную таблицу или запрос нужно выбрать из списка **Таблицы и запросы**, и дополнительные поля нужно перенести в список **Выбранные поля**

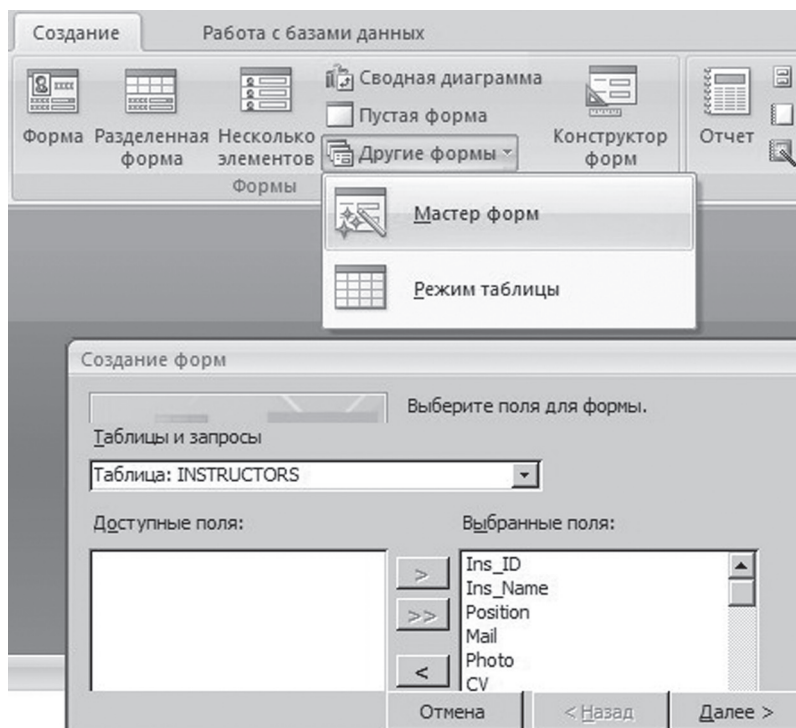


Рис. 39. Запуск мастера форм

### 3.2.2. Выбор внешнего вида формы

Во втором диалоговом окне **Мастера форм** выберем внешний вид формы и щелкнем по кнопке **Далее** (рис. 40).

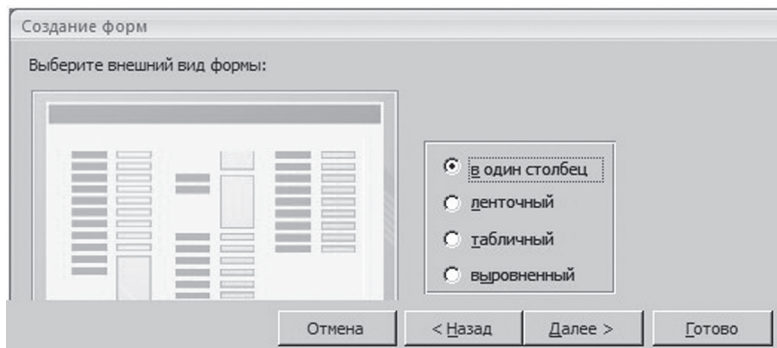


Рис. 40. Выбор внешнего вида формы

### 3.2.3. Выбор стиля формы

В третьем диалоговом окне **Мастера форм** выберем стиль формы и щелкнем по кнопке **Далее** (рис. 41).

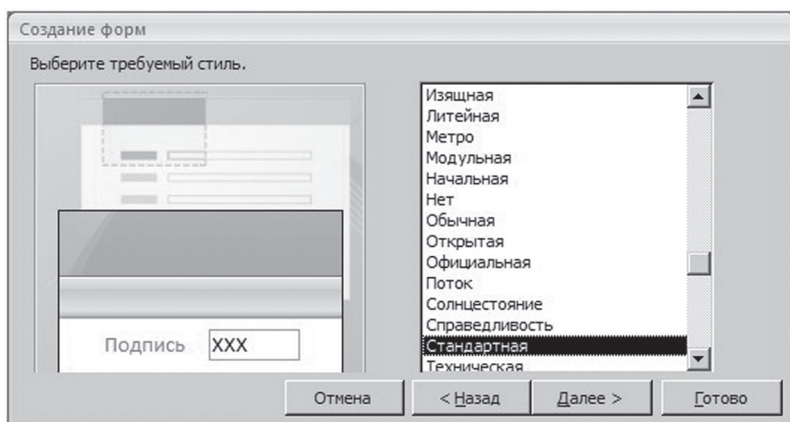


Рис. 41. Выбор стиля формы

### 3.2.4. Задание имени формы

В четвертом диалоговом окне **Мастера форм** зададим имя формы и щелкнем по кнопке **Готово**. Форма будет создана; ее можно редактировать в **Режиме макета** или использовать для просмотра и изменения данных в **Режиме формы** (рис. 42).

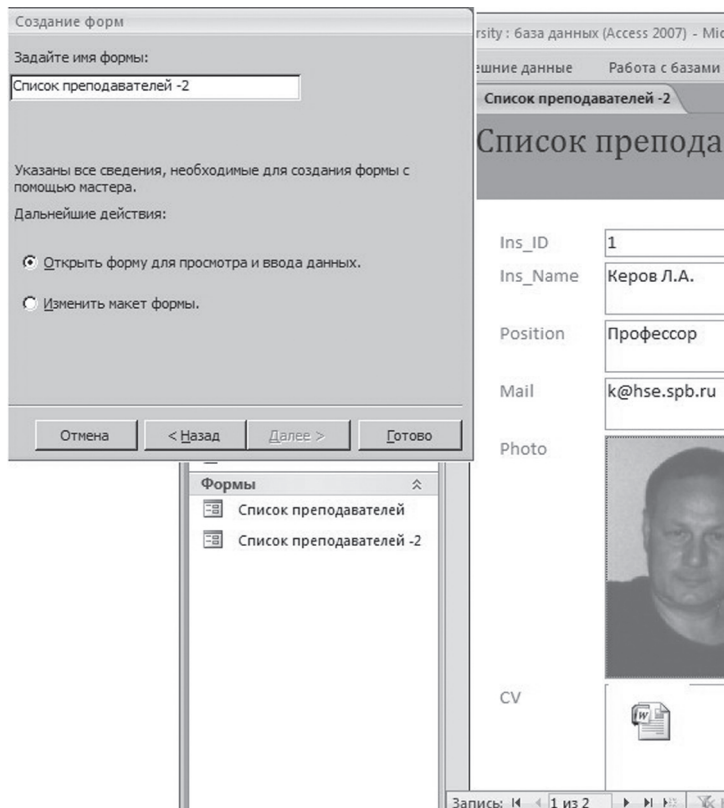


Рис. 42. Задание имени формы

### 3.3. МАСТЕР ОТЧЕТОВ

Отчеты похожи на формы, но они, как правило, предназначены для вывода информации из базы данных на принтер. Поэтому в отчетах данные формируются так, чтобы их было удобно размещать на отдельных страницах.

#### 3.3.1. Запуск мастера отчетов

Для запуска **Мастера отчетов** нужно выполнить следующие действия (рис. 43):

- 1) На вкладке **Создание** в группе **Отчеты** щелкнем по кнопке **Мастер отчетов**
- 2) В первом диалоговом окне **Мастера отчетов** раскроем список **Таблицы и запросы** и выберем нужную таблицу
- 3) Поля таблицы, которые должны отображаться в отчете, перенесем из списка **Доступные поля** в список **Выбранные поля** с помощью кнопки со стрелкой «>»
- 4) Щелкнем по кнопке **Далее**

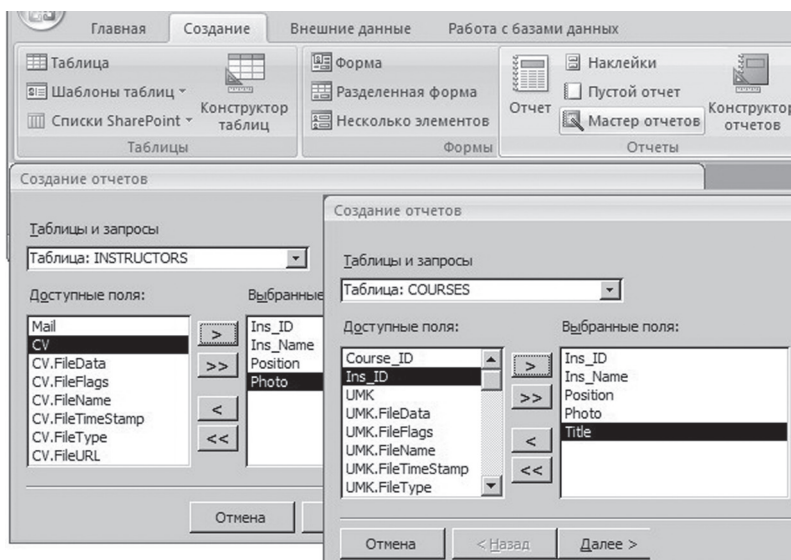


Рис. 43. Запуск мастера отчетов



### 3.3.2. Вид представления данных

В втором диалоговом окне **Мастера отчетов** выберем вид представления данных и щелкнем по кнопке **Далее** (рис. 44).

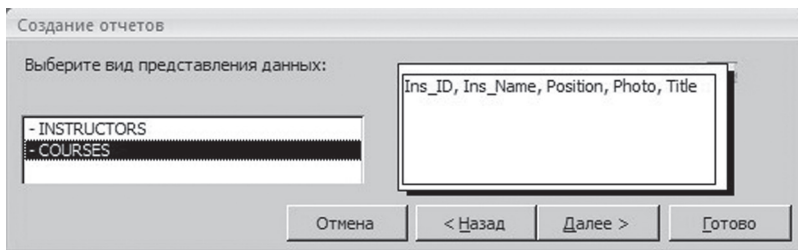


Рис. 44. Выбор вида представления данных

### 3.3.3. Выбор уровней группировки

В третьем диалоговом окне **Мастера отчетов** выберем уровни группировки и щелкнем по кнопке **Далее** (рис. 45).

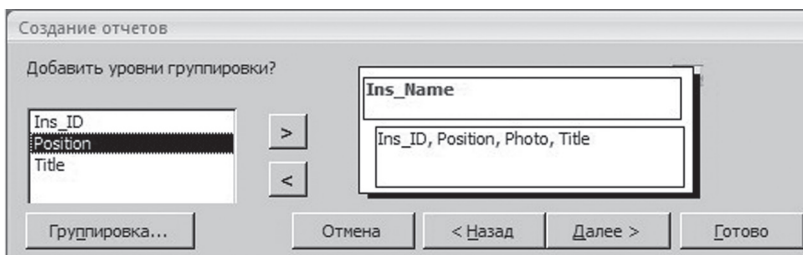


Рис. 45. Выбор уровней группировки

### 3.3.4. Порядок сортировки записей

В четвертом диалоговом окне **Мастера отчетов** выберем порядок сортировки записей и щелкнем по кнопке **Далее** (рис. 46).

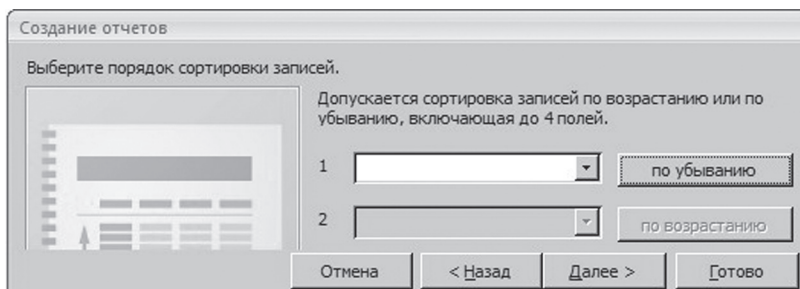


Рис. 46. Выбор порядка сортировки записей

### 3.3.5. Выбор вида макета

В пятом диалоговом окне **Мастера отчетов** выберем вид макета для отчета и щелкнем по кнопке **Далее** (рис. 47).

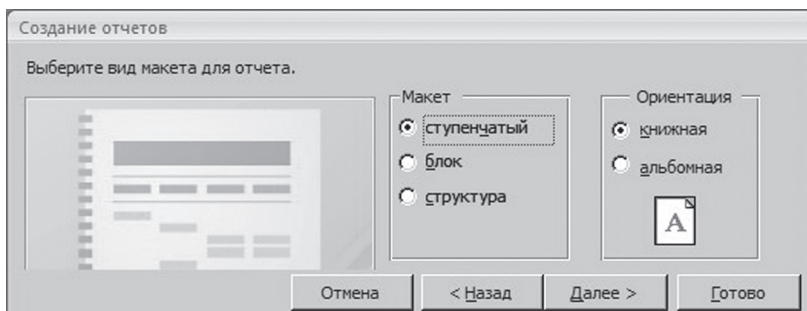


Рис. 47. Выбор вида макета

### 3.3.6. Выбор стиля

В шестом диалоговом окне **Мастера отчетов** выберем требуемый стиль и щелкнем по кнопке **Далее** (рис. 48).

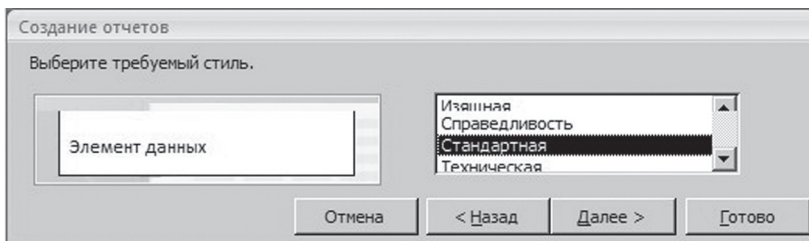


Рис. 48. Выбор стиля отчета

### 3.3.7. Задание имени отчета

В седьмом диалоговом окне **Мастера отчетов** зададим имя отчета и щелкнем по кнопке **Готово** (рис. 49).

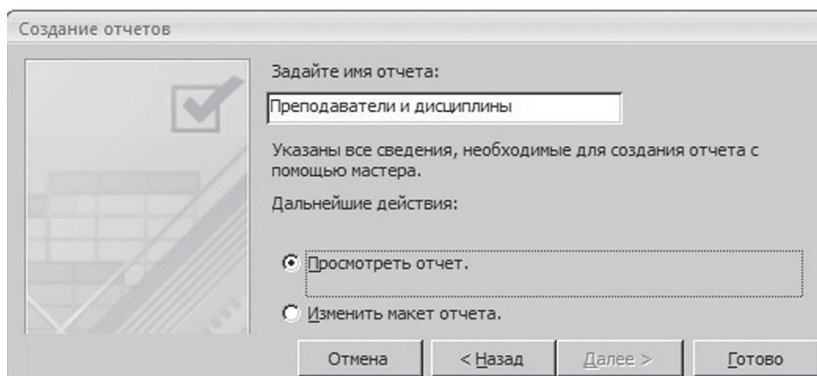


Рис. 49. Завершение создания отчета

### 3.3.8. Редактирование отчета

Если необходимо изменить надписи или размеры областей отчета, то это можно сделать в **Режиме макета** (рис. 50).

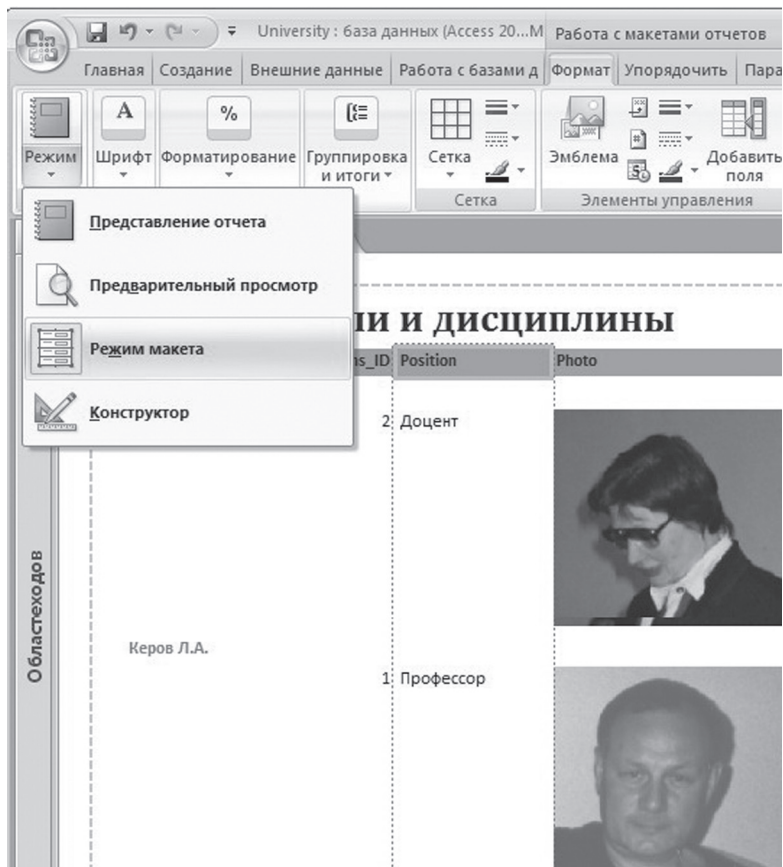
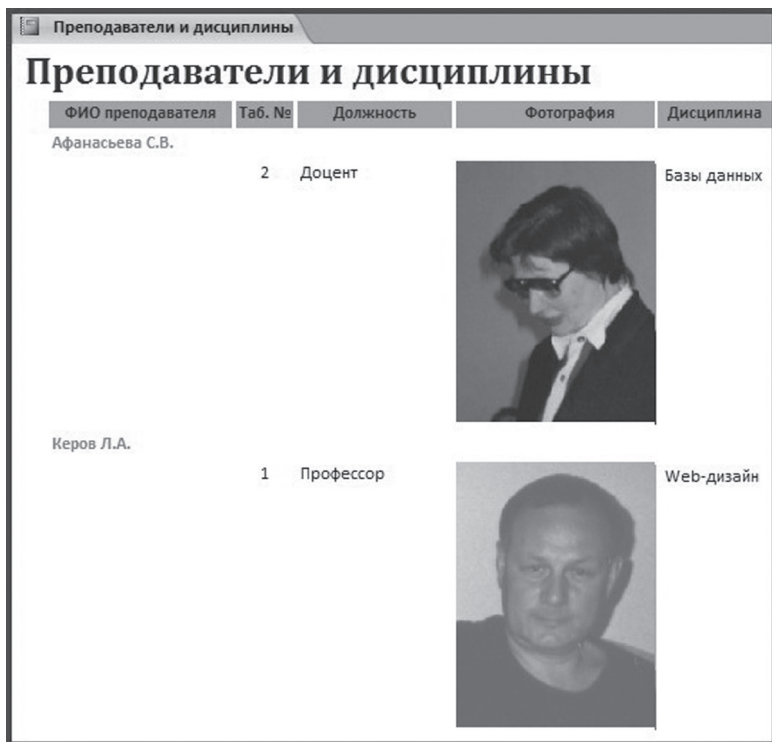


Рис. 50. Редактирование отчета

### 3.3.9. Окончательный вид отчета

Для получения окончательного вида отчета выполним следующие действия (рис. 51):

- Изменим ширину полей, чтобы в них помещались все данные
- Отредактируем заголовки полей, написав их по-русски
- Зададим выравнивание по центру для заголовков полей





ФИО преподавателя	Таб. №	Должность	Фотография	Дисциплина
Афанасьева С.В.	2	Доцент		Базы данных
Керов Л.А.	1	Профессор		Web-дизайн

Рис. 51. Окончательный вид отчета

## СПИСОК ЛИТЕРАТУРЫ

1. Керов Л.А., Афанасьева С.В. Компьютерные технологии офисной деятельности. Учебное пособие. – СПб: Издательство «Ютас», 2010. – 90 с.
2. Стоцкий Ю., Васильев А., Телина И. Office 2007. Изучаем самостоятельно. – СПб.: Питер, 2008. – 524 с.
3. Зиборов В.В. Visual C# 2010 на примерах. – СПб.: БХВ – Петербург, 2011. – 432 с.
4. Рендольф Н., Гарднер Д., Минутилло М., Андерсон К. Visual Studio 2010 для профессионалов: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 1184 с.
5. Уотсон К., Нейгел К., Педерсен Я.-Х., Рид Дж., Скиннер М. Visual C# 2010: полный курс: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 960 с.
6. Харитоновна И.А., Рудикова Л.В. Microsoft Office Access 2007. – СПб.: БХВ-Петербург, 2008. – 1280 с.
7. Chen P.P.-S. The Entity – Relationship Model – Toward a Unified View of Data. – ACM Transactions on Database Systems, 1976, V.1 (March), pp. 9–6.
8. Codd E.F. A Relational Model of Data for Large Shared Data Banks. – Communications of the ACM, 1970, V.12, № 6 (June), pp.377-387
9. Johnson G., Northrup T. Microsoft .NET Framework 2.0 Web-Based Client Development. MCTS Self-Paced Training Kit (Exam 70-528).- Redmond: Microsoft Press, 2007.
10. McFadden F.R., Hoffer J.A. Modern Database Management. Fourth Edition. – Redwood: The Benjamin / Cummings Publishing Company, 1994.
11. Microsoft SQL Server 2000 Database Design and Implementation. MCAD/MCSE, MCDBA Self-Paced Training Kit (Exam 70-229). – Redmond: Microsoft Press, 2001.
12. Stoecker M.A., Stein S.J., Northrup T. Microsoft .NET Framework 2.0 Windows-Based Client Development. MCTS Self-Paced Training Kit (Exam 70-526).- Redmond: Microsoft Press, 2007.

## ГЛОССАРИЙ

### A

**Active Data Objects (ADO)** — интерфейс прикладного программирования, служащий оболочкой для доступа к базам данных.

### M

**Memo** — тип данных, применяемый для столбцов с длинными текстовыми строками (обычно больше 255 символов).

### N

**NULL** — пустое значение, которое может принимать поле, если ему не назначено значение явным способом. NULL не совпадет с нулем или пустой строкой и его нельзя сравнивать с любым другим значением (в том числе с другим пустым значением).

### S

**Structured Query Language (SQL)** — язык структурированных запросов, используемый для извлечения, добавления, изменения и удаления данных в реляционных базах данных.

### U

**Unicode** — стандарт кодирования символов, определяющий набор букв, цифр и символов, позволяет представить алфавиты всех существующих в мире языков.

### A

**Авторизация ~ authorization** — операция проверки разрешений и прав, предоставленных пользователю.

**Аутентификация ~ authentication** — процесс, проверяющий, что пользователь, пытающийся соединиться с сервером, имеет на это право.

### Б

**База данных (БД) ~ database** — совокупность таблиц и других объектов, организованная таким образом, чтобы облегчить поиск, сортировку и комбинирование данных. База данных хранятся в файлах.

**Бизнес-правила ~ business rules** — логика, описывающая предметную область.

## В

**Вариант ~ case** — в технологии извлечения информации — абстрактное представление данных, характеризующееся атрибутами и связями с другими вариантами.

**Внешнее соединение ~ outer join** — соединение, включающее все строки из соединяемых таблиц, для которых выполняются условия поиска (даже те строки одной таблицы, для которых нет соответствующих строк в другой соединяемой таблице). В результирующем наборе для тех строк, которым не находится соответствия в соединяемой таблице, в соответствующих столбцах устанавливаются пустые значения.

**Внешний ключ ~ foreign key (FK)** — столбец или комбинация столбцов, значения которого соответствуют первичному или уникальному ключу из той же или другой таблицы. Внешний ключ также называют ссылочным ключом.

**Внешняя таблица ~ foreign table** — таблица, содержащая внешний ключ.

**Внутреннее соединение ~ inner join** — операция выборки строк из нескольких таблиц на основе сравнения значений из общих для этих таблиц столбцов. Внутреннее соединение исключает строки исходной таблицы, для которых нет соответствующих строк в других исходных таблицах.

**Возможность хранения пустых значений ~ nullability** — атрибут столбца, параметра или переменной, определяющий допускают ли они пустые значения (NULL).

**Выражение ~ expression** — комбинация символов и операторов, дающая в результате вычисления единственное значение. Константы, переменные, столбцы и скалярные функции — примеры простых выражений. Составные выражения представляют собой несколько простых выражений, связанных операторами.

**Вычисляемое поле ~ calculated field** — поле в запросе, отображающее результат вычисления выражения, а не хранимые данные.

**Вычисляемый столбец ~ calculated column** — столбец таблицы, отображающий результат вычисления выражения, а не хранимые данные.

## Г

**Гетерогенные данные ~ heterogeneous data** — данные, хранящиеся в различных форматах, например в базах данных, текстовом файле и электронной таблице.



**Д**

**Диаграмма базы данных ~ database diagram** — графическое представление объектов базы данных. Диаграмма базы данных может быть полной или частичной картиной структуры базы данных. Эта диаграмма включает таблицы, составляющие их столбцы и связи между ними.

**З**

**Запись ~ record** — группа связанных полей (столбцов), рассматриваемая как единое целое.

**Запрос SELECT ~ SELECT query** — запрос, возвращающий результирующий набор, включающий данные из одной или нескольких таблиц. В запросе указываются возвращаемые столбцы, выбираемые строки, порядок следования строк и способ группирования информации.

**Запрос UPDATE ~ UPDATE query** — запрос, изменяющий значения в одном или нескольких столбцах таблицы.

**Запрос определения данных ~ data-definition query** — SQL-запрос, содержащий операторы языка определения данных, которые позволяют создавать или изменять объекты базы данных (такие, как таблицы, индексы, представления и другие) и импортировать объекты базы данных из Microsoft Access.

**Запрос перекрестных табличных данных ~ crosstab query** — отображает итоговые значения, группируя их по двум наборам фактов: один указывая слева, а другой сверху страницы.

**И**

**Иерархия ~ hierarchy** — логическая древовидная структура, в которой каждый элемент имеет один родительский элемент и ноль или более дочерних элементов.

**Изменение данных ~ data modification** — операция добавления, удаления или обновления базы данных, осуществляемая такими операторами, как INSERT, DELETE и UPDATE.

**Индекс ~ index** — объект реляционной базы данных, обеспечивающий быстрый доступ к данным в строках таблице на основе значений ключа. Индексы также обеспечивают уникальность для определенных строк таблицы. Первичный ключ таблицы автоматически индексируется.

**Интерфейс прикладного программирования ~ application programming interface (API)** — набор процедур, доступных приложению, который программист использует при его разработке.

**Информационная модель ~ information model** — объектно-ориентированная схема, определяющая метаданные для описания структуры и поведения приложения, процесса, компонента или программы.

## К

**Каскадное обновление ~ cascading update** — операция обновления значения первичного ключа, на которое указывает внешний ключ в существующих строках другой таблицы. При каскадном обновлении обновляются все значения внешнего ключа так, чтобы они соответствовали новому значению первичного ключа.

**Каскадное удаление ~ cascading delete** — операция удаления строки, содержащей значение первичного ключа, на которую указывает внешний ключ в существующих строках другой таблицы. При каскадном удалении удаляются также все строки, значение внешнего ключа которых указывает на удаленное значение первичного ключа.

**Клиентское приложение ~ client application** — приложение, получающее данные от сервера (из реляционной или многомерной базы данных) и выполняющее собственный анализ и отображение данных.

**Ключ ~ key** — столбец (или группа столбцов) уникально идентифицирующий строку (первичный ключ), определяющий связь между таблицами (внешний ключ) или применяемый при построении индекса.

**Ключевой столбец ~ key column** — столбец, указанный в первичном, внешнем или индексном ключе.

**Кодовая страница ~ code page** — совокупность битовых комбинаций, используемых для представления конкретных букв, цифр и символов (например, 0x20 представляет пробел, а 0x74 символ «t») для символьных и Unicode-данных. Некоторые типы данных для кодировки каждого символа используют один байт, причем каждый байт может иметь 256 битовых комбинаций.

**Компонент доступа ~ provider** — компонент доступа OLEDB, обеспечивающая доступ к базе данных.

**Компонент доступа OLEDB ~ OLEDB provider** — программный компонент, предоставляющий интерфейсы OLEDB. Каждый компонент доступа OLEDB поддерживает работу с источником данных конкретного типа (например, с базой данных SQL Server, Access или электронными таблицами Excel).

**Конкатенация ~ concatenation** — применяется для объединения двух или более текстовых (или двоичных) строк или выражений в единую текстовую (или двоичную) строку или выражение.

**Константа ~ constant** — группа символов, представляющая конкретное значение. Формат константы зависит от типа данных значения. Например, «abc» — символьная строковая константа, «123» — целочисленная константа, «December\_16\_1999» — константа даты/времени, а «0x02FA» — двоичная константа.

**Контрольная точка ~ checkpoint** — момент, когда механизм базы данных записывает измененные страницы на диск. Измененные страницы содержат изменения, не сохраненные на диске. В каждой контрольной точке на диск записываются все страницы, измененные, но не сохраненные на диске с момента последней контрольной точки. Периодичность контрольных точек определяется числом записей в журнале транзакций при модификации данных. Контрольная точка также может иметь место по запросу пользователя или при отключении системы.

## П

**Левое внешнее соединение ~ left outer join** — разновидность внешнего соединения, включающего все строки из таблицы, указанной слева от конструкции JOIN. Когда строки в левой таблице не соответствуют строкам правой таблицы, все строки результирующего набора, полученные из правой таблицы, принимают значение NULL.

**Логические операторы ~ logical operators** — операторы AND, OR и NOT, применяемые для задания условий в конструкциях WHERE.

**Логическое имя ~ logical name** — имя, используемое SQL Server для идентификации файла. Логическое имя должно соответствовать правилам, принятым для идентификаторов, и не должно содержать больше 30 символов.

**Локализация ~ locale** — атрибут операционной системы Windows, определяющий особенности функционирования, связанные с языком. Локализация определяет кодовую страницу с используемыми символами и порядок сортировки символьных строк. Кроме того, локализация определяет формат представления даты и времени и разделитель дробной части десятичных чисел.

**Локальный сервер ~ local server** — экземпляр SQL Server, работающий на том же компьютере, что и приложение.

## Н

**Набор символов ~ character set** — определяет типы символов, которые SQL Server различает в данных типов char, varchar и text. Каждый набор символов включает 256 букв, цифр и символов, применяемых в конкретном языке. Первые 128 значений во всех наборах символов

имеют одинаковые печатные эквиваленты. Остальные 128 символов (их иногда называют дополнительными) уникальны для каждого набора. Набор символов связан, но отличается от Unicode-символов.

## О

**Обновление ~ update** — изменение одного или нескольких значений в строке или строках (обычно оператором UPDATE). Иногда обновлением называют любое изменение данных (включая добавление и удаление).

**Объект ~ object** — один из компонентов базы данных: таблица, запрос, форма и т.п.

**Объект базы данных ~ database object** — компонент базы данных: таблица, запрос, форма и т.п.

**Объектная переменная ~ object variable** — переменная, являющаяся ссылкой на объект.

**Ограничение ~ constraint** — свойство, назначаемое столбцу таблицы, которое позволяет предотвратить занесение недопустимых данных в столбец. Например, ограничения UNIQUE и PRIMARY KEY предотвращают занесение значений, дублирующих существующие. Ограничение CHECK предотвращает занесение значения, не соответствующего критерию поиска, а NOT NULL — значения NULL.

**Ограничение UNIQUE ~ UNIQUE constraint** — ограничение, обеспечивающее сущностную целостность для ключей, не являющихся первичными. Ограничение UNIQUE гарантирует отсутствие повторяющихся значений и создание индекса для увеличения производительности.

**Ограничение на уровне столбца ~ column-level constraint** — ограничение, налагаемое на отдельный столбец при создании или изменении конструкции таблицы.

**Ограничение на уровне таблицы ~ table-level constraint** — ограничения, налагаемые на несколько столбцов при создании или модификации таблицы с целью обеспечения целостности данных.

**Оператор соединения ~ join operator** — оператор сравнения в условии соединения, определяющий, как следует интерпретировать соединяемые сущности и какие строки возвращать.

**Определение данных ~ data definition** — описание атрибутов, свойств и объектов в базе данных.

**Откат ~ rollback** — отмена обновлений, выполненных одной или несколькими незавершенными транзакциями. Обеспечивает целостность базы данных после сбоя.

**Очистка данных ~ data scrubbing** — часть процесса построения хранилища данных из данных, поступающих из нескольких систем оперативной обработки транзакций. В процессе очистки должны обнаруживаться такие ошибки, как неверное написание, различия в соглашениях о правописании двух систем и конфликты данных (например, наличие двух шифров для одного изделия).

## П

**Первичная таблица ~ primary table** — одна из двух таблиц, связанных отношением «один ко многим». Эта таблица должна иметь первичный ключ и все ее записи должны быть уникальны.

**Первичный ключ ~ primary key (PK)** — столбец (или набор столбцов), уникально идентифицирующий все строки таблицы. Первичный ключ не допускает пустых значений. Уникально идентифицировать строки таблицы могут несколько ключей, которые называются потенциальными. Только один из потенциальных ключей является первичным ключом таблицы, остальные называются альтернативными. Хотя наличие первичного ключа в таблице не является обязательным, его определение считается правильным подходом.

**Переменные ~ variables** — сущности, которым присваиваются значения.

**Поддержка принятия решений ~ decision support** — системы, поддерживающие сложный анализ, необходимый для изучения тенденций бизнеса. Информация, предоставляемая такими системами, позволяет менеджерам принимать решения по ведению бизнеса на основе своевременного и точного анализа тенденций его развития.

**Поле ~ field** — область окна или записи, в которой хранится единственное значение. В некоторых базах данных, например в Microsoft Access, слово «поле» используется как синоним слова «столбец».

**Полнотекстовый запрос ~ full-text query** — запрос SELECT для поиска слов, фраз или нескольких форм слов или фраз в символьных столбцах (с типами данных char, varchar, text, ntext, nchar или nvarchar) Возвращает строки, соответствующие заданному критерию.

**Полнотекстовый индекс ~ full-text index** — часть полнотекстового каталога, в которой хранятся слова и их расположение для определенной таблицы.

**Пользовательская база данных ~ user database** — база данных, созданная пользователем для хранения прикладных данных. В большинстве случаев пользователи обращаются только к пользовательским, а не системным базам данных.

**Порядок сортировки ~ sort order** — набор правил, определяющий, как сравниваются между собой символы при выполнении операций сравнения, а также как они сортируются.

**Правила нормализации ~ normalization rules** — совокупность правил проектирования базы данных, призванных минимизировать избыточность данных и облегчающих приложениям и механизму базы данных обеспечение целостности.

**Правое внешнее соединение ~ right outer join** — внешнее соединение, при котором в результат включаются все строки из таблицы, стоящей справа в конструкции JOIN.

**Предоставление ~ grant** — процесс выделения разрешений на выполнение каких-либо действий или на работу с данными пользовательскому учетному имени.

**Проект базы данных ~ database project** — набор из одного или нескольких подсоединений к данным. При создании проекта базы данных можно подсоединиться к одной или нескольким базам данных через ODBC и просматривать их компоненты с помощью утилит с графическим пользовательским интерфейсом.

**Прямое подключение ~ direct connect** — режим подсоединения к базе данных, при котором любые изменения в диаграмме базы данных приводят к автоматическому обновлению базы данных при сохранении диаграммы или выбранных на ней элементов.

**Псевдоним ~ alias** — альтернативное имя таблицы или столбца в выражении, позволяет укоротить имя, избежать неопределенности или применять более понятные имена при выводе результатов.

**Публикация ~ publication** — набор из одной или нескольких статей из одной базы данных. Группирование статей облегчает репликацию логически связанных данных или объектов базы данных.

## Р

**Резервная копия ~ backup** — копия базы данных, журнала транзакций, файла или группы файлов, используемая для восстановления данных после сбоя системы.

**Реляционная база данных ~ relational database** — набор информации, организованный в виде таблиц. Таблица описывает некоторый класс объектов, важных для организации. Например, в базе данных какой-либо компании одна таблица может быть предназначена для сведений о работниках, другая — для сведений о клиентах и еще одна — для сведений о магазинах. Любая таблица состоит из столбцов и строк (в теории реляционных баз данных это атрибуты и кортежи соответ-

ственно). Каждый столбец определяет атрибут класса объектов, представленных таблицей. Например, в таблице для сведений о работниках могут быть столбцы, определяющие такие атрибуты, как имя, фамилия, идентификационный номер, отдел, тарифный разряд и название должности работника. А каждая строка описывает отдельный экземпляр объекта, представленного таблицей. Например, одна из строк таблицы со сведениями о работниках может описывать работника, имеющего идентификационный номер 12345.

**Реляционная СУБД ~ relational database management system (RDBMS)** — система управления базами данных, в которых данные организованы в связанные строки и столбцы.

**Репликация ~ replication** — процесс копирования и распределения данных и объектов их одной базы данных в другую с последующей синхронизацией этой информации.

## С

**Свойство ~ property** — именованный атрибут элемента управления, поля или объекта базы данных, устанавливаемый для определения одной из характеристик объекта (например, размера, цвета или координат на экране) или его поведения (например, отображается ли он на экране).

**Связь ~ relationship** — соединение двух таблиц, при котором внешний ключ одной таблицы указывает на первичный ключ другой таблицы. На диаграмме базы данных связь показана сплошной линией, если обеспечивается ссылочная целостность, или пунктирной линией, если для INSERT и UPDATE ссылочная целостность не обеспечивается.

**Связь «многие ко многим» ~ many-to-many relationship** — связь между двумя таблицами, при которой строкам в одной таблице соответствует множество строк в связанной таблице. Связь «многие ко многим» обеспечивается применением третьей таблицы, называемой связующей, в которой присутствуют первичные ключи из двух соединяемых таблиц.

**Связь «один к одному» ~ one-to-one relationship** — связь между двумя таблицами в реляционной базе данных, при которой одной строке первой таблицы соответствует единственная строка второй таблицы и наоборот.

**Связь «один ко многим» ~ one-to-many relationship** — связь между двумя таблицами в реляционной БД, при которой одной строке первой таблицы соответствует несколько строк второй таблицы, но любой строке второй таблицы может соответствовать только одна строка первой таблицы.

**Секционирование ~ partitioning** — процесс замены таблицы несколькими меньшими таблицами. Каждая меньшая таблица имеет тот же формат, что и исходная, но содержит лишь некоторое подмножество данных. Строки помещаются в новые таблицы в соответствии с некоторыми признаками, например в соответствии с некоторым диапазоном ключей. Секционирование позволяет увеличить производительность, уменьшить риск возникновения конфликтов при репликации.

**Сертификат ~ certificate** — набор данных, используемых для аутентификации и защищенного обмена информацией в общедоступных сетях, таких, как Интернет. Сертификат связывает открытый криптографический ключ с владельцем соответствующего закрытого ключа. Сертификаты подписываются цифровой подписью центра сертификации, выдавшего сертификат, и могут использоваться на уровне пользователя, компьютера или службы.

**Символьный формат ~ character format** — данные, хранящиеся в файле массового копирования данных с использованием текстовых символов.

**Системный администратор ~ system administrator** — лицо или группа лиц, ответственных за управление экземпляром SQL Server. Системные администраторы имеют полный доступ ко всем функциям экземпляра SQL Server. Системные администраторы или являются членами фиксированной серверной роли sysadmin или регистрируются с идентификатором sa.

**Соединение ~ join** — процесс объединения содержимого двух или более таблиц и создания результирующего набора, включающего строки и столбцы каждой таблицы. Таблицы обычно соединяются посредством общих данных. Этот термин также используется для обозначения процесса или результата соединения таблиц (например, внутреннее соединение), указывая на конкретный метод соединения таблиц.

**Соединяющая таблица ~ junction table** — таблица, определяющая связь между другими таблицами. Содержит внешние ключи, указывающие на связываемые таблицы.

**Соединяющий столбец ~ join column** — столбец, указываемый в условии соединения.

**Список выбора ~ select list** — часть оператора SELECT, в которой определены столбцы результирующего набора, возвращаемого этим оператором. Список выбора представляет собой список выражений, разделенных запятыми. В качестве выражений могут быть имена столбцов, функции или ограничения.



**Статья ~ article** — объект, подлежащий репликации. Статья является компонентом подписки и может быть таблицей, определенными с помощью фильтра столбцами или строками, определением хранимой процедуры или представления, запросом на выполнение хранимой процедуры, представлением, индексированным представлением или пользовательской функцией.

**Степень детализации ~ granularity** — определяет, насколько подробная информация содержится в элементах данных. Таблица фактов с высокой степенью детализации содержит множество отдельных фактов, таких, как отдельные транзакции, связанные с продажами. В таблицах с меньшей степенью детализации хранятся обобщенные факты, например суммарные дневные продажи.

**Столбец ~ column** — область в каждой строке таблицы, хранящая значение некоторого атрибута объекта, моделируемого данной таблицей.

**Столбец с уникальным идентификатором ~ identity column** — столбец таблицы, которому назначено свойство IDENTITY, позволяющее генерировать неповторяющиеся числа.

**Страницы свойств ~ property pages** — диалоговые окна, имеющие вкладки, на которых определяются характеристики таблиц, связей, индексов, ограничений и ключей. Каждый объект на диаграмме базы данных имеет совокупность свойств. Страница свойств отображает свойства только выбранного объекта. Если выбрано несколько объектов, на странице свойств показаны свойства первого из выбранных объектов.

**Строка ~ row** — совокупность элементов таблицы, представляющая один экземпляр объекта, моделируемого таблицей.

**Строковые функции ~ string functions** — функции, выполняющие операции над символьными и двоичными строками.

**Сущность ~ entity** — реальный объект, обозначаемый существительным (человек, место, вещь, понятие). Сущности являются семантическими объектами.

**Схема ~ schema** — совокупность объектов базы данных, владельцем которых является один пользователь и которые формируют единое пространство имен. Пространство имен — это набор объектов, имена которых не могут повторяться. Например, две таблицы могут иметь одно и то же имя, только если они принадлежат разным схемам.

**Схема базы данных ~ database schema** — имена таблиц, полей, типов данных, первичных и внешних ключей базы данных; известна также как структура базы данных.

**Сценарий ~ script** — набор операторов для выполнения некоторой операции.

**Сценарий базы данных ~ database script** — набор операторов для создания объектов базы данных.

**Сценарий изменений ~ change script** — текстовый файл, содержащий SQL-операторы, выполняющие изменения в базе данных.

## Т

**Таблица ~ table** — двумерный объект, состоящий из строк и столбцов, который используется для хранения данных в реляционных базе данных. Каждая таблица хранит информацию об одном типе объектов, моделируемых базой данных. Например, база данных учебного заведения имеет одну таблицу для преподавателей, вторую — для учащихся и третью — для занятий. Столбцы таблицы представляют атрибуты моделируемого объекта (например, имя, фамилию и адрес). Каждая строка представляет один экземпляр моделируемого объекта.

**Таблица измерений ~ dimension table** — таблица в хранилище данных, запись которой описывают данные в таблице фактов. Таблицы измерений содержат данные, на основании которых создаются измерения.

**Таблица фактов ~ fact table** — центральная таблица в схеме хранилища данных, содержащая числовые значения мер и ключей, связывающих факты с таблицами измерений. Таблицы фактов содержат данные, описывающие конкретные события в бизнесе, такие, как банковские транзакции или продажи.

**Тип данных ~ data type** — атрибут, указывающий, какого типа информация может храниться в столбце, параметре или переменной.

**Точность ~ precision** — максимальное число десятичных разрядов числа (как слева, так и справа от десятичной точки).

**Транзакция ~ transaction** — группа операций над базой данных, рассматриваемая как единая логическая единица выполнения, которая или целиком фиксируется или откатывается.

**Триггер ~ trigger** — хранимая процедура, исполняемая при модификации данных в определенной таблице. Триггеры часто применяют для обеспечения ссылочной целостности или согласованности логически связанных данных в разных таблицах.

## У

**Умолчание ~ default** — значение данных устанавливаемое автоматически, если пользователь не указал соответствующее значение. Умолчанием также называется действие, предпринимаемое автоматически в определенных ситуациях, если пользователь не определил конкретное действие для данной ситуации

**Уникальный индекс ~ unique index** — индекс, не допускающий наличия более одной строки с одним значением индекса и, следовательно, повторяющихся значений индекса или ключа. Система проверяет повторяющиеся значения ключа при создании индекса и при выполнении операторов INSERT и UPDATE.

**Условие соединения ~ join condition** — условие, определяющее, как таблицы связаны через соединяющие столбцы.

**Устаревание кэша ~ cache aging** — механизм кэширования, позволяющий определить, когда строка кэша устарела и ее нужно обновить.

**Устройство резервного копирования ~ backup device** — лента или диск, используемый в операциях резервного копирования и восстановления.

## Ф

**Файл ~ file** — основное средство хранения данных.

**Файл базы данных ~ database file** — один из физических файлов, из которых состоит БД.

**Файл данных ~ data file** — в операциях массового копирования — файл, передающий данные.

**Файл журнала транзакций ~ log file** — файл или совокупность файлов, содержащих сведения об изменениях, сделанных в базе данных.

**Файл резервной копии ~ backup file** — файл, в котором хранится полная или частичная резервная копия базы данных, журнала транзакций файла или группы файлов.

**Факт ~ fact** — строка в таблице фактов хранилища данных. Факт содержит значения, связанные с некоторыми событиями (например, с продажами).

**Фильтр ~ filter** — совокупность критериев, определяющих состав возвращаемого набора данных. Фильтры также определяют последовательность, в которой возвращаются строки данных.

**Фильтр по столбцам ~ column filter** — определяет столбцы, которые будут использоваться при репликации моментальных снимков, транзакций или при репликации сведением.

**Фильтр соединения ~ join filter** — фильтр для строк, применяемый в репликации сведением, который определяет связь между таблицами, которую необходимо учитывать при синхронизации, что аналогично определению соединения между двумя таблицами.

**Фрагментация ~ fragmentation** — возникает при модификации данных. Фрагментацию можно уменьшить и увеличить производительность упреждающего чтения, удалив и создав заново кластерный индекс.

**Функция ~ function** — фрагмент кода, выполняющийся как единая логическая единица. К функциям обращаются по имени, им можно передавать входные параметры и они могут возвращать информацию о состоянии или необязательные выходные параметры. Функции применяются во многих языках программирования.

## Х

**Хранилище данных ~ data warehouse** — база данных, структура которой оптимизирована для аналитических запросов. Хранилище данных обычно содержит данные, описывающие историю ведения бизнеса организации.

**Хранимая процедура ~ stored procedure** — набор скомпилированных операторов языка SQL, которому присвоено имя и который обрабатывается как единое целое.

## Ц

**Целостность данных ~ data integrity** — состояние, при котором все данные, хранимые в базе данных, являются корректными. Если в базе данных хранится неверное значение, говорят, что база данных утратила целостность данных.

## Ч

**Числовое выражение ~ numeric expression** — любое выражение, являющееся комбинацией переменных, констант, функций и операторов, в результате вычисления которого получается число.

## Ш

**Шифрование ~ encryption** — метод преобразования конфиденциальной информации в формат, недоступный для чтения.

## Э

**Экземпляр ~ instance** — одна из копий SQL Server, работающая на компьютере. На одном компьютере может работать несколько экземпляров SQL Server.

## Я

**Язык HTML ~ Hypertext Markup Language (HTML)** — метод разметки документа с помощью тэгов для публикации его в Web. Для просмотра HTML-документов применяются Web-браузеры, например Microsoft Internet Explorer.

**Язык базы данных ~ database language** — язык, используемый для доступа, выборки, обновления и управления данными в реляционной СУБД. Наиболее распространенным языком баз данных является SQL.

**Язык манипулирования данными ~ data manipulation language (DML)** — подмножество SQL-операторов для получения и манипулирования данными.

**Язык определения данных ~ data definition language (DDL)** — язык (обычно являющийся частью СУБД), используемый для описания всех атрибутов и свойств базы данных, таких, как спецификация строк, определение столбцов, ключевые столбцы (иногда и метод применения ключей), расположение файлов и стратегия хранения данных.

**Язык управления доступом к данным ~ data control language (DCL)** — подмножество SQL-операторов, используемых для управления разрешениями для объектов базы данных.

**Ячейка ~ cell** — совокупность свойств куба (включая значение), определенная конъюнкцией единичных элементов, выбранных из каждого измерения.

## ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

### Б

- база данных 6
- бизнес-правило 6

### Г

- групповая операция 39

### Д

- данные для тестов 23
- добавление связи 26

### З

- запись 8
- запрос
  - итоговый 38
  - хвостисты 36

### К

- каскадное
  - обновление связанных полей 26
  - удаление связанных полей 26
- ключ
  - внешний 20
  - первичный 12
  - простой 12
  - составной 12
- команда «Выполнить» 40
- конструктор запросов
  - запуск 35
  - конструирование запроса 36
- концептуальная модель 8

### М

- мастер запросов
  - выбор таблиц и полей 31
  - выбор типа отчета 32

- запуск 30
- редактирование запроса 33
- результат запроса 33
- указание имени запроса 32

**О**

- обеспечение целостности данных 26
- область переходов 14
- операция
  - групповая 39
  - статистическая 40
- отображение подчиненных таблиц 28
- отчет
  - вид представления данных 54
  - выбор вида макета 55
  - выбор стиля 55
  - выбор уровней группировки 54
  - задание имени 56
  - запуск мастера 53
  - окончательный вид
  - порядок сортировки записей 54
  - редактирование 56

**П**

- поле
  - «Вложение» 18
  - объекта OLE 17
  - таблицы 8

**Р**

- редактирование
  - запроса 33
  - надписей 48
  - отчета 56
  - формы 44
- реляционная база данных 6
- режим
  - макета 44
  - таблицы 13
  - формы 48

**С**

- свойства поля таблицы 15
- система управления данными 6

## создание

- базы данных 9
- итогового запроса 38
- строки таблицы 14
- структуры таблицы 10
- формы 44

## сущность 7

схема данных 25, 28

**Т**

## таблица

- COURCES 20
- INSTRUCTORS 17
- REGISTRATION 21
- RESULTS 24
- STUDENTS 10
- TESTS 22

типы данных СУБД ACCESS 10

**Ф**

## форма

- выбор внешнего вида 51
- выбор стиля 51
- задание имени 52
- запуск мастера 50
- изменение заголовка 46
- изменение размеров элемента 47
- редактирование надписей 48
- редактирование формы 48
- режим макета 44
- режим формы 48
- создание 44

## фильтр

- установка 41
- преобразование в запрос 43
- редактирование 46





*Л. А. Керов*

**ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ  
ТЕХНОЛОГИИ В УПРАВЛЕНЧЕСКОЙ ПРАКТИКЕ**

Часть 2. Базы данных

---

Подписано в печать 11.05.2012.  
Гарнитура Муриад Pro. Печать цифровая.  
Тираж 70 экз. Заказ 275.

---

Подготовлено к печати и отпечатано  
Отделом оперативной полиграфии  
НИУ ВШЭ — Санкт-Петербург.  
198099, Санкт-Петербург, ул. Промышленная, д. 17а  
Тел./факс (812) 786-58-95