



# ОСНОВЫ РАБОТЫ С БАЗАМИ ДАННЫХ ACCESS 2007

Учебное пособие

КРАСНОЯРСК 2011

Федеральное государственное образовательное учреждение  
высшего профессионального образования  
«СИБИРСКИЙ ЮРИДИЧЕСКИЙ ИНСТИТУТ  
МИНИСТЕРСТВА ВНУТРЕННИХ ДЕЛ РОССИЙСКОЙ ФЕДЕРАЦИИ  
(г. Красноярск)»

С.Н.Ефимов  
В.В.Молоков  
А.С.Шерстяных

## ОСНОВЫ РАБОТЫ С БАЗАМИ ДАННЫХ ACCESS 2007

Учебное пособие

Красноярск  
СибЮИ МВД России  
2011

УДК 681.3  
ББК 32.973

Рецензенты: В.Д. Лаптенко – д.т.н., профессор, заведующий кафедрой ИУС Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнева;  
В.А. Лапко – д.т.н., профессор, ведущий научный сотрудник института вычислительного моделирования СО РАН

**Ефимов, С.Н.**

Основы работы с базами данных Access 2007 : учебное пособие / С.Н.Ефимов, В.В.Молоков, А.С.Шерстяных; Сибирский юридический институт МВД России. – Красноярск: СибЮИ МВД России, 2011. – 92 с.

В пособии рассматриваются основные вопросы разработки баз данных при помощи Microsoft Office Access 2007, в том числе: основные понятия баз данных, создание таблиц, форм, запросов, отчетов, макросов и модулей, подготовка документов к печати.

Предназначено для курсантов, слушателей и студентов Сибирского юридического института МВД России.

© Сибирский юридический институт МВД России, 2011  
© С.Н.Ефимов, В.В.Молоков, А.С.Шерстяных, 2011

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ БАЗ ДАННЫХ .....	6
1.1. Классификация баз данных .....	6
1.2. Структурные элементы базы данных.....	7
1.3. Модели данных .....	8
1.4. Свойства СУБД и базы данных .....	12
1.5. Обзор СУБД.....	12
1.6. База данных Access .....	14
2. ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ .....	17
2.1. Разработка макета базы данных .....	17
2.2. Создание новой базы данных .....	22
2.3. Создание таблицы путем ввода данных.....	23
2.4. Создание таблицы в режиме конструктора .....	25
2.5. Создание таблицы Адреса сотрудников .....	29
2.6. Создание таблицы Родственники сотрудника.....	29
2.7. Схема данных .....	29
3. ФОРМЫ .....	33
3.1. Мастер форм.....	33
3.2. Конструктор форм.....	35
3.3. Подчиненная форма .....	42
4. ЗАПРОСЫ.....	45
4.1. Мастер запросов .....	45
4.2. Конструктор запросов.....	47
4.3. Условия отбора и вычисляемые поля .....	49
5. ОТЧЕТЫ .....	54
5.1. Мастер отчетов .....	54
5.2. Конструктор отчетов.....	55
5.3. Сортировка и группировка .....	58
5.4. Подведение промежуточных итогов .....	61
6. МАКРОСЫ И МОДУЛИ .....	65
6.1. Макросы.....	65
6.2. Модули.....	67
7. ИЗМЕНЕНИЕ БАЗЫ ДАННЫХ.....	70
8. СОЗДАНИЕ КНОПОЧНОЙ ФОРМЫ.....	74
9. ВЫВОД РЕЗУЛЬТАТОВ.....	78
9.1. Предварительный просмотр .....	78
9.2. Файловые операции .....	79
10. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ.....	81
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ .....	92

## ВВЕДЕНИЕ

Современные информационные системы, основанные на концепции интеграции данных, характеризуются огромными объемами хранимой информации, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

*Информационная система* – система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.

Цель любой информационной системы – обработка данных об объектах реального мира. Основой информационной системы является база данных. В широком смысле слова база данных – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под предметной областью принято понимать часть реального мира, подлежащая изучению для организации управления и, в конечном счете, автоматизации, например предприятие, вуз и т.п.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным их сочетанием. Большое значение при этом приобретает структурирование данных.

*Структурирование данных* – это введение соглашений о способах представления данных.

*Неструктурированными* называют данные, записанные, например, в текстовом файле.

Ниже приведен пример неструктурированных и структурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя, отчество и год рождения).

### **Неструктурированные данные:**

Личное дело № 16493. Сергеев Петр Михайлович, дата рождения 1 января 1992 г.; л/д № 16593, Петрова Анна Владимировна, дата рожд. 15 марта 1991 г.; № личн. дела 16693, д.р. 14.04.92, Анохин Андрей Борисович.

Легко убедиться, что сложно организовать поиск необходимой информации, хранящейся в неструктурированном виде.

## Структурированные данные:

№ лично-го дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.92
16593	Петрова	Анна	Владимировна	15.03.91
16693	Анохин	Андрей	Борисович	14.04.92

Чтобы автоматизировать поиск и систематизировать эти данные необходимо выработать определенные соглашения о способах их представления, например, дату рождения нужно записывать одинаково для каждого студента, она должна иметь одинаковую длину и определенное место среди остальной информации. Эти же замечания справедливы и для остальных данных (номер личного дела, фамилия, имя, отчество).

Пользователями базы данных могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые конечными пользователями.

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария – системы управления базами данных.

*База данных (БД)* – это поименованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области, систематизированная для обработки с помощью ЭВМ.

*Объектом* называется элемент предметной области, информацию о котором мы сохраняем.

Объект может быть реальным (например, человек, изделие или населенный пункт) и абстрактным (например, событие, счет покупателя или изучаемый студентами курс). Так, в области продажи автомобилей примерами объектов могут служить МОДЕЛЬ АВТОМОБИЛЯ, КЛИЕНТ и СЧЕТ. На товарном складе – это ПОСТАВЩИК, ТОВАР, ОТПРАВЛЕНИЕ и т.д.

*Система управления базами данных (СУБД)* – это комплекс программных и языковых средств, предназначенных для создания, ведения и совместного применения баз данных многими пользователями.

Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данных, хранимых в базе.

# 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ БАЗ ДАННЫХ

## 1.1. КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

По *технологии обработки данных* базы данных подразделяются на централизованные и распределенные.

*Централизованная база данных* хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен многопользовательский доступ к базе. Такой способ использования баз данных часто применяют в компьютерных локальных сетях.

*Распределенная база данных* состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых на различных компьютерах вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

По *способу доступа к данным* базы данных разделяются на базы данных с локальным доступом и базы данных с удаленным (сетевым) доступом.

Системы централизованных баз данных с сетевым доступом предполагают различные архитектуры подобных систем:

- файл-сервер;
- клиент-сервер.

**Файл-сервер.** Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут также создавать на рабочих станциях локальные БД, которые используются ими монопольно.

**Клиент-сервер.** В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и

извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL.

## 1.2. СТРУКТУРНЫЕ ЭЛЕМЕНТЫ БАЗЫ ДАННЫХ

Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, таблица (рис.1.1).



Рис.1.1. Основные структурные элементы базы данных

**Поле** – элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. Для описания поля используются следующие характеристики:

- *имя* (например, фамилия, имя, отчество, дата рождения);
- *тип* (например, символьный, числовой, календарный);
- *длина* (например, 15 байт, причем длина будет определяться максимально возможным объемом хранимых данных);
- *точность* (для числовых данных, например, два десятичных знака для отображения дробной части).

**Запись** – совокупность логически связанных полей.

**Экземпляр записи** – отдельная реализация записи, содержащая конкретные значения ее полей.

**Таблица** – совокупность экземпляров записей одной структуры.

В структуре записи таблицы указываются поля, значения которых являются ключами: первичными (ПК) и вторичными (ВК).

**Первичный ключ (ПК)** – это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей – составным ключом.

**Вторичный ключ (ВК)** – это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение вторичного ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один-единственный экземпляр записи, то по вторичному – несколько.

### 1.3. МОДЕЛИ ДАННЫХ

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.

Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную.

**Иерархическая модель** организует данные в виде древовидной структуры.

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. Дерево представляет собой иерархию элементов, называемых узлами. Узел – это совокупность атрибутов данных, описывающих некоторый объект. На самом верхнем уровне иерархии имеется один и только один узел – корень. Каждый узел, кроме корня, связан с одним узлом на более высоком уровне, называемом исходным для данного узла. Ни один элемент не имеет более одного исходного. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне. Они называются порожденными (рис.1.2).

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рисунка 1.2, для записи С4 путь проходит через записи А и В3 соответственно.

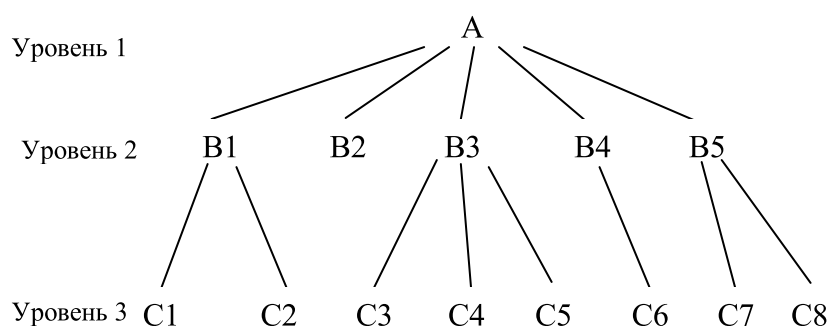


Рис.1.2. Графическое изображение иерархической структуры БД

**Сетевая модель** является расширением иерархического подхода к организации данных. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков.

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом. На рисунке 1.3 изображена сетевая структура базы данных в виде графа.

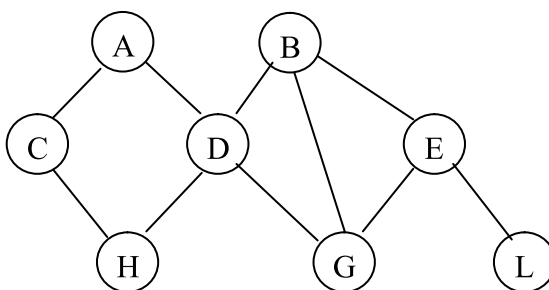


Рис.1.3. Графическое изображение сетевой структуры

### **Реляционная модель данных.**

Понятие «реляционный» (англ. relation – отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структурой 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. Например, реляционной табли-

цей можно представить информацию о студентах, обучающихся в вузе (рис.1.4).

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.92	111
16593	Петрова	Анна	Владимировна	15.03.91	112
16693	Анохин	Андрей	Борисович	14.04.92	111

Рис.1.4. Пример реляционной таблицы

Реляционная модель данных поддерживает совокупность взаимосвязанных двумерных таблиц – объектов модели.

Связи между двумя логически связанными таблицами в реляционной модели устанавливаются по равенству значений одинаковых атрибутов этих 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.

При описании реляционной модели часто используют следующие термины: отношение, кортеж, домен.

Отношения представлены в виде таблиц, строки которых соответствуют записям (кортежам), а столбцы – полям, атрибутам отношений (доменам).

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет составной ключ.

В примере, показанном на рисунке 1.5, ключевым полем таблицы является «№ личного дела».

Между двумя реляционными таблицами могут быть сформированы связи. Различные таблицы могут быть связаны между собой через общее поле данных.

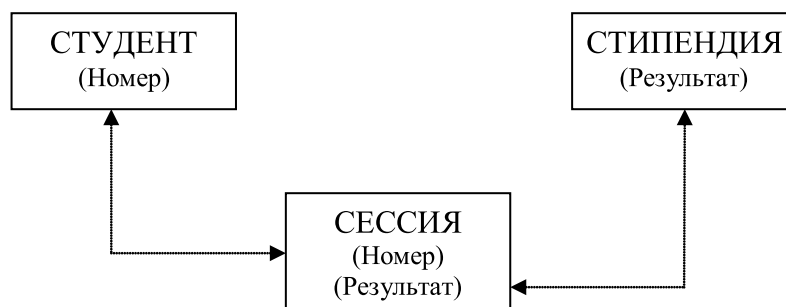


Рис.1.5. Пример реляционной модели

Таблица **СТУДЕНТ** имеет поля: Номер, Фамилия, Имя, Отчество, Дата рождения, Группа;

**СЕССИЯ** – Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат;

**СТИПЕНДИЯ** – Результат, Процент.

Таблицы **СТУДЕНТ** И **СЕССИЯ** имеют совпадающие ключи (**Номер**), что дает возможность легко организовать связь между ними. Таблица **СЕССИЯ** имеет первичный ключ **Номер** и содержит внешний ключ **Результат**, который обеспечивает ее связь с таблицей **СТИПЕНДИЯ**.

Благодаря имеющимся связям достигаются следующие преимущества:

1. Удастся избежать дублирования информации. Все необходимые данные можно хранить только в одной таблице. Так, например, нет необходимости в таблице **СЕССИЯ** хранить номер группы каждого студента, сдающего экзамены, достаточно задать связь с таблицей **СТУДЕНТ**.

2. В реляционных базах данных легко производить изменения. Если в таблице **СЕССИЯ** изменить какие-нибудь значения, то правильная информация автоматически будет связана с другими таблицами, ссылающимися на первую (например, таблица **СТИПЕНДИЯ**).

3. В нереляционных базах данных сложно передать все имеющиеся зависимости, т.е. связать друг с другом данные из различных таблиц. Реляционная база данных выполняет все эти действия автоматически.

4. В реляционных базах данных удастся легко избежать установления ошибочных связей между различными таблицами.

ми данных, а необходимый объем памяти сокращен до минимума.

#### 1.4. СВОЙСТВА СУБД И БАЗЫ ДАННЫХ

К основным свойствам СУБД и базы данных можно отнести:

- отсутствие дублирования данных в различных объектах модели, обеспечивающее однократный ввод данных и простоту их корректировки;
- непротиворечивость данных;
- целостность БД;
- возможность многоаспектного доступа;
- всевозможные выборки данных и их использование различными задачами и приложениями пользователя;
- защиту и восстановление данных при аварийных ситуациях, аппаратных и программных сбоях, ошибках пользователя;
- защиту данных от несанкционированного доступа средствами разграничения доступа для различных пользователей;
- возможность модификации структуры базы данных без повторной загрузки данных;
- обеспечение независимости программ от данных, позволяющее сохранить программы при модификации структуры базы данных;
- реорганизацию размещения данных базы на машинном носителе для улучшения объемно-временных характеристик БД;
- наличие языка запросов высокого уровня, ориентированного на конечного пользователя, который обеспечивает вывод информации из базы данных по любому запросу и предоставление ее в виде соответствующих отчетных форм, удобных для пользователя.

#### 1.5. ОБЗОР СУБД

Рынок программного обеспечения ПК располагает большим числом разнообразных по своим функциональным возможностям коммерческих систем управления базами данных общего назначения, а также средствами их окружения практически для всех массовых моделей машин и для различных операционных систем.

Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и безопасности, что дает возможность разработчикам гарантировать большую безопасность данных при меньших затратах сил на низкоуровневое программирование.

Наиболее распространенная классификация средств СУБД по способу доступа к базе данных. Различают файл-серверные и клиент-серверные системы управления базами данных.

**В файл-серверных СУБД** файлы данных располагаются централизованно на файл-сервере. СУБД располагается на каждом клиентском компьютере (рабочей станции). Доступ СУБД к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера. Недостатки: потенциально высокая загрузка локальной сети; затруднённое централизованное управление; затруднённое обеспечения таких важных характеристик как высокая надёжность, высокая доступность и высокая безопасность. Применяются чаще всего в локальных приложениях, которые используют функции управления БД.

На данный момент файл-серверные СУБД считаются устаревшими. Примерами являются: Microsoft Access, Paradox, dBase, FoxPro, Visual FoxPro.

**Клиент-серверная СУБД** располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме. Все клиентские запросы на обработку данных обрабатываются клиент-серверной СУБД централизованно. Недостаток клиент-серверных СУБД состоит в повышенных требованиях к серверу. Достоинства: потенциально более низкая загрузка локальной сети; удобство централизованного управления; удобство обеспечения таких важных характеристик как высокая надёжность, высокая доступность и высокая безопасность.

Примеры: Oracle, Firebird, Interbase, IBM DB2, MS SQL Server, Sybase, PostgreSQL, MySQL, Caché.

## 1.6. БАЗА ДАННЫХ ACCESS

Microsoft Office Access способен обрабатывать разнообразные данные, сохранять информацию в специальной форме, выбирать из таблиц необходимые сведения, обрабатывать, подготавливать отчеты и т.п.

Microsoft Office Access по своей структуре и способу хранения информации относится к реляционным базам данных, так как вся информация хранится в таблицах.

В Microsoft Office Access под *базой данных* понимается совокупность всех информационных данных и средств обработки (объектов): *таблицы, формы, запросы, отчеты, макросы, модули*. Фактически базой данных в Microsoft Access называется вся информационно-поисковая система. В Microsoft Office Access 2007-2010 перечисленные выше данные хранятся в одном файле, имеющем расширение **accdb**. В некоторых СУБД для хранения каждой таблицы, отчета, модуля создаются отдельные файлы, имена которых имеют различные расширения.

Рассмотрим более подробно терминологию Microsoft Office Access.

*Таблица* — набор данных, имеющих табличную структуру. Каждый объект учета (сотрудник, автомобиль и т.д.) характеризуется набором *реквизитов (полей)*. Данные одного объекта располагаются в одной строке таблицы, называемой *записью*. Столбцы таблицы соответствуют различным реквизитам. Данные одного столбца таблицы имеют один тип и формат.

*Запрос* — совокупность условий отбора данных из существующих таблиц базы данных.

*Форма* — диалоговое окно, предназначенное для ввода и изменения данных таблиц и запросов в удобном для пользователя виде, а также размещения командных кнопок и т.п.

*Отчет* — средство оформления данных таблиц и запросов в наглядном виде для вывода на печать.

*Макрос* — записанная последовательность команд Microsoft Access, предназначенная для автоматического выполнения некоторой последовательности операций пользователя.

*Модуль* — программа, написанная на алгоритмическом языке Visual Basic и предназначенная для автоматического

выполнения ряда операций. Модули составляют обычно в тех случаях, когда разработка макроса невозможна или затруднена. Программирование модулей предоставляет более широкие возможности, но требует также и больших знаний.

*Индекс* — выражение, составленное на основе данных реквизитов таблицы и используемое для упорядочивания записей базы данных. В большинстве случаев в качестве индекса может выступать один из реквизитов. Индексируют обычно поля таблиц, по которым часто производится поиск информации.

*Ключ* — уникальное выражение, составленное на основе данных реквизитов таблицы и используемое для идентификации записей базы данных. В большинстве случаев в качестве ключа может выступать один из реквизитов или индекс. Ключевые поля таблиц используются для установления связи между ними.

*Счетчик* — специальный тип данных, значения которого для каждой записи уникальны. Значения данного поля устанавливаются СУБД автоматически. Обычно счетчики используются для создания ключевых полей таблиц.

*Мастер* — средство, организующее построение таблиц, форм, отчетов и т.д. посредством диалога с пользователем.

*Конструктор* — инструментарий для ручной разработки таблиц, форм, отчетов, запросов.

### **Основные операции по разработке БД.**

*Создание новой базы* — создание нового файла, предназначенного для хранения перечисленных выше объектов.

*Создание таблиц* — описание свойств реквизитов таблицы и сохранение их под определенным именем.

*Создание форм* — размещение в диалоговом окне полей данных таблиц или запросов, пояснительных надписей к ним, а также создание командных кнопок и их связь с макросами или модулями. Каждая форма сохраняется в базе данных под своим именем.

*Создание запросов* — описание полей данных, включаемых в запрос, а также условий отбора записей. Каждому запросу в базе данных присваивается собственное имя.

*Создание фильтров* — описание условий отбора данных. Фильтры могут быть сохранены в виде запросов.

*Создание макросов* — описание последовательности действий с таблицами, формами и запросами базы данных, кото-

рые обычно пользователь выполняет при помощи меню СУБД, командных кнопок и клавиатуры или манипулятора «мышь».

*Создание отчетов* — размещение на макете страницы полей таблиц или запросов, пояснительных надписей к ним. Организация группировки данных, порядка их следования.

*Корректировка данных* — добавление, удаление или обновление данных таблиц при помощи форм или непосредственно в таблицах.

## 2. ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

### 2.1. РАЗРАБОТКА МАКЕТА БАЗЫ ДАННЫХ

Созданию файла базы данных должна предшествовать подготовительная работа. Первое, что необходимо уяснить разработчику, – это для каких целей и задач будет использоваться создаваемая база данных, а также объем и тип данных, размещаемых в ней.

На этапе определения набора данных, предназначенных для хранения в БД, следует определить документальные источники исходной информации: приказы, нормативные документы, письма, распоряжения, карточки учета и т.п. Любые данные, вносимые в БД, должны иметь документальное подтверждение.

Документальные источники первичной информации определяют в значительной степени свойства реквизитов таблиц базы данных. Характер поступления документов, их структура влияют на распределение данных по различным таблицам БД.

В таблицы базы данных может быть добавлена справочная информация, облегчающая работу пользователя с базой.

#### **Постановка задачи.**

*Для определенности будем считать в дальнейшем, что перед нами стоит задача по разработке базы данных для учета данных сотрудников подразделений ОВД «Кадры».*

База данных должна позволять ввод и корректировку информации по штатному расписанию, данных о сотрудниках и составе их семей, получать справочную информацию в зависимости от условий поиска и выдавать на печать или экран различные отчеты.

#### **Структура таблиц.**

Разделение всей информации на отдельные таблицы позволяет уменьшить объем базы данных и оптимизировать поиск информации. Для рассматриваемой задачи разумно выделить сведения о штатном расписании, сотруднике, адресе проживания и данные о родственниках в разных таблицах. Это обусловлено тем, что информация об адресе проживания должна быть доступна не всем пользователям, и количество родственников у сотрудников различно, что не позволяет объединить все данные в одной таблице.

В ряде случаев реквизиты записей могут принимать только определенные значения из заданного списка, например звание. Для таких реквизитов обычно создаются специальные таблицы-словари. Таблица-словарь имеет два столбца: код (ключевое поле) и наименование. Современные версии Microsoft Office Access позволяют создавать словари, непосредственно связанные с полями таблиц (поле со списком). Для связи данных со словарем в основной таблице вводят числовые поля, содержащие код словарного реквизита. Это позволяет также уменьшить объем основной таблицы.

Основная таблица связывается с дополнительными таблицами обычно при помощи числовых полей.

После определения количества таблиц и их реквизитов необходимо выбрать типы данных для каждого реквизита. Microsoft Office Access допускает следующие типы данных:

**Счетчик.** Хранит уникальное число, генерируемое программой автоматически. При добавлении в таблицу новой записи значение в этом поле увеличивается на единицу. При удалении записей счетчик продолжает нумерацию так, что числа не повторяются.

**Текстовый.** Текстовые поля (числа, буквы, знаки пунктуации и символы не более 255) для всех записей имеют одну длину, задаваемую пользователем. По умолчанию создается поле длиной 255 символов.

**Мемо.** Это поле примечаний. В данном поле может храниться большой объем неформатированного текста до 65 536 символов. Размер данного поля может быть различным у разных записей. Но операции поиска и сортировки по данному полю работают более медленно.

**Числовой.** Числовые поля могут иметь размер: *байт* — числа от 0 до 255; *целое* — числа от -32 768 до +32 767; *длинное целое* — числа от -2 147 483 648 до +2 147 483 647; *одинарное с плавающей точкой* — длина 4 байта; *двойное с плавающей точкой* — длина 8 байт; *действительное* — длина 8 байт.

**Дата/время.** В этих полях хранятся данные о дате и времени. Такие поля имеют несколько форматов отображения даты и времени.

**Денежный.** Денежная сумма с точностью до 15 целых и 4 десятичных разрядов.

**Логический.** Хранится логическое значение «ИСТИНА – ДА» или «ЛОЖЬ – НЕТ».

**Объект OLE.** В этих полях хранятся объекты, разработанные другими программными приложениями, например рисунки и документы. Этот тип унаследован от старых версий Microsoft Access.

**Гиперссылка.** Можно хранить ссылки на другие объекты (файлы, интернет-страницы, объекты базы данных и электронную почту).

**Вложение.** Позволяет сохранять в поле базы данных один или несколько отдельных файлов. Наиболее предпочтительный вариант для внедрения рисунков, документов и т.п.

**Мастер подстановки.** Позволяет организовать в поле подстановку значений из других объектов базы данных (таблиц или запросов) либо задать перечень возможных значений.

Каждое поле таблицы имеет ряд свойств, которые пользователь может также при необходимости изменить.

Ниже приводятся описания структуры таблиц базы данных **Кадры.**

*Таблица 2.1*

#### СТРУКТУРА ТАБЛИЦЫ ШТАТНОЕ РАСПИСАНИЕ

<b>Имя поля</b>
Код
Подразделение
Должность
Кол-во единиц
Оклад

*Таблица 2.2*

#### СТРУКТУРА ТАБЛИЦЫ КАРТОЧКА СОТРУДНИКА

<b>Имя поля</b>	<b>Тип данных</b>	<b>Свойства поля</b>
Код	Счетчик	
Личный номер	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 10</b>
Фамилия	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 20</b>

<b>Имя поля</b>	<b>Тип данных</b>	<b>Свойства поля</b>
Имя	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 15</b>
Отчество	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 15</b>
Звание	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 30</b>
		вкладка <b>Подстановка</b>
		<b>Тип элемента управления – Поле со списком</b> <b>Тип источника строк – Список значений</b> <b>Источник строк – рядовой; сержант; лейтенант; капитан; майор</b>
Оклад по званию	Числовой	вкладка <b>Общие</b>
		<b>Размер поля – Целое</b>
Должность	Мастер подстановки	вкладка <b>Подстановка</b>
		<b>Ограничиться списком – Да</b>
Дата рождения	Дата/время	вкладка <b>Общие</b>
		<b>Формат поля – Краткий формат даты</b>
		<b>Маска ввода – Краткий формат даты</b>
Дата приема на работу	Дата/время	вкладка <b>Общие</b>
		<b>Формат поля – Краткий формат даты</b>
		<b>Маска ввода – Краткий формат даты</b>

Таблица 2.3

### СТРУКТУРА ТАБЛИЦЫ АДРЕСА СОТРУДНИКОВ

<b>Имя поля</b>	<b>Тип данных</b>	<b>Свойства поля</b>
Код	Числовой	вкладка <b>Общие</b>
		<b>Размер поля – Длинное целое</b> <b>Индексированное поле – Да</b> <b>(Совпадения не допускаются)</b>

<b>Имя поля</b>	<b>Тип данных</b>	<b>Свойства поля</b>
Город	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 50</b>
Улица	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 30</b>
Дом	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 5</b>
Квартира	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 5</b>

Таблица 2.4

### СТРУКТУРА ТАБЛИЦЫ РОДСТВЕННИКИ СОТРУДНИКА

<b>Имя поля</b>	<b>Тип данных</b>	<b>Свойства поля</b>
Код	Числовой	вкладка <b>Общие</b>
		<b>Размер поля – Длинное целое</b> <b>Индексированное поле – Да</b> (Допускаются совпадения)
Степень родства	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 20</b>
		вкладка <b>Подстановка</b>
		<b>Тип элемента управления – Поле со списком</b> <b>Тип источника строк – Список значений</b> <b>Источник строк – мать;отец;сын;дочь;сестра;брат</b>
Фамилия	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 20</b>
Имя	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 15</b>
Отчество	Текстовый	вкладка <b>Общие</b>
		<b>Размер поля – 15</b>
Дата рождения	Дата/ время	вкладка <b>Общие</b>
		<b>Формат поля – Краткий формат даты</b> <b>Маска ввода – Краткий формат даты</b>

## 2.2. СОЗДАНИЕ НОВОЙ БАЗЫ ДАННЫХ

Новая база данных создается после запуска Microsoft Access. Для запуска приложения необходимо найти пиктограмму Microsoft Office Access и дважды щелкнуть левой кнопкой мыши при размещенном на ней указателе. В отличие от текстовых редакторов и электронных таблиц Microsoft Office Access не создает автоматически бланк пустой базы данных. Пользователю предлагается сделать выбор между открытием существующей базы данных или созданием новой. В Access также предусмотрены разнообразные шаблоны, с помощью которых можно быстро создать базу данных. Шаблон – это уже готовая к использованию база данных, включающая все необходимые таблицы, запросы, формы и отчеты для выполнения определенной задачи. Шаблоны баз данных можно использовать без изменений или настроить в соответствии с конкретными потребностями.

Для создания новой базы данных следует выполнить следующие действия:

1. Запустите Microsoft Office Access.
2. На стартовой панели (рис 2.1) выберите опцию **Новая база данных**.

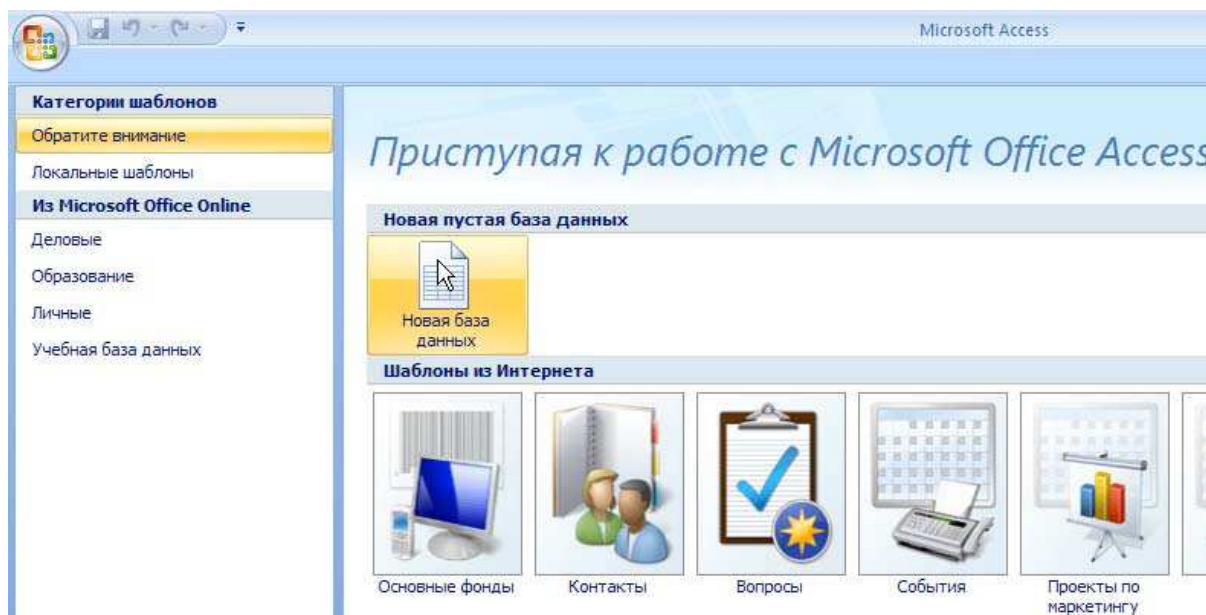



Рис.2.1. Стартовая панель Microsoft Office Access

3. С помощью кнопки **Обзор**  укажите папку вашей группы для размещения новой базы данных. В поле **Имя файла** введите имя **Кадры**. Подтвердите создание файла кнопкой **Создать**.

Откроется окно базы данных, изображенное на рисунке 2.2. Оно является основным окном приложения и позволяет открывать, добавлять и удалять любые объекты базы данных. В левой части окна, называемой областью переходов, находятся списки групп объектов. По умолчанию в новой базе данных нет всех объектов, поэтому на экране отображается пустой шаблон таблицы с именем **Таблица 1**. Для дальнейшего удобства работы с объектами базы данных в заголовке области переходов нажмите на кнопку со списком **Все таблицы** и выберите **Тип объекта** для группировки объектов по типам.

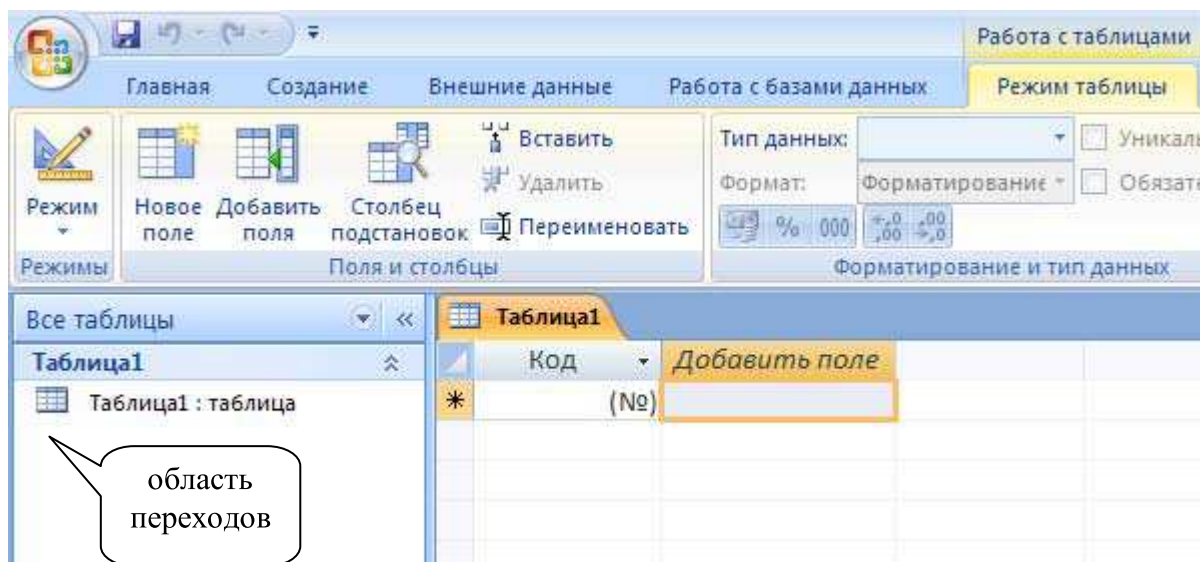


Рис.2.2. Окно базы данных

### 2.3. СОЗДАНИЕ ТАБЛИЦЫ ПУТЕМ ВВОДА ДАННЫХ

Новая база данных не содержит никаких объектов, поэтому по умолчанию создается пустая таблица в режиме ввода данных. Режим **Таблица** позволяет в доступной форме добавлять новые поля и вводить данные. Если в базе данных не была создана пустая таблица, то следует в ленте меню на вкладке **Создание** из группы **Таблицы** выбрать команду **Таблица**, после чего появится окно **Таблица1** (рис. 2.2).

В открывшейся таблице переименуйте названия столбцов (полей) присвоив имена из таблицы **Штатное расписание** (табл.2.1).

Переименовать столбцы можно несколькими способами:

– выделите столбец и воспользуйтесь вкладкой ленты **Режим таблицы** ⇒ группа **Поля и столбцы** ⇒ **Переименовать**;


– щелкните правой кнопкой мыши по заголовку столбца и воспользуйтесь контекстным меню;

– дважды щелкните по заголовку столбца и введите новое имя.

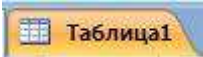

Не изменяйте поле **Код**, созданное автоматически в качестве индексного поля.

После создания полей заполните таблицу записями данных о штатном расписании подразделений, придумав 7-10 должностей самостоятельно. Например

Код	Подразделение	Должность	Кол-во единиц	Оклад
5	Следствие	следователь	5	4000
6	Уголовный розыск	оперуполномоченный	6	3900
7	Дознание	дознатель	3	3800
8	Дежурная часть	водитель	2	3500
9	Отдел кадров	инспектор	2	3600
10	Уголовный розыск	старший оперуполномоченный	2	4200
11	Следствие	старший следователь	2	4500
12	Дежурная часть	оперативный дежурный	3	3500
*	(№)			

Помните, что даже мелкие опечатки могут привести к нарушению целостности данных, а иногда к серьезным ошибкам в подготовленных документах. Поэтому *после завершения ввода данных в таблицу рекомендуется проверить орфографию*. Для проверки орфографии выделите мышью всю таблицу кнопкой в верхнем левом углу таблицы  и выполните команду на вкладке **Главная** ⇒ группа **Записи** ⇒ **Орфография**.

Сохраните созданную таблицу, присвоив ей имя **Штатное расписание**. Для этого воспользуйтесь пунктом **Сохранить** контекстного меню, нажав правую кнопку мыши на имени

, или закройте таблицу кнопкой **Закрытие** . Microsoft Access автоматически предложит указать новое имя таблицы для сохранения.

## 2.4. СОЗДАНИЕ ТАБЛИЦЫ В РЕЖИМЕ КОНСТРУКТОРА

Конструктор таблиц предназначен для задания и изменения структуры таблиц. С помощью конструктора можно формировать сколь угодно сложные таблицы с полями любого типа.

При построении таблицы в этом режиме Access не создает по умолчанию никаких полей. Вы сами должны определить структуру и типы данных полей будущей таблицы.

На вкладке **Создание** в группе **Таблицы** щелкните по кнопке **Конструктор таблиц**. В верхней части окна **Конструктора** (рис.2.3.) расположена таблица (точнее – бланк для ее создания), которая должна содержать перечень всех полей с указанием их типов и заголовков.

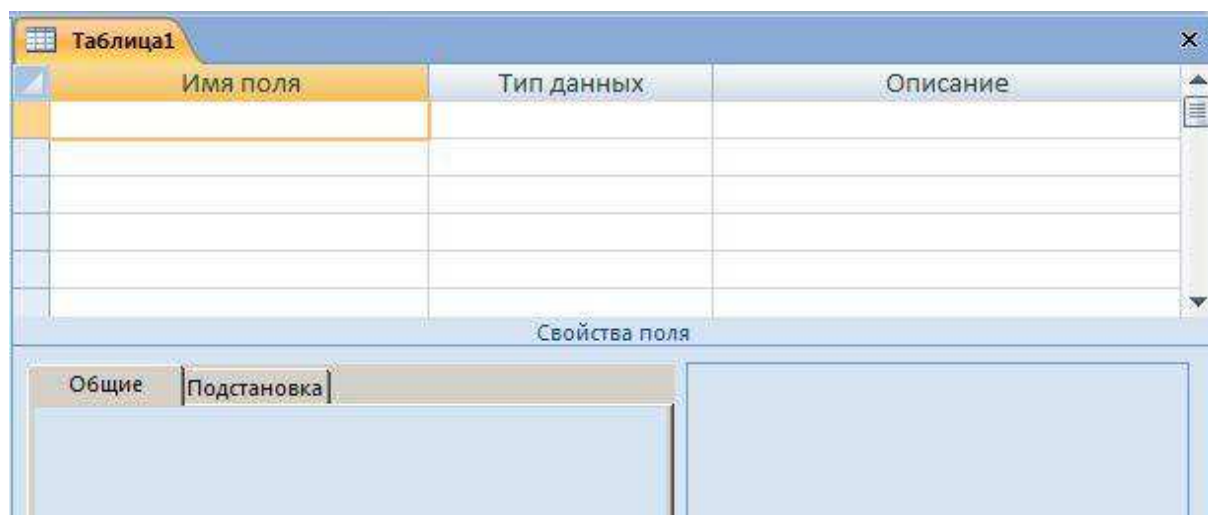


Рис.2.3. Конструктор таблиц

Колонка **Имя поля** содержит список создаваемых полей таблицы. Наименование каждого из полей таблицы, как правило, выбирается произвольно, но старайтесь делать это так, чтобы оно в максимальной степени характеризовало информацию, которая будет храниться в данном поле. Наименование поля может содержать до 64 символов (буквы, цифры, пробелы и специальные символы, за исключением точки, восклицательного знака и квадратных скобок). Наименование поля не должно начинаться с пробела, а само имя поля не должно повторяться. Чтобы придерживаться единого стиля рекомендуется создавать имена полей, начиная с заглавной буквы.

Колонка **Тип данных** позволяет задать соответствующий тип данных для текущего в строке поля. После того, как Вы выберете **Тип данных**, в нижней секции окна **Свойства поля**

выведется краткая информация о полях и их свойствах, которая меняется при перемещении указателя мыши от поля к полю в окне конструктора.

В столбце **Описание** Вы можете внести любой пояснительный текст для каждого из полей. В дальнейшем при выборе поля в режиме таблицы или запроса в строке состояния будет появляться текст этого описания. Поле **Описание** – не является обязательным и может не заполняться.

Теперь с помощью конструктора добавим в базу данных таблицы, предусмотренные в макете базы **Кадры**, а именно таблицы **Карточка сотрудника**, **Адреса сотрудников**, **Родственники сотрудника**.

Для таблицы **Карточка сотрудника** (табл. 2.2):

1. В первой строке колонки **Имя поля** введите слово **Код**, которое будет играть роль названия нового поля таблицы.

2. Щелкните на соответствующей ячейке **Тип данных** строки поля **Код**. Здесь необходимо выбрать тип поля, который определяет структуру и размер данных.

3. Щелкните на появившейся стрелке раскрывающегося списка и выберите пункт **Счетчик**.

4. Нажмите клавишу **Tab**. В столбце **Описание** для текущего поля базы можно указать дополнительную информацию, которая является *необязательной*.

5. Щелкните на второй ячейке столбца **Имя поля** и введите имя **Личный номер**.

6. Задайте для этого поля тип данных **Текстовый** и укажите в нижней части окна конструктора **Свойства поля** в свойстве **Размер поля** значение 8 (символов) согласно рисунку 2.4.

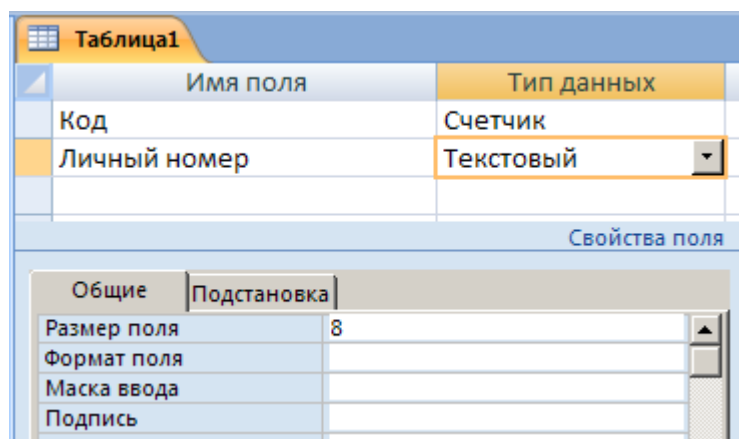


Рис. 2.4. Окно свойств поля **Личный номер**

7. Добавьте поля **Фамилия**, **Имя** и **Отчество**. Укажите тип полей **Текстовый**. В свойстве **Размер поля** поставьте соответственно значения 20, 15, 15.

8. Добавьте поле **Звание**. Укажите тип поля **Текстовый** и **Размер поля** – 30. На вкладке **Подстановка** для свойства поля **Тип элемента управления** выберите **Поле со списком**. Для свойства **Тип источника строк** выберите **Список значений**. В свойстве **Источник строк** укажите через точку с запятой требуемые значения подстановки для этого поля (рис.2.5).

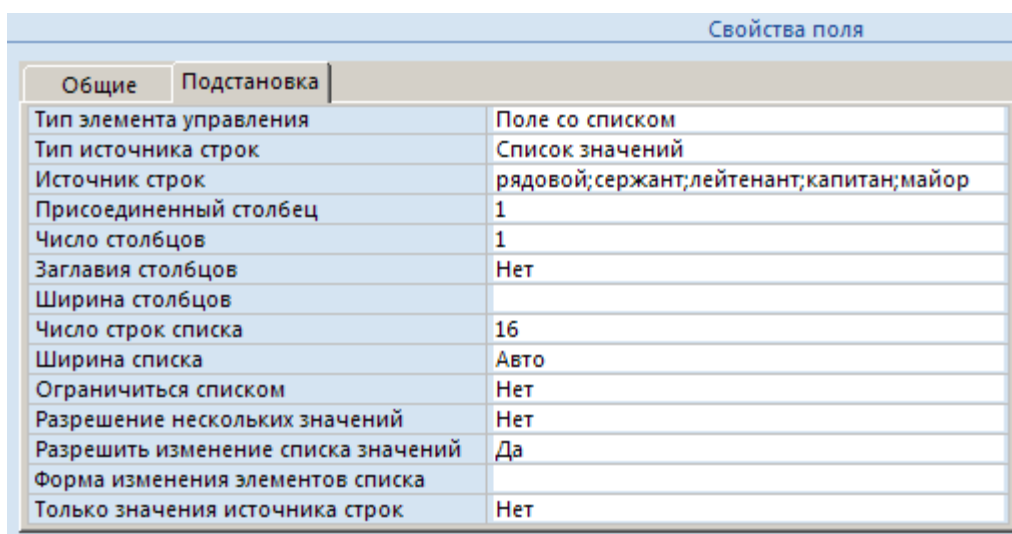



Рис.2.5. Окно свойств поля **Звание**

9. Добавьте поле **Оклад по званию**. Укажите тип поля **Числовой** и свойство **Размер поля** – **Целое** (для простоты будем считать, что размер оклада будет принимать только целочисленные значения).


10. Добавьте поле **Должность**. Укажите тип поля **Мастер подстановки**. В открывшемся окне выберите опцию **Объект "столбец подстановки"** будет использовать значения из **таблицы** или **запроса** и нажмите кнопку **Далее**. В следующем окне выберите параметр **Таблицы**, источник данных – **таблица Штатное расписание** и нажмите кнопку **Далее**. Выберите поле **Должность** и щелкните по кнопке . Нажмите **Далее** и в следующем окне задайте сортировку для поля **Должность** по возрастанию. В следующем окне настройте ширину столбца подстановки для того чтобы он полностью выводил его содержимое. Опцию **Скрыть ключевой столбец (рекомендуется)** оставьте без изменения включенной. Нажмите кнопку **Готово**. При этом Microsoft Access потребует сохранения таблицы. В

ответ щелкните на кнопке **Да** и в появившемся окне введите имя таблицы – **Карточка сотрудника**. Далее необходимо для таблицы **Карточка сотрудника** назначить ключевое поле, в ответ на сообщение Microsoft Access **Ключевые поля не заданы** нажмите кнопку **Да** для их создания.

11. На вкладке **Подстановка** в свойстве **Ограничиться списком** выберите **Да** (рис. 2.5).

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	SELECT [Штатное расписание].[Код], [Штатное рас
Присоединенный столбец	1
Число столбцов	2
Заглавия столбцов	Нет
Ширина столбцов	0 см; 3,254 см
Число строк списка	16
Ширина списка	3,254 см
Ограничиться списком	Да
Разрешение нескольких значений	Нет
Разрешить изменение списка значений	Да
Форма изменения элементов списка	
Только значения источника строк	Нет

Рис.2.5. Окно свойств поля **Должность**

12. Для поля **Дата рождения** укажите тип поля **Дата/время**. В разделе **Свойства поля** для свойства **Формат поля** выберите из списка **Краткий формат даты**. В свойстве **Маска ввода** с помощью мастера (кнопка в конце строки ) установите **Краткий формат даты**, затем подтвердите **Готово**. После выбора мастера Microsoft Access попросит сохранить таблицу – ответьте **Да**.

13. Для поля **Дата приема на работу** укажите тип поля **Дата/время**. В свойстве **Формат поля** выберите **Краткий формат даты**. В свойстве **Маска ввода**, аналогично с помощью мастера, установите **Краткий формат даты**.

14. После создания всех полей таблицы **Карточка сотрудника** (рис. 2.6) щелкните на кнопке закрытия окна конструктора, подтвердив предложение сохранить изменения макета таблицы.

Штатное расписание		Карточка сотрудника	
Имя поля		Тип данных	
Код	Счетчик		
Личный номер	Текстовый		
Фамилия	Текстовый		
Имя	Текстовый		
Отчество	Текстовый		
Звание	Текстовый		
Оклад по званию	Числовой		
Должность	Числовой		
Дата рождения	Дата/время		
Дата приема на работу	Дата/время		

Рис.2.6. Список полей таблицы **Карточка сотрудника**

## 2.5. СОЗДАНИЕ ТАБЛИЦЫ АДРЕСА СОТРУДНИКОВ

Создайте таблицу **Адреса сотрудников** согласно макету (табл. 2.3). Для организации в дальнейшем связи **один-к-одному** не забудьте правильно задать для поля **Код** в свойстве **Индексированное поле** значение **Да (Совпадения не допускаются)**. Учтите, что поле **Код** в этой таблице служит для связи с главной таблицей **Карточка сотрудника** и ключевым быть не может, поэтому при закрытии или сохранении на запрос создания ключевого поля отвечайте – **Нет**.

## 2.6. СОЗДАНИЕ ТАБЛИЦЫ РОДСТВЕННИКИ СОТРУДНИКА

Создайте таблицу **Родственники сотрудника** согласно макету (табл. 2.4). Для организации связи **один-ко-многим** не забудьте правильно указать для поля **Код** в свойстве **Индексированное поле** значение **Да (Допускаются совпадения)**. Учтите, что поле **Код** в этой таблице служит для связи с главной таблицей **Карточка сотрудника** и ключевым быть не может, поэтому на запрос создания ключевого поля отвечайте – **Нет**.

## 2.7. СХЕМА ДАННЫХ

После создания таблиц базы данных необходимо установить связи между ними. Существует несколько типов возможных связей между таблицами. Наиболее распространенными

являются связи *один-к-одному* и *один-ко-многим*. Связь между двумя таблицами организуется на основе общего поля. Оно должно присутствовать как в той, так и в другой таблице, причем в одной из таблиц оно обязательно должно быть ключевым. То есть на стороне *один* всегда должно выступать ключевое поле, содержащее уникальные, неповторяющиеся значения. Значения того же поля на стороне *многие* могут повторяться. Итак, связь *один-к-одному* означает, что одной записи в таблице соответствует только одна запись в другой связанной таблице. Связь типа *один-ко-многим* означает, что одной записи в таблице могут соответствовать несколько записей в связанной таблице.

Перед работой со схемой данных необходимо убедиться, что все таблицы закрыты и не используются.

Для определения связей между таблицами в Microsoft Access следует на вкладке **Работа с базами данных** в группе **Показать или скрыть** нажать кнопку **Схема данных**.

Конструктор схемы данных представляет собой окно, в котором пользователь может разместить списки полей существующих в базе данных таблиц (рис.2.7). Для помещения в окно списка полей таблиц следует щелкнуть правой кнопкой мыши внутри окна схемы данных и в контекстном меню выполнить пункт **Добавить таблицу**.

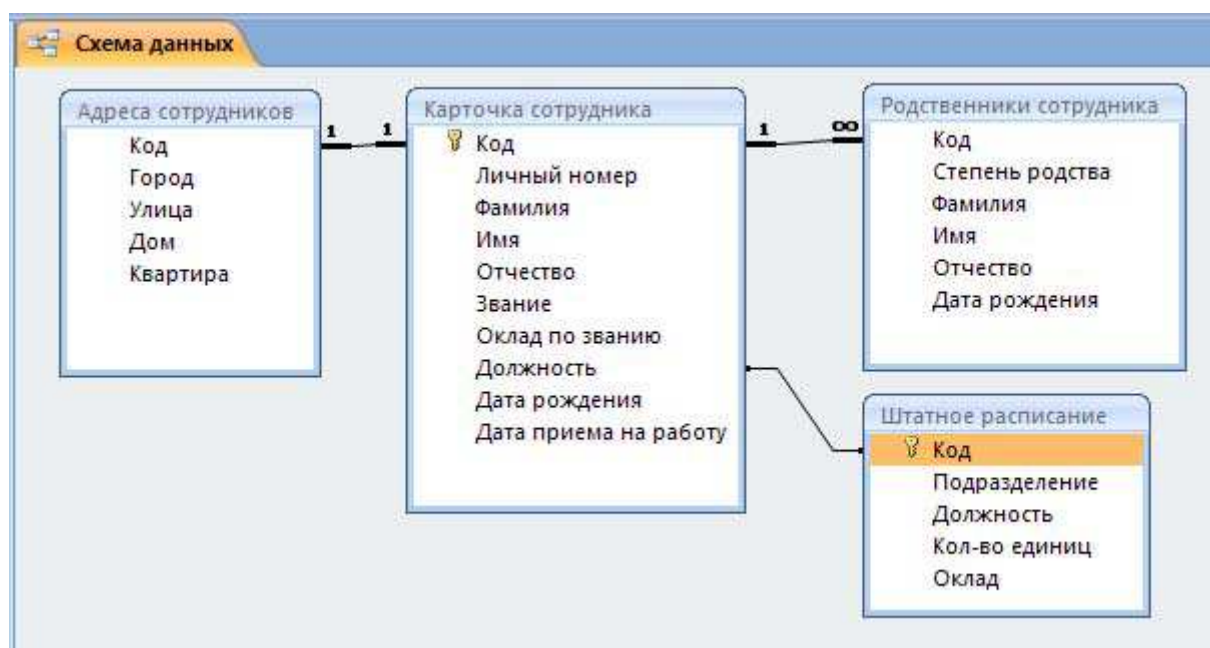


Рис.2.7. Схема данных

Добавьте в **Схему данных** все имеющиеся в базе данных **Кадры** таблицы, исключая их дублирование. Обратите внимание, что между таблицами **Карточка сотрудника** и **Штатное расписание** уже существует связь по полю **Должность** и **Код**. Она была создана автоматически при использовании **Мастера подстановок**. Теперь создадим связи между остальными таблицами:

1. Поместите указатель мыши на поле **Код** таблицы **Карточка сотрудника**.

2. Нажмите кнопку мыши и перетащите указатель на поле **Код** таблицы **Адреса сотрудников**.

3. В открывшемся окне диалога установите флажки **Обеспечение целостности данных**, **каскадное обновление связанных полей** и **каскадное удаление связанных записей** (рис.2.8). Обеспечение целостности данных устанавливает набор правил позволяющих Access поддерживать допустимые межтабличные связи, изменения в связанных полях, контролировать удаление записей в связанных таблицах. Установка дополнительных флажков **каскадное обновление связанных полей** и **каскадное удаление связанных записей** позволяет наиболее полно обеспечивать целостность информации в базе данных и соблюдать основные свойства БД (п. 1.4). Так как поле **Код** в таблице **Адреса сотрудников** содержит уникальные значения (свойство **Индексированное поле – Да (Совпадения не допускаются)**), автоматически создается связь *один-к-одному*. Щелкните на кнопке **Создать**.

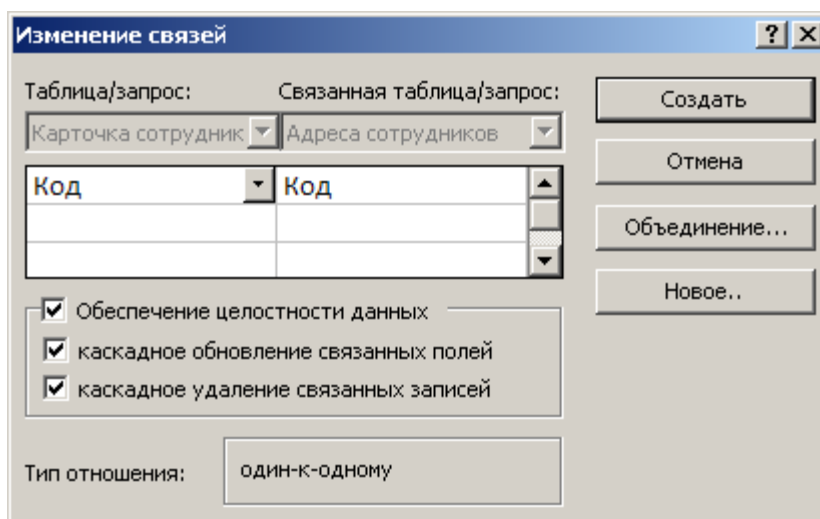


Рис.2.8. Параметры связи

4. Далее поместите указатель мыши на поле **Код** таблицы **Карточка сотрудника**.

5. Нажмите кнопку мыши и перетащите указатель на поле **Код** таблицы **Родственники сотрудника**.

6. В открывшемся окне диалога установите флажки **Обеспечение целостности данных, каскадное обновление связанных полей и каскадное удаление связанных записей**. Так как значения поля **Код** в таблице **Родственники сотрудника** могут повторяться (свойство **Индексированное поле – Да (Допускаются совпадения)**), автоматически создается связь *один-ко-многим*. Щелкните на кнопке **Создать**.

7. Закройте окно **Схема данных**, подтвердив ее сохранение.

### 3. ФОРМЫ

В то время как таблицы и запросы позволяют отобразить на экране длинные списки записей, формы дают возможность сосредоточиться на конкретной записи. Они облегчают ввод, редактирование и восприятие информации, могут содержать вспомогательные подписи и элементы оформления.

#### 3.1. МАСТЕР ФОРМ

Мастер форм позволяет сберечь время и быстро сконструировать привлекательную форму для записей любой таблицы.

1. В ленте перейдите на вкладку **Создание**.
2. В группе **Формы** разверните список **Другие формы** и выберите **Мастер форм**.
3. В раскрывающемся списке **Таблицы и запросы** первого окна мастера выберите таблицу **Карточка сотрудника** (рис.3.1).

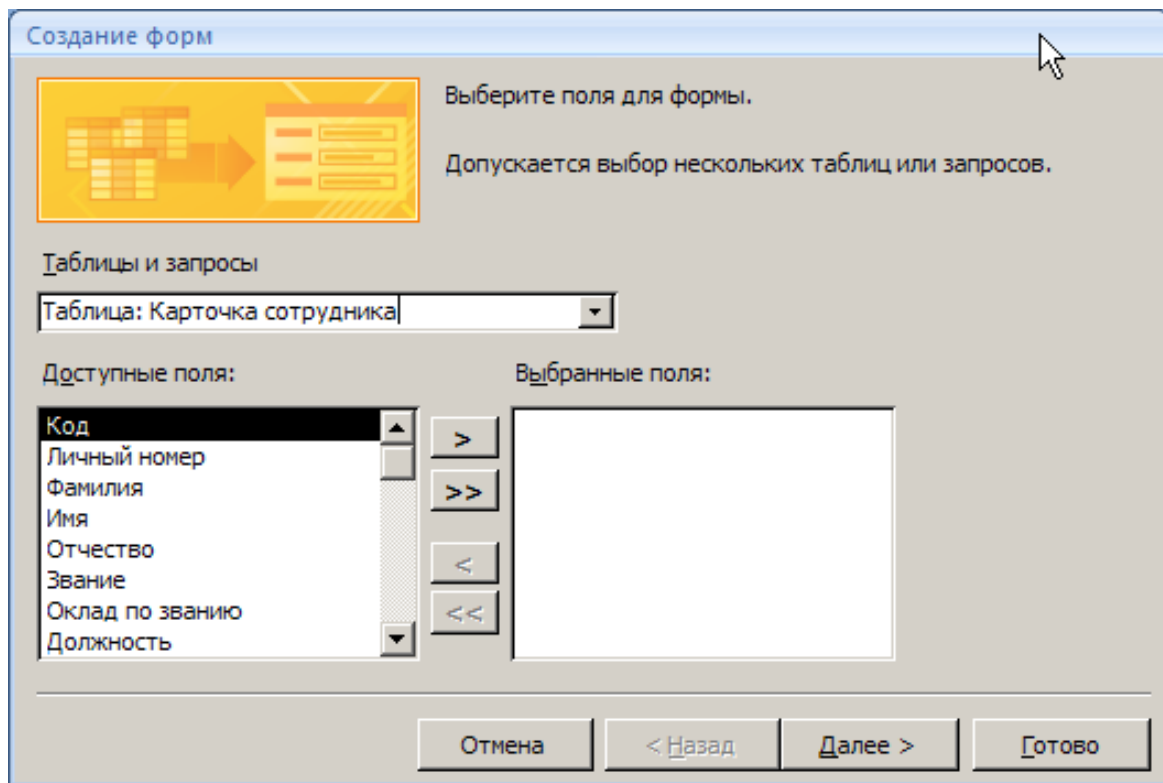


Рис.3.1. Мастер форм

4. Щелкните на кнопке **>>**, чтобы добавить в список **Выбранные поля** все поля таблицы.

5. Затем в раскрывающемся списке **Таблицы и запросы** выберите таблицу **Адреса сотрудников**.

6. Щелкните на кнопке >>, чтобы добавить к списку **Выбранные поля** все поля таблицы **Адреса сотрудников**.

7. В списке **Выбранные поля** выделите пункт **Адреса сотрудников.Код**. Щелкните на кнопке <, чтобы переместить выделенное поле в список **Доступные поля**, так как данное поле уже добавлено в форму из таблицы **Карточка сотрудника**.

8. Щелкните на кнопке **Далее**.

9. В следующем окне диалога выберите для формы стиль **в один столбец** и щелкните на кнопке **Далее**.

10. В списке следующего окна выберите стиль оформления **Стандартная** или любой другой понравившийся вам и снова щелкните на кнопке **Далее**.

11. В последнем окне мастера форм введите имя формы **Карточка сотрудника** и щелкните на кнопке **Готово**. Мастер сгенерирует форму и откроет ее в режиме просмотра (рис.3.2).

Код	(№)
Личный номер	
Фамилия	
Имя	
Отчество	
Звание	
Оклад по званию	
Должность	
Дата рождения	
Дата приема на работу	
Город	
Улица	
Дом	
Квартира	

Запись: 1 | 5 из 5 | Нет фильтра | Поиск

Рис.3.2. Форма **Карточка сотрудника**

Окно формы содержит наименования и области отображения для всех полей таблиц, выбранных в первом окне мастера. В нижней части формы расположены кнопки перемещения по записям, аналогичные кнопкам окна таблицы.

### 3.2. КОНСТРУКТОР ФОРМ

Недостатком форм, создаваемых мастером, является то, что они однообразны и не содержат пояснительных записей. Чтобы приукрасить форму и расположить поля более удобным образом, следует воспользоваться конструктором форм, который позволяет передвигать и масштабировать элементы формы, связывать их с источником данных и настраивать любые другие параметры формы.

1. В ленте перейдите на вкладку **Главная** и в группе **Режимы** с помощью кнопки **Режим** из списка выберите **Конструктор**. Форма **Карточка сотрудника** перейдет в режим конструктора (рис.3.3). Либо, если вы только что загрузили базу данных, перейдите на области переходов в раздел **Формы**, выберите форму **Карточка сотрудника**, нажмите правую кнопку мыши и в контекстном меню укажите **Конструктор**.

Заголовок формы	
Карточка сотрудника	

Область данных	
Код	Код
Личный номер	Личный номер
Фамилия	Фамилия
Имя	Имя
Отчество	Отчество
Звание	Звание
Оклад по званию	Оклад по званию
Должность	Должность
Дата рождения	Дата рождения
Дата приема на работу	Дата приема на работу

Рис.3.3. Режим конструктора формы **Карточка сотрудника**

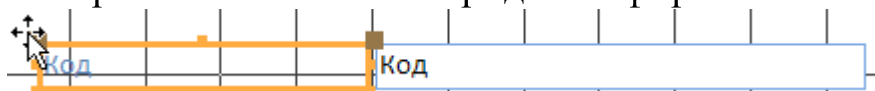
По умолчанию все элементы формы сгруппированы с помощью макета, формат которого мы указали на одном из шагов мастера. Для удаления макета следует левой кнопкой мыши выделить все поля таблицы (рис. 3.4), затем на вкладке **Упорядочить** группы **Макет элемента управления** выбрать команду **Удалить**.

Код	Код
Личный номер	Личный номер
Фамилия	Фамилия
Имя	Имя
Отчество	Отчество
Звание	Звание
Оклад по званию	Оклад по званию
Должность	Должность
Дата рождения	Дата рождения
Дата приема на работу	Дата приема на работу
Город	Город
Улица	Улица
Дом	Дом
Квартира	Квартира

Рис.3.4. Выбор всех элементов формы **Карточка сотрудника**

После этого мы имеем возможность управлять размещением каждого элемента поля отдельно. Обычно слева находится надпись поля, а справа непосредственно поле. Вводить данные в форму в режиме конструктора категорически запрещено.

Любой объект формы можно выделить щелчком мыши, при этом вокруг него появляется габаритная рамка с боковыми и угловыми маркерами масштабирования объекта. В левом верхнем углу объекта появляется большой квадрат, за который можно перетаскивать его в пределах формы.



С помощью маркеров масштабирования объектов можно изменять их размеры. Чтобы одновременно переместить поле

вместе с его подписью, тяните объект за тело подписи, но не за маркер перемещения объекта.

2. Разместите поля с надписями и измените их размеры, как это показано на рисунке 3.5, предварительно расширив область данных формы.

Рис.3.5. Форма Карточка сотрудника

3. Щелкните правой кнопкой мыши на поле **Код** и выберите в контекстном меню команду **Свойства**.

Любой элемент формы имеет контекстное меню, в котором есть команда **Свойства**. Выбор такой команды открывает **Окно свойств**. Вкладка **Макет** этого окна содержит параметры форматирования объекта. Вкладка **Данные** – описание источника данных, различные ограничения и условия, накладываемые на данные поля. Вкладка **События** позволяет описать функции отклика на различные события (например, перемещение мыши на объект). Параметры, не вошедшие в первые три группы, размещаются на вкладке **Другие**. Раздел **Все** обеспечивает одновременный доступ ко всем параметрам.

4. На вкладке **Данные** окна диалога свойств поля **Код** выберите из раскрывающегося списка поля **Доступ** пункт **Нет** (рис.3.6).

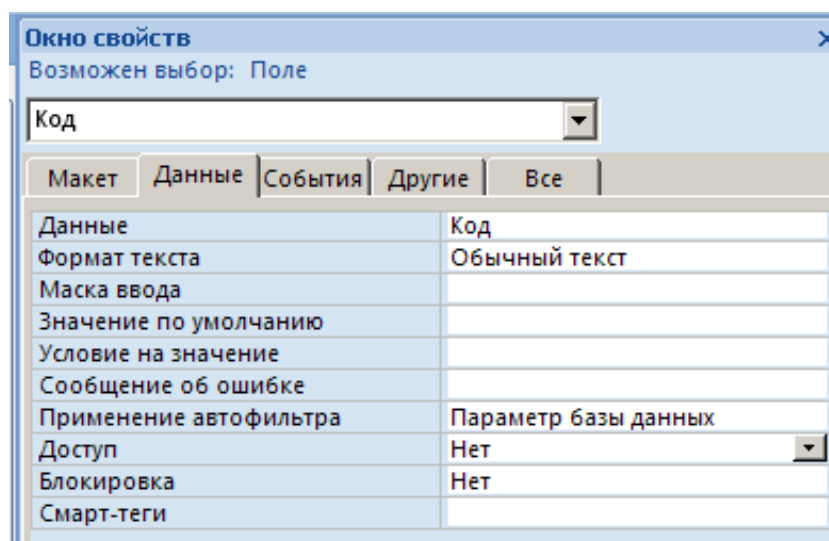


Рис.3.6. Окно свойств поля **Код**

Тем самым мы запретили доступ для изменения поля **Код**, так как значение данного поля генерируется независимо от пользователя.

На вкладке **Конструктор** в группе **Элементы управления** расположены ряд кнопок расширяющих возможности конструктора форм. С помощью кнопок элементов управления в форму можно добавлять различные объекты, типы которых перечислены в таблице 3.1. Элементы управления форм и отчетов сходны между собой, поэтому эти элементы доступны и в конструкторе отчетов, который мы рассмотрим позже.

5. Добавим в форму вычисляемое поле **Выслуга**, в котором будет отображаться количество проработанных лет.

Щелкните на кнопке **Поле**  группы **Элементы управления**.

6. Растяните мышью прямоугольник в разделе **Область данных**, справа от поля **Дата приема на работу**.

7. Введите в область надписи поля имя **Выслуга**.

8. Щелкните правой кнопкой мыши по новому полю, выберите **Свойства**.



















9. На вкладке **Данные** окна диалога свойств нового поля, в строке **Данные**, воспользуйтесь построителем выражений, нажав на кнопку  в конце строки.

Таблица 3.1

## ЭЛЕМЕНТЫ УПРАВЛЕНИЯ ФОРМ И ОТЧЕТОВ

Кнопка	Название	Назначение
	Надпись	Надписи создаются автоматически вместе с элементами типа текстовое поле, поле со списком и т.п. Они разъясняют смысл поля ввода. Дополнительные надписи могут использоваться для заголовков и пояснений
	Поле	Используется для ввода и отображения информации полей таблиц и запросов, а также для вывода результатов вычислений
	Группа переключателей	Группа переключателей ссылается на некоторое поле таблицы. Каждый переключатель группы соответствует определенному целочисленному значению этого поля
	Выключатель	Позволяет вводить информацию типа да/нет. Величине «да» или «истина» соответствует утопленное положение выключателя
	Переключатель	Элемент группы переключателей, в которой может быть отмечен только один из них (его кружок выглядит зачерненным)
	Флажок	Предназначен для представления полей типа да/нет. Отмеченный квадратик соответствует величине «да», а пустой – «нет»
	Поле со списком	Разрешает как непосредственный ввод значения в поле, так и его выбор в раскрывающемся списке определенных значений
	Список	Позволяет выбирать данные из набора пунктов и не разрешает непосредственный ввод значений

Кнопка	Название	Назначение
 Кнопка	Кнопка	Щелчок на такой кнопке выполняет команду, с помощью которой можно перейти в другую форму, переместиться по записям и выполнить многие другие операции
	Рисунок	Рисунок любого графического формата с рамкой
	Свободная рамка объекта	Объект OLE, не связанный ни с каким источником данных
	Присоединенная рамка объекта	Объект OLE, связанный с полем данных таблицы или запроса
	Разрыв страницы	Линия, по которой формируется перевод страницы при выводе формы или отчета на принтер
	Набор вкладок	Позволяет разместить на одном и том же пространстве экрана несколько наборов элементов управления. Удобен в тех случаях, когда элементы легко разделяются на логические группы
	Подчиненная форма/отчет	Вставляет в форму (или отчет) область с другой формой (или отчетом), связанной с главной
	Линия	Добавляет прямую линию
	Прямоугольник	Рисует прозрачный или непрозрачный прямоугольник с цветной границей любой толщины

Выслугу будем считать как разницу между текущей датой и датой приема на работу. Поскольку операция вычитания дат возвращает разницу в днях между двумя датами, а выслугу принято считать в годах, то, поделив результат вычитания на 365, получим количество лет. Поскольку нас интересует полное количество лет результат необходимо округлить. Функция **Date()** возвращает текущую дату, а функция **Int()** позволяет округлять до целого числа.

10. В построителе выражений наберите следующую формулу (рис. 3.7):

**=Int((Date()-[Дата приема на работу])/365)**

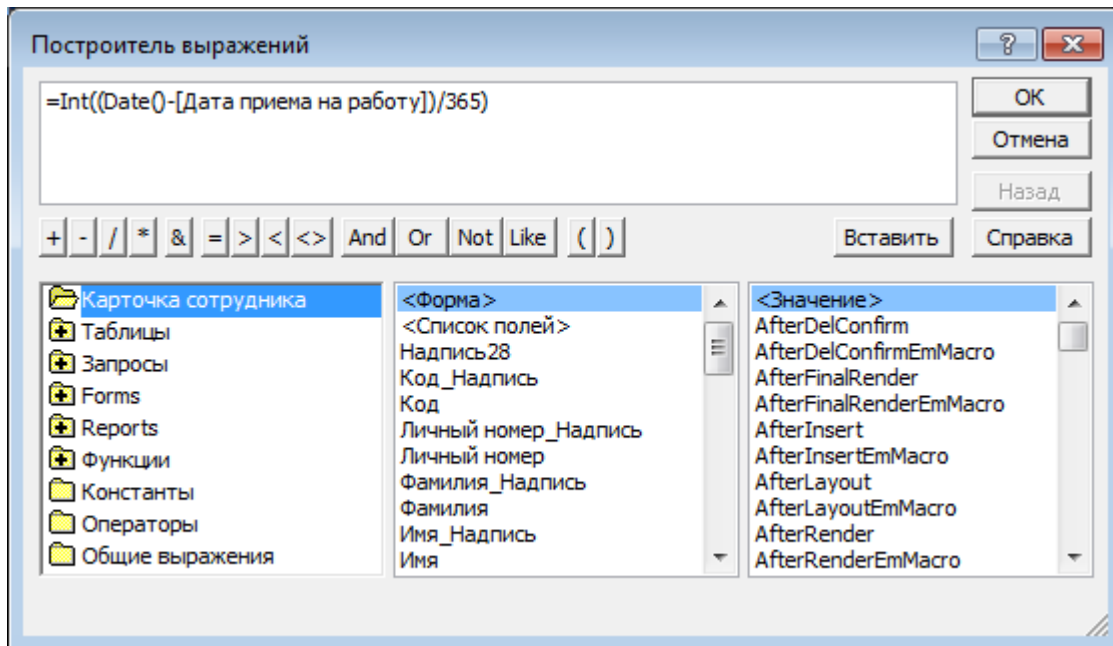


Рис.3.7. Окно построителя выражений

11. На вкладке **Данные** окна диалога свойств вычисляемого поля в строке **Имя** вместо имени по умолчанию введите **Выслуга**. В дальнейшем мы сможем ссылаться на этот элемент по имени **Выслуга**.

12. На вкладке **Макет** в строке **Оформление** выберите **обычное**.

13. На вкладке **Данные** для свойства **Доступ** укажите **Нет**, тогда редактирование данных в этом поле будет запрещено. Закройте окно свойств.

14. Если расположение полей на форме изменялось, то может потребоваться настройка переходов между полями. Для этого необходимо на вкладке **Упорядочить** в группе **Макет** элемента управления выбрать команду **Переходы**. В окне **Последовательность перехода** поля обычно упорядочивают по их фактическому расположению в форме. В нашем случае изменение переходов по полям не требуется.

15. Щелкните на кнопке **Режим ⇌ Режим формы** вкладки **Главная** для просмотра результата создания формы и ввода данных (рис.3.8). Сохраните форму.

16. Введите в форму около 7-10 записей на вымышленных сотрудников. По окончании закройте форму.

**Карточка сотрудника**

Код: 1      Личный номер: A-123546

Фамилия: Иванов      Имя: Иван      Отчество: Иванович

Звание: лейтенант      Оклад по званию: 2000

Должность: инспектор

Дата рождения: 01.01.1980

Дата приема на работу: 01.02.2000      Выслуга: 10

Город: Красноярск

Улица: Ленина      Дом: 120      Квартира: 25

Запись: 1 из 4      Нет фильтра      Поиск

Рис.3.8. Готовая форма **Карточка сотрудника**

### 3.3. ПОДЧИНЕННАЯ ФОРМА

Если между таблицами существует связь *один-ко-многим*, когда одной записи в главной таблице соответствует несколько записей в подчиненной, то для удобства ввода данных необходимо создать подчиненную форму.

1. В ленте перейдите на вкладку **Создание** и в группе **Формы** с помощью кнопки со списком **Другие формы** выполните команду **Мастер форм**.

2. Из таблицы **Карточка сотрудника** в список **Выбранные поля** добавьте последовательно поля **Фамилия**, **Имя**, **Отчество**, а из таблицы **Родственники сотрудника** – **Степень родства**, **Фамилия**, **Имя**, **Отчество**, **Дата рождения**.

3. Щелкните на кнопке **Далее**.

4. В следующем окне диалога выберите тип представления данных – **Карточка сотрудника** и включите переключатель **Подчиненные формы**, щелкните на кнопке **Далее**.

5. Затем выберите внешний вид подчиненной формы **табличный** и щелкните на кнопке **Далее**.

6. В списке следующего окна выберите стиль оформления **Официальная** или любой другой понравившийся вам и снова щелкните на кнопке **Далее**.

7. В последнем окне мастера форм введите имя формы **Родственники сотрудника**, имя подчиненной формы **Родственники сотрудника подчиненная форма** и щелкните на кнопке **Готово**.

8. Щелкните на кнопке **Режим** на вкладке **Главная** и выберите **Конструктор** для перехода в режим конструктора формы.

9. Заголовок формы выровняйте по центру.

10. Выделите подчиненную форму **Родственники сотрудника** и на вкладке **Упорядочить** в группе **Макет элемента управления** выполните команду **Удалить** для разгруппировки входящих полей.

11. Затем щелкните по надписи подчиненной формы **Родственники сотрудника** и удалите ее с помощью клавиши **Delete**.

12. Настройте ширину подчиненной формы на всю область данных основной формы.

13. Перейдите в область макета подчиненной формы **Родственники сотрудника подчиненная форма**.

14. Измените надпись поля **Степень родства** на **Родственники**.

15. Закройте главную форму, сохранив сделанные изменения.

16. Снова откройте форму **Родственники сотрудника** в режиме формы.

17. Перемещая горизонтально границы заголовков полей подчиненной формы, настройте ширину отображения столбцов по вашему усмотрению.

Результат создания подчиненной формы представлен на рисунке 3.9. Теперь переходя по записям основной формы, автоматически будут изменяться данные записей в подчиненной форме.

Добавьте каждому сотруднику по 2-4 записи родственников.

Родственники сотрудника

## Родственники сотрудника

Фамилия:   
 Имя:   
 Отчество:

Родственники	Фамилия	Имя	Отчество	Дата рождения
Мать	Иванова	Людмила	Викторовна	20.03.1960
Отец	Иванов	Иван	Владимирович	16.10.1958
Сестра	Иванова	Ирина	Ивановна	01.01.1985
*				

Запись: 4 из 4    Нет фильтра    Поиск

Запись: 1 из 1    Нет фильтра    Поиск

Рис.3.9. Подчиненная форма Родственники сотрудника

## 4. ЗАПРОСЫ

Запросы являются инструментом поиска и структурирования данных. Запрос, адресованный одной или несколькими таблицам, инициирует выборку определенной части данных и их передачу в таблицу, формируемую самим запросом. В результате вы получаете некоторое подмножество информационного множества исходных таблиц, сформированное по определенному закону. Если обрабатываемый объем информации велик, выделение необходимых данных в такое подмножество позволяет существенно сократить время их обработки.

### 4.1. МАСТЕР ЗАПРОСОВ

Чтобы упростить работу пользователя, в состав Microsoft Access включен мастер запросов, позволяющий автоматизировать процесс построения запроса. С помощью этого мастера выполним выборку информации из базы данных **Кадры**.

1. В ленте перейдите на вкладку **Создание** и в группе **Другие** выберите команду **Мастер запросов**.

2. В окне **Новый запрос** в списке укажите **Простой запрос** и нажмите кнопку **ОК**.

3. В раскрывающемся списке **Таблицы и запросы** первого окна мастера выберите таблицу **Карточка сотрудника** (рис.4.1).

4. В списке **Доступные поля** щелкните на строке **Фамилия**.

5. Щелкните на кнопке **>**, чтобы переместить выделенное поле в список **Выбранные поля**.

6. Повторяя шаги 4-5, добавьте последовательно в список **Выбранные поля** поля **Имя**, **Отчество** из таблицы **Карточка сотрудника**, а также поля **Город**, **Улица**, **Дом**, **Квартира** из таблицы **Адреса сотрудников**.

7. Щелкните на кнопке **Далее**.

8. Введите имя запроса **Адреса** и щелкните на кнопке **Готово**.

Microsoft Access автоматически построит запрос и выполнит его. *Запрос* – это набор условий, согласно которым производится выборка информации из таблиц. Запуск запроса формирует новую таблицу данных, единственным отличием которой от обычных таблиц является то, что с помощью по-

вторных запусков запроса ее содержимое можно обновлять в соответствии с изменением информации источников данных.

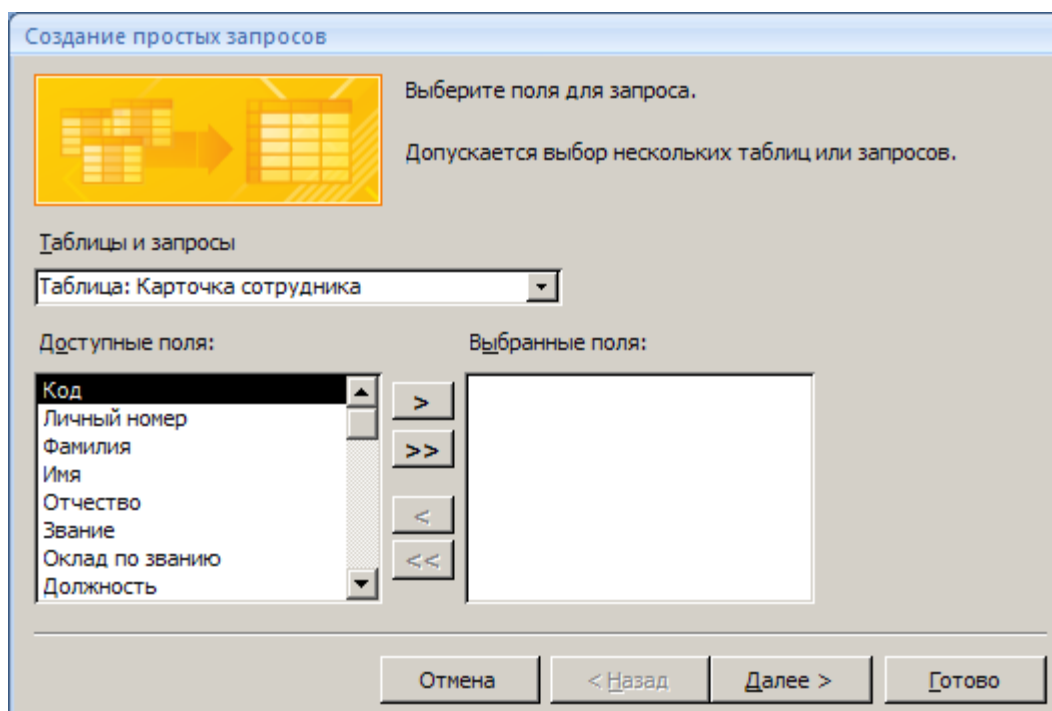


Рис.4.1. Мастер запросов

Примерный результат выполнения запроса показан на рисунке 4.2.

Фамилия	Имя	Отчество	Город	Улица	Дом	Квартира
Иванов	Иван	Иванович	Красноярск	Ленина	120	25
Сидоров	Михаил	Юрьевич	Новосибирск	Горная	18	49
Карамазов	Владимир	Андреевич	Красноярск	Мира	15	3
Петров	Сергей	Александрович	Красноярск	Новосибирска	40	70
*						

Рис.4.2. Результат выполнения запроса Адреса

В рассматриваемом случае условие отбора инициирует получение из таблицы **Карточка сотрудника** полей **Фамилия**, **Имя**, **Отчество**, а также полей **Город**, **Улица**, **Дом**, **Квартира** из таблицы **Адреса сотрудников**. Таблицы **Карточка сотрудника** и **Адреса сотрудников** связаны между собой через поле **Код**, при этом **Карточка сотрудника** является главной таблицей, а **Адреса сотрудников** – подчиненной (при этом каждой записи таблицы **Карточка сотрудника** соответствует одна запись таблицы **Адреса сотрудников**). Поэтому в полях **Город**, **Улица**, **Дом**, **Квартира** результата запроса выводится инфор-

мация из той записи таблицы **Адреса сотрудников**, значение поля **Код** которой совпадает со значением одноименного поля таблицы **Карточка сотрудника**.

## 4.2. КОНСТРУКТОР ЗАПРОСОВ

Мастер запросов умеет конструировать только простые типовые условия отбора. Чтобы наложить дополнительные ограничения, следует пользоваться конструктором запросов, обеспечивающим полное управление параметрами запроса и построение сложных условий отбора данных.

1. Чтобы создать запрос в режиме конструктора, в ленте на вкладке **Создание** в группе **Другие** выберите команду **Конструктор запросов**.

2. Из раскрывшегося списка **Добавление таблицы** выберите и добавьте в запрос таблицы **Карточка сотрудника** и **Родственники сотрудника** (рис.4.3).

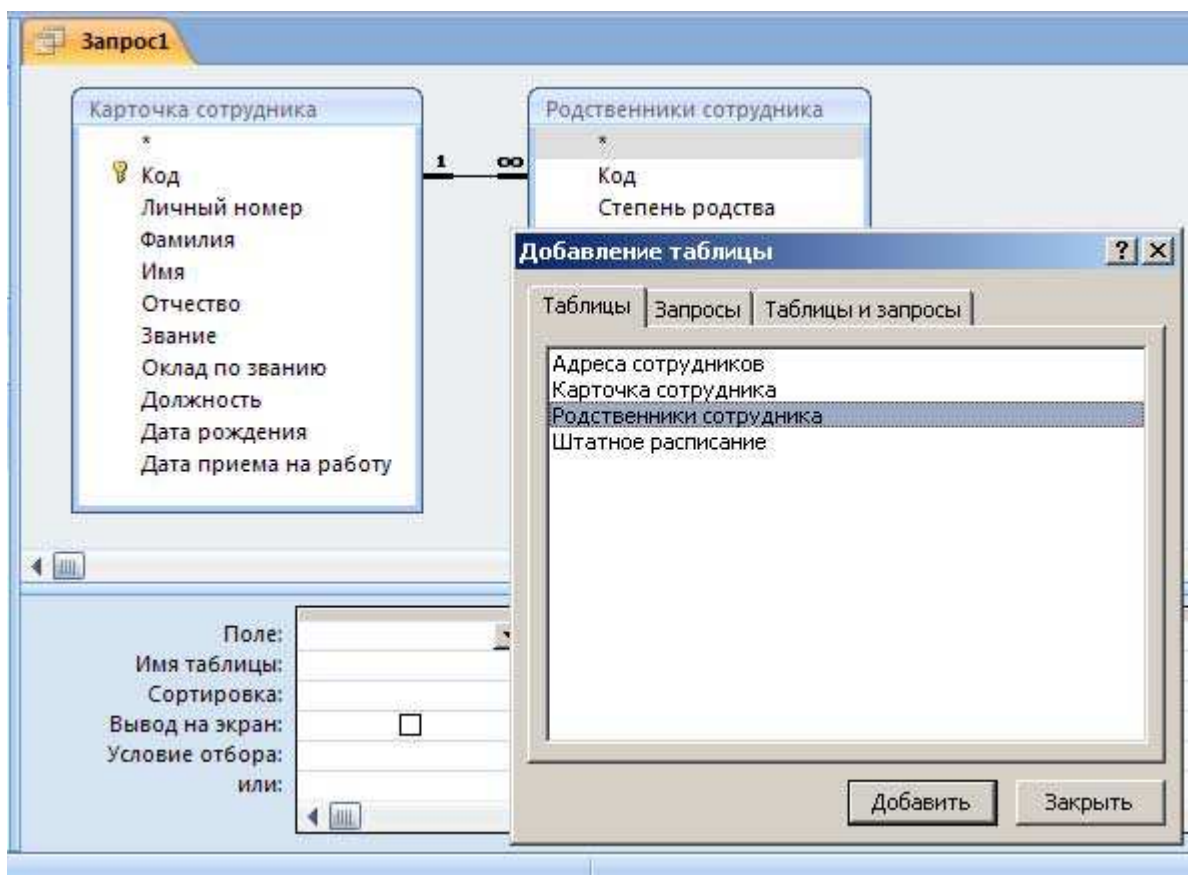


Рис.4.3. Добавление таблиц в конструкторе запросов

В верхней части окна конструктора отображаются списки полей выбранных таблиц и связи между таблицами. Нижняя

область содержит бланк выбора полей таблиц, условий отбора и режимов сортировки.

3. Наведите указатель мыши на поле **Фамилия** таблицы **Карточка сотрудника**.

4. Нажмите кнопку мыши и перетащите поле **Фамилия** в первый столбец бланка запроса (эту же операцию можно выполнить с помощью двойного щелчка мыши на поле **Фамилия**). Его имя появится в первой ячейке, а имя соответствующей таблицы – во второй. Третья строка бланка запроса позволяет сортировать результат запроса по тому или иному полю.

5. В раскрывающемся списке третьей ячейки первого столбца бланка выберите пункт **по возрастанию**. Теперь записи результата запроса будут упорядочены по алфавиту значения поля **Фамилия**.

6. Далее добавьте в бланк запроса поле **Имя** из таблицы **Карточка сотрудника** и поля **Степень родства**, **Фамилия**, **Имя**, **Отчество** из таблицы **Родственники сотрудника**.

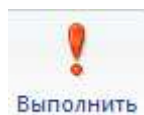
7. Назначьте для поля **Степень родства** режим сортировки **по убыванию**.

Имеющийся вариант связи добавляет в результат запроса только те записи связанных таблиц, в которых значения полей **Код** равны. Так как для некоторых людей из таблицы **Карточка сотрудника** нет записей в таблице **Родственники сотрудника**, то информация о таких людях не включается в результат запроса. Чтобы запрос возвращал данные даже о тех людях (включенных в таблицу **Карточка сотрудника**), у которых нет родственников, следует изменить параметры объединения.

8. Дважды щелкните мышкой по линии связи между таблицами. При этом откроется окно диалога **Параметры объединения** (рис.4.4).

9. Выберите положение переключателя **Объединение ВСЕХ записей из "Карточка сотрудника" и только тех записей из "Родственники сотрудника", в которых связанные поля совпадают**.

10. Щелкните на кнопке **ОК**. На одном конце линии связи появится стрелка, указывающая на смену режима объединения.



11. Щелчком на кнопке **Выполнить** вкладки **Конструктор** выведите результат выполнения запроса (рис.4.5).

12. Закройте окно запроса.

13. В ответ на вопрос о необходимости сохранения макета или структуры запроса ответьте **Да**. Введите имя запроса **Родственники** и нажмите **ОК**.

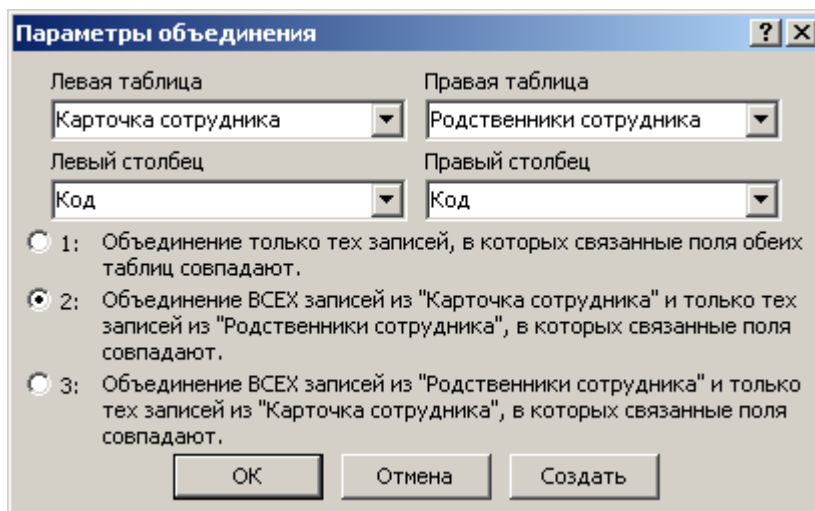


Рис.4.4. Изменение параметров объединения

Карточка сс	Карточка сс	Степень ро	Родственни	Родственни	Отчество
Иванов	Иван	Сестра	Иванова	Ирина	Ивановна
Иванов	Иван	Отец	Иванов	Иван	Владимирови
Иванов	Иван	Мать	Иванова	Людмила	Викторовна
Карамазов	Владимир	Отец	Карамазов	Андрей	Петрович
Карамазов	Владимир	Брат	Карамазов	Сергей	Андреевич
Петров	Сергей				
Сидоров	Михаил	Мать	Сидорова	Елена	Владимировн

Рис.4.5. Результат выполнения запроса **Родственники**

### 4.3. УСЛОВИЯ ОТБОРА И ВЫЧИСЛЯЕМЫЕ ПОЛЯ

Добавление в запрос *условия отбора* позволяет выбирать из таблицы не все записи, а лишь те, которые удовлетворяют определенным критериям. Например, вам необходимо полу-

чить сведения о сотрудниках, проживающих в Красноярске, которым на данный момент исполнилось от 23 до 45 лет.


1. Одним из известных способов создайте запрос с именем **Красноярцы от 23 до 45**, например, с помощью конструктора. В запрос включите следующие поля:

– из таблицы **Карточка сотрудника** – **Фамилия, Имя, Отчество, Звание, Должность, Дата рождения**;

– из таблицы **Адреса сотрудников** – поле **Город**.

2. С помощью контекстного меню войдите в конструктор запроса **Красноярцы от 23 до 45**.

3. Перед полем **Город** вставьте пустой столбец (установите курсор в поле **Город** и выберите команду на вкладке **Конструктор** ⇒ группа **Настройка запроса** ⇒ **Вставить столбцы**). В бланке запроса щелкните правой кнопкой мыши на пустой ячейке в строке **Поле** нового столбца и в контекст-

ном меню выберите команду  **Построить...**. Откроется окно построителя выражений (рис.4.6).

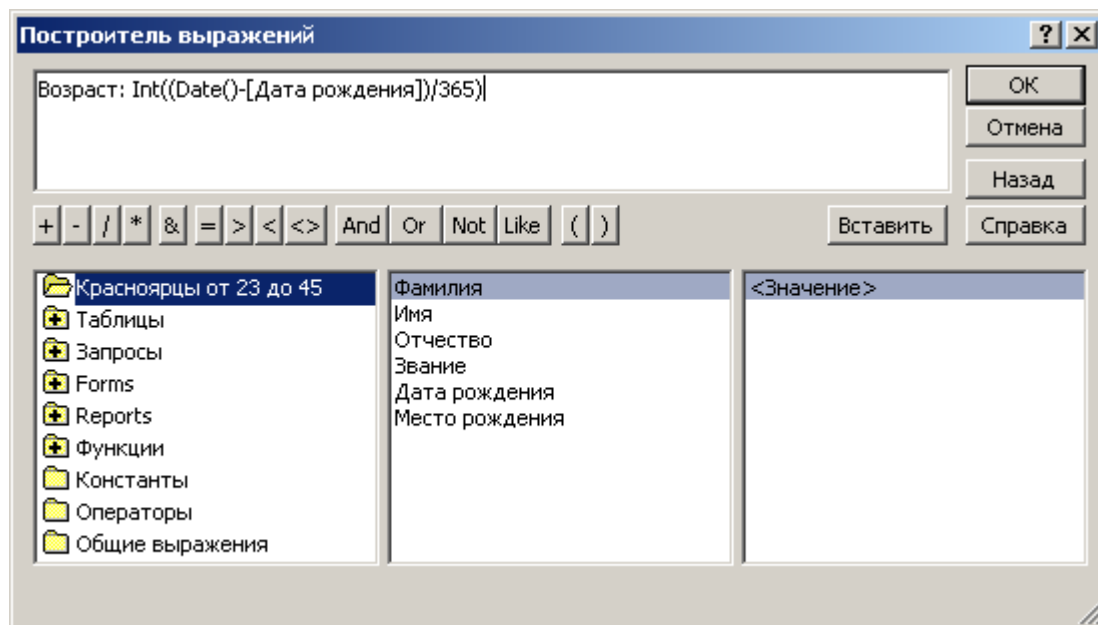


Рис.4.6. Построитель выражений

4. В верхней части окна построителя выражений наберите **Возраст:** (обязательно поставьте двоеточие) – это будет название нового вычисляемого поля. Возраст будем вычислять так: от текущей даты (функция **Date()**) вычтем дату рождения (поле **[Дата рождения]**), поскольку операция вычитания дат возвращает разницу в днях, а возраст принято считать в годах,

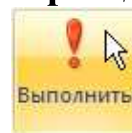
то, поделив результат на 365, получим количество лет. Так как нас интересует полное количество лет, то дробную часть необходимо отбросить (функция **Int()** возвращает целую часть числа).

5. В итоге у вас должна получиться следующая формула:  
**Возраст:Int((Date()-[Дата рождения])/365)**

По ней вычисляется полное количество лет на текущую дату.

6. Щелкните на кнопке **ОК**. Построенная формула появится в ячейке **Поле** добавленного столбца бланка запроса.

При формировании данного запроса может быть не важна конкретная дата рождения сотрудников, поэтому поле **Дата рождения** можно не включать в результат запроса. Для этого сбросьте флажок **Вывод на экран** столбца **Дата рождения**.



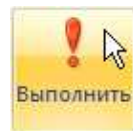
7. Щелкните по кнопке **Выполнить** вкладки **Конструктор** группы **Результаты**, чтобы просмотреть результат выполнения запроса.

8. Теперь нужно добавить ограничение на возраст от 23 до 45 лет. Перейдите в режим конструктора, щелкнув по кнопке



вкладки **Главная**. В поле **Возраст**, в строке **Условие отбора** наберите **>=23 and <=45** (будут выбраны все сотрудники, возраст которых больше либо равен 23 и меньше либо равен 45).

9. В ячейке **Условие отбора** столбца **Город** введите слово **Красноярск**.



10. Щелчком на кнопке **Выполнить** выведите результат выполнения запроса (рис.4.7).

*Задание.* Самостоятельно создайте запрос, вычисляющий выслугу сотрудников (для вычисления выслуги вместо поля **Дата рождения** в формуле используйте поле **Дата приема на работу**). Запрос сохраните под именем **Выслуга**.

С помощью запроса вычислим зарплату сотрудников. Для простоты будем предполагать, что зарплата складывается из

оклада по званию, оклада по должности и районного коэффициента (30% от суммы окладов).

Фамилия	Имя	Отчество	Звание	Возраст	Город
Иванов	Иван	Иванович	лейтенант	30	Красноярск
Карамазов	Владимир	Андреевич	сержант	29	Красноярск
Петров	Сергей	Александрович	майор	33	Красноярск

Рис.4.7. Сотрудники от 23 до 45 лет, родившиеся в Красноярске

Выберите вкладку **Создание** ⇒ группа **Другие** ⇒ **Конструктор запросов**. Нам потребуются таблицы **Карточка сотрудника** и **Штатное расписание**. Из этих таблиц добавьте в запрос следующие поля: **Фамилия**, **Имя**, **Отчество**, **Должность**, **Оклад по званию** и **Оклад**.

Для вычисления зарплаты создадим поле **Зарплата**. В строке **Поле** первого пустого столбца введите **Зарплата:** (для большего удобства и наглядности воспользуйтесь построителем выражений из контекстного меню). Это будет заголовок столбца, обязательно поставьте двоеточие (оно отделяет заголовок от формулы вычислений), а затем следующее выражение (рис.4.8):

**[Оклад]+[Оклад по званию]+([Оклад]+[Оклад по званию])\*0,3**

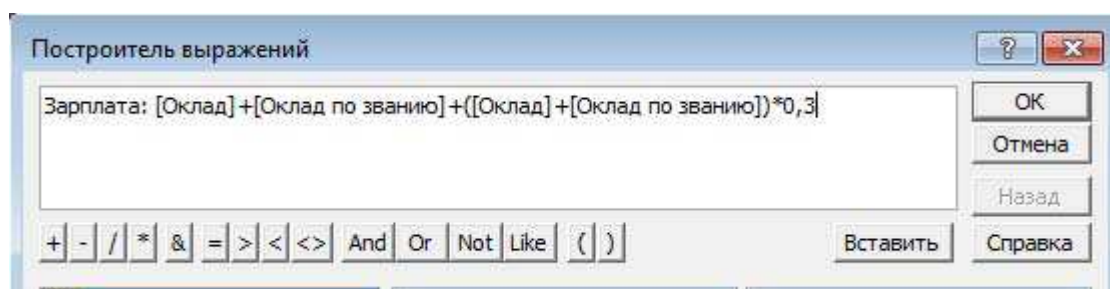


Рис.4.8. Выражение для вычислимого поля **Зарплата**

Щелчком на кнопке **Выполнить** вкладки **Конструктор** выведите результат выполнения запроса. Запрос сохраните под именем **Зарплата**.

**Оператор Like.** Оператор **Like** очень удобен для сравнения строк, при этом для большей гибкости могут использоваться символы подстановки: **?** – заменяет один символ, **\*** – заменяет любое количество символов.

Приведем примеры использования оператора **Like**. Если в запросе в столбце **Имя** в строке **Условие отбора** указать **Like "?рина"**, то, выполнив запрос, получим список сотрудников, у которых имя состоит из пяти букв и оканчивается на «рина», (т.е. Ирина, Арина и др.). А если в столбце **Имя** в строке **Условие отбора** указать **Like "\*рина"**, то, выполнив запрос, получим список сотрудников, в который кроме указанных имен, дополнительно попадут Марина, Екатерина, Александрина и др., т.е. те имена, которые имеют окончание «рина», но могут состоять из любого количества символов.

Если в запросе в столбце **Фамилия** в строке **Условие отбора** указать **Like "С\*"**, то, выполнив запрос, получим список сотрудников, фамилия которых начинается на букву «С». Если в строке **Условие отбора** указать **Like "\*ов"**, то получим список сотрудников, чья фамилия оканчивается на «ов». Если в строке **Условие отбора** указать **Like "[ИВ]\*"**, то получим список сотрудников, чья фамилия начинается на «И» или «В».

*Задание.* Самостоятельно создайте запрос, который выдает список сотрудников, проживающих на улицах, начинающихся на букву «К». Если в базе данных нет улиц начинающихся на букву «К», то условие запроса можно изменить по ситуации.

## 5. ОТЧЕТЫ

В целом отчеты похожи на формы, но они, как правило, предназначены для вывода информации из базы данных на принтер. Поэтому в отчетах данные форматируют так, чтобы их было удобно размещать на отдельных страницах. Отчеты поддерживают самые разнообразные способы оформления и позволяют группировать данные, разбивая их на логически цельные блоки.

### 5.1. МАСТЕР ОТЧЕТОВ

Чтобы облегчить работу пользователя, в Microsoft Access имеется специальный мастер, который при недостатке времени позволяет быстро создавать достаточно привлекательные отчеты. Создадим отчет, выводящий информацию о родственниках сотрудников:

1. Перейдите в ленте на вкладку **Создание**.
2. Щелкните по кнопке **Мастер отчетов** группы **Отчеты**.
3. Добавьте в список **Выбранные поля** из таблицы **Карточка сотрудника** поля **Фамилия**, **Имя**, **Отчество**, из таблицы **Родственники сотрудника** все поля, кроме поля **Код**.  
Поле **Код** содержит служебную информация, предназначенную для обеспечения связи данных таблиц. Поэтому помещать такие поля в отчет не имеет смысла.  
Щелкните на кнопке **Далее**.
4. Выберите вид представления данных – **Карточка Сотрудника** и нажмите **Далее**.
5. В следующем окне диалога не добавляйте уровней группировки. Щелкните на кнопке **Далее**.
6. Четвертое окно мастера позволяет задать порядок сортировки записей. Выберите в первом списке имя поля **Степень родства**, щелкните на кнопке сортировки справа от списка, чтобы сортировка производилась **по убыванию**, и вновь на кнопке **Далее**.

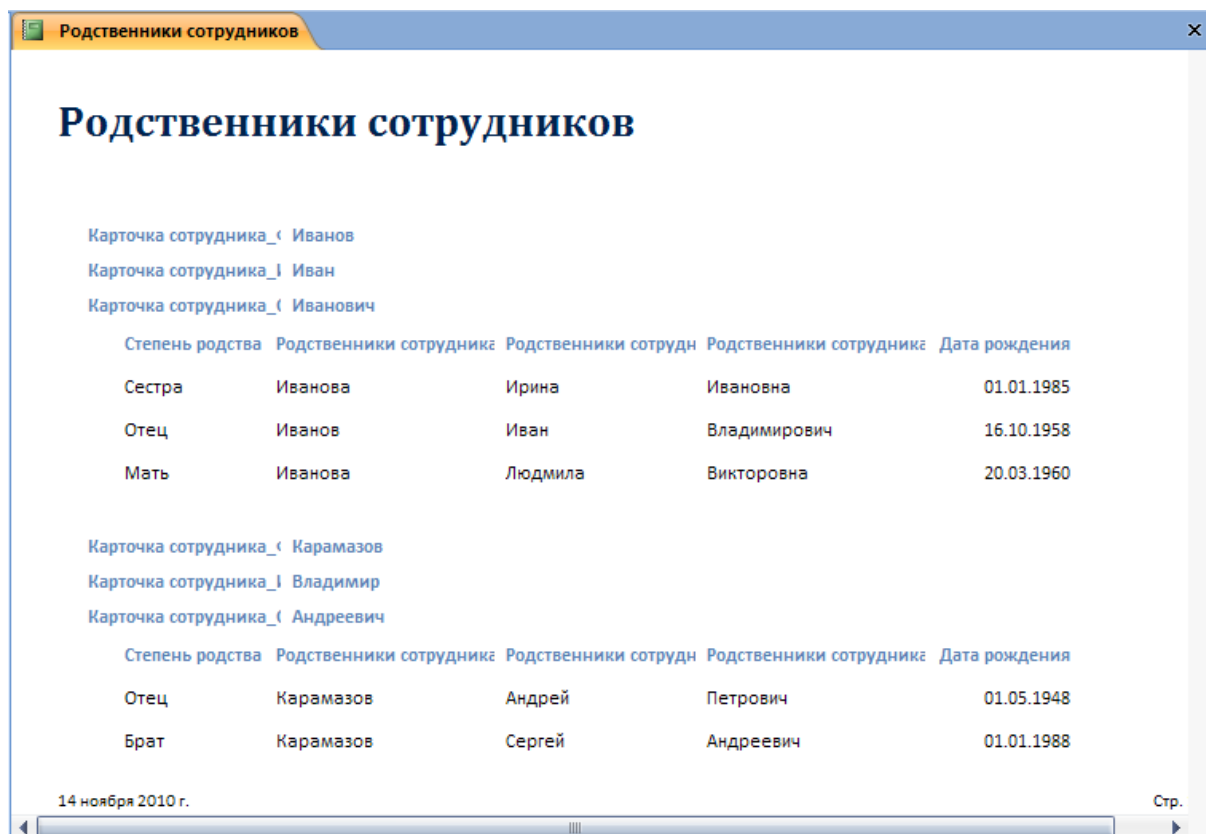
Следующее окно позволяет выбрать вид макета для отчета и ориентацию страницы.

7. Выберите макет **Структура** и щелкните на кнопке **Далее**.

8. В следующем окне мастера выберите стиль оформления **Стандартная**. Затем щелкните на кнопке **Далее**.

9. В последнем окне мастера отчетов введите имя отчета **Родственники сотрудников** и щелкните на кнопке **Готово**.

Мастер сформирует отчет и откроет его в режиме предварительного просмотра (рис.5.1): сначала фамилия, имя, отчество сотрудника, информация о его родственниках, затем располагаются те же данные, касающиеся второго человека, и т.д.



The screenshot shows a window titled "Родственники сотрудников" with a blue header. The main content area displays the title "Родственники сотрудников" in large blue font. Below it, there are three sections, each starting with "Карточка сотрудника\_<" followed by a name. Each section contains a table of relatives with columns: "Степень родства", "Родственники сотрудник", "Родственники сотрудн", "Родственники сотрудник", and "Дата рождения".

Степень родства	Родственники сотрудник	Родственники сотрудн	Родственники сотрудник	Дата рождения
Сестра	Иванова	Ирина	Ивановна	01.01.1985
Отец	Иванов	Иван	Владимирович	16.10.1958
Мать	Иванова	Людмила	Викторовна	20.03.1960

Степень родства	Родственники сотрудник	Родственники сотрудн	Родственники сотрудник	Дата рождения
Отец	Карамазов	Андрей	Петрович	01.05.1948
Брат	Карамазов	Сергей	Андреевич	01.01.1988

At the bottom left, the date "14 ноября 2010 г." is displayed, and at the bottom right, there is a "Стр." label and a scrollbar.

Рис.5.1. Фрагмент отчета, сформированного мастером

## 5.2. КОНСТРУКТОР ОТЧЕТОВ

В созданном отчете присутствует множество недостатков, которые можно исправить с помощью конструктора отчетов.

1. Щелчком на кнопке **Режим** вкладки **Главная** перейдите в режим конструктора отчета **Родственники сотрудников** (рис.5.2). Либо, если вы только что загрузили базу данных, перейдите в области переходов в группу **Отчеты**, выберите отчет **Родственники сотрудников** и правым щелчком мыши перейдите в контекстное меню, где выполните команду **Конструктор**.

2. В схеме отчета присутствует несколько разделов. Их назначение указано в таблице 5.1. Поля и подписи можно размещать в любом разделе отчета, регулируя этим их положение на странице.

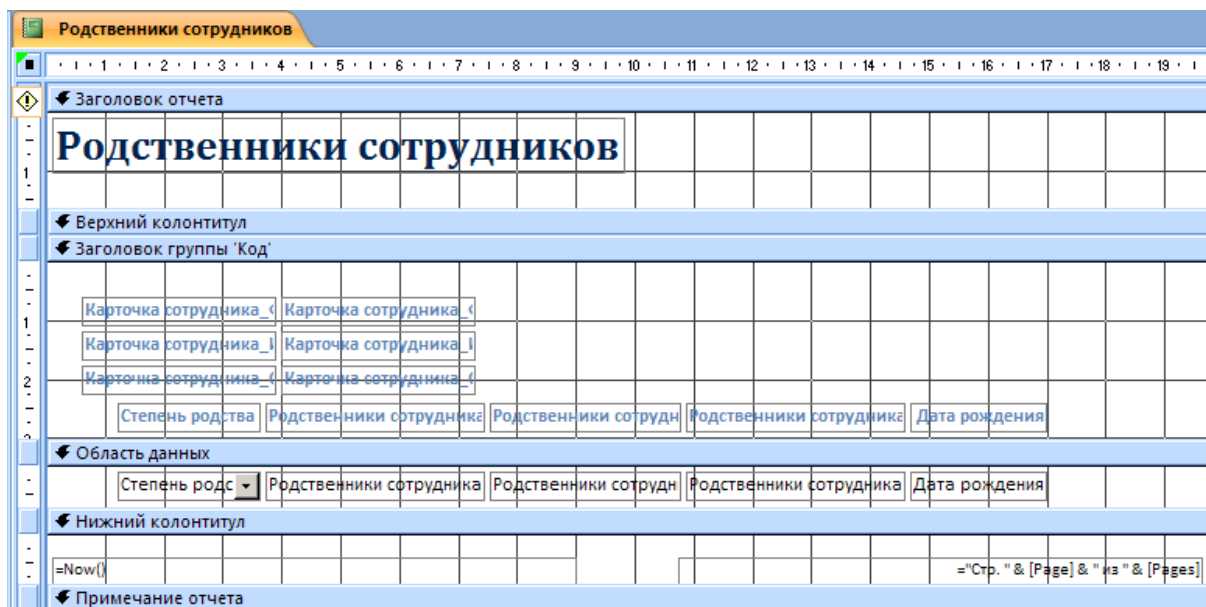


Рис.5.2. Конструктор отчетов

Таблица 5.1

### РАЗДЕЛЫ ОТЧЕТА

Раздел	Назначение
Заголовок отчета	Печатается в самом начале отчета один раз. Выводится и скрывается командой вкладки <b>Упорядочить</b> ⇒ группы <b>Отображение</b> ⇒ <b>Заголовков/примечание отчета</b>
Верхний колонтитул	Печатается в верхнем поле каждой страницы. Выводится и скрывается командой вкладки <b>Упорядочить</b> ⇒ группы <b>Отображение</b> ⇒ <b>Колонтитулы страницы</b>
Заголовок группы <b>Код</b>	Появляется в начале каждой группы записей, соответствующих очередному значению поля <b>Код</b>
Область данных	Основные данные отчета
Нижний колонтитул	Печатается в нижнем поле каждой страницы
Примечание отчета	Печатается в самом конце отчета

3. Для начала изменения элементов отчета выделите мышкой группу полей **Карточка сотрудника\_Фамилия, Карточка сотрудника\_Имя, Карточка сотрудника\_Отчество**. В ленте перейдите на вкладку **Упорядочить** и в группе **Макет элемента управления** выполните команду **Удалить**. Будет удален макет отображения полей по умолчанию.

4. Затем щелчком выделите надпись поля **Карточка сотрудника\_Фамилия**.

5. Удалите надпись нажатием клавиши **Delete**.

6. Повторяя шаги 4 и 5, удалите из раздела **Заголовки группы 'Код'** надписи **Карточка сотрудника\_Имя** и **Карточка сотрудника\_Отчество**. Назначение соответствующих полей очевидно и не нуждается в пояснении.

7. Разместите поля **Фамилия, Имя, Отчество** из таблицы **Карточка сотрудника** в одну строку. Установите для данных полей размер шрифта 14.

8. Щелчком выделите поле **Карточка сотрудника\_Фамилия**. Увеличьте его горизонтальный размер, перетащив вправо серединный маркер правой границы поля.

9. Повторите шаг 8 для полей **Карточка сотрудника\_Имя** и **Карточка сотрудника\_Отчество**.

10. Заголовок отчета **Родственники сотрудников** разместите по центру страницы.

11. В области **Заголовки группы 'Код'** выделите все надписи полей из таблицы **Родственники сотрудника**. В ленте перейдите на вкладку **Упорядочить** и в группе **Макет элемента управления** выполните команду **Удалить**.

12. В области заголовка группы измените надпись **Степень родства** на **Родственники**, удалите из оставшихся надписей название таблицы **Родственники сотрудника\_** оставив наименование полей: **Фамилия, Имя, Отчество**.

13. Разместите надписи с полями из таблицы **Родственники сотрудника** по ширине страницы.

14. Удерживая нажатой клавишу **Shift**, щелчками мыши выделите все объекты, находящиеся в области заголовка группы **Код**.

15. Используя маркер перемещения, перетащите выделенные объекты выше и уменьшите по вертикали область раздела **Заголовки группы 'Код'**.

16. На вкладке **Главная** в группе **Режимы** выполните команду **Представление отчета** для перехода в режим предварительного просмотра.

На рисунке 5.3 показан фрагмент нового варианта отчета **Родственники сотрудников**.

В нижней части страницы отчета выводится текущая дата и номер страницы. Эта информация генерируется элементами управления, размещенными в разделе **Нижний колонтитул** конструктора отчета.

Закройте отчет. При появлении запроса о необходимости сохранения изменений макета или структуры отчета щелкните на кнопке **Да**.

Родственники сотрудников				
Родственники сотрудников				
Иванов	Иван	Иванович		
Родственники	Фамилия	Имя	Отчество	Дата рождения
Сестра	Иванова	Ирина	Ивановна	01.01.1985
Отец	Иванов	Иван	Владимирович	16.10.1958
Мать	Иванова	Людмила	Викторовна	20.03.1960
Сидоров	Михаил	Юрьевич		
Родственники	Фамилия	Имя	Отчество	Дата рождения
Мать	Сидорова	Елена	Владимировна	23.07.1961
Карамазов	Владимир	Андреевич		
Родственники	Фамилия	Имя	Отчество	Дата рождения
Отец	Карамазов	Андрей	Петрович	01.05.1948
Брат	Карамазов	Сергей	Андреевич	01.01.1988
14 ноября 2010 г.				Стр. 1 из 1

Рис.5.3. Фрагмент нового варианта отчета

### 5.3. СОРТИРОВКА И ГРУППИРОВКА

Отчет, который мы сконструировали, имеет еще один существенный недостаток. Список сотрудников выводится не по алфавиту. Мастер отчетов автоматически сгруппировал данные таблицы **Родственники сотрудника** по их принадлежности к тем или иным записям таблицы **Карточка сотрудника**,

которая определяется связью между полями **Код** этих двух таблиц. По указанной причине сотрудники в отчете упорядочены по возрастанию числового параметра **Код** – идентификатора записей таблицы **Карточка сотрудника**. Это неудобно, в отчетах с большим числом записей при такой сортировке будет довольно трудно отыскать нужного человека. Есть смысл изменить порядок сортировки.

1. Откройте отчет **Родственники сотрудников** в режиме конструктора.

2. На вкладке **Конструктор** группы **Группировка и итоги** выполните команду **Группировка**, чтобы открыть в нижней части экрана окно диалога, показанное на рисунке 5.4.

В этом окне назначены два уровня сортировки. Сначала в порядке возрастания по полю **Код**, а затем в пределах списка родственников, относящихся к одному сотруднику, по убыванию степени родства. Верхняя строка окна **Группировка, сортировка и итоги** указывает, что записи группируются по полю **Код**, то есть при переборе записей таблицы **Карточка сотрудника** для каждого нового значения поля **Код** генерируется заголовок группы, в котором для нашего отчета отображаются фамилия, имя, отчество сотрудника и надписи полей из таблицы **Родственники сотрудника**.

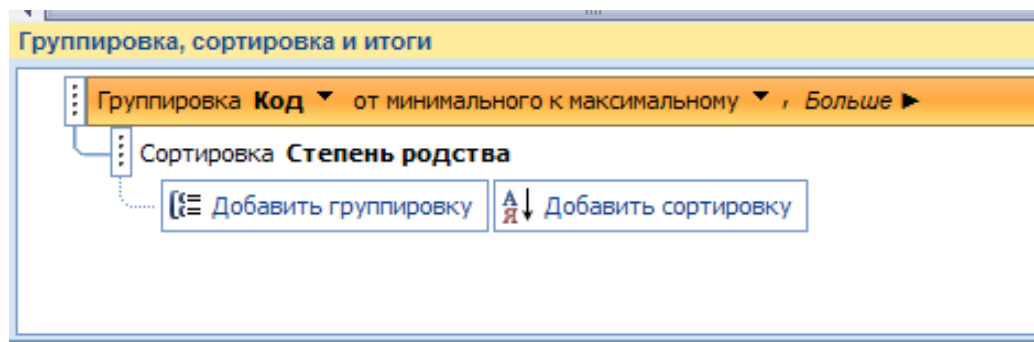



Рис.5.4. Сортировка и группировка

Казалось бы, можно просто изменить поле сортировки **Код** на **Фамилия**, и записи в отчете сразу же выстроятся по алфавиту. Но в этом случае для четырех людей с одинаковой фамилией (например, Петровский) будет выведен только один заголовок группы, ниже которого разместится список родственников всех Петровских. Вспомните, что новый заголовок группы выводится только для каждого нового значения поля, по которому выполняется группировка. **Код** – уникальный идентификатор записей таблицы **Карточка сотрудника**, груп-

пировка по которому обеспечивает вывод в отчет всех записей этой таблицы.

Для сортировки записей отчета по алфавиту фамилий и имен следует добавить в список окна диалога **Группировка, сортировка и итоги** соответствующие поля, разместив их выше поля группировки **Код**.


3. В окне **Группировка, сортировка и итоги** щелкните на кнопке .

4. В появившемся списке полей выберите пункт **Карточка сотрудника\_Фамилия**.

5. Аналогично добавьте сортировку по полю **Карточка сотрудника\_Имя**.

При добавлении нового поля сортировки Microsoft Access автоматически назначает ему порядок сортировки по возрастанию **начиная с А**. Чтобы реализовать обратный порядок записей, выберите в раскрывающемся списке строки соответствующего поля порядок сортировки **начиная с Z**.

6. Щелчком по строке Сортировка **Карточка сотрудника\_Фамилия** выделите ее.

7. Удерживая нажатой левую кнопку мыши в области выбора поля  переместите его перед полем группировки **Код**.

8. То же самое проделайте с полем сортировки **Карточка сотрудника\_Имя**, переместив его непосредственно после поля **Фамилия**. Новый вид окна **Группировка, сортировка и итоги** представлен на рисунке 5.5.

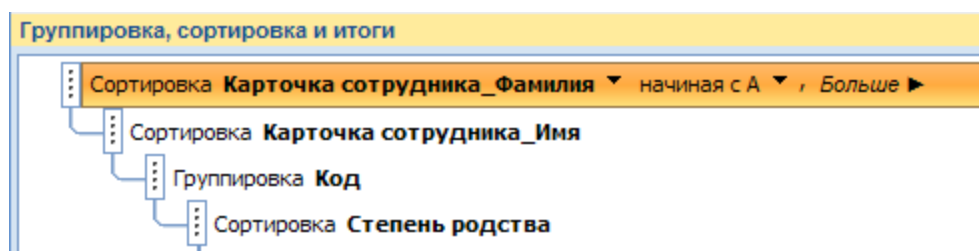


Рис.5.5. Сортировка и группировка

9. Закройте окно **Сортировка и группировка**.

10. Перейдите в режим **Представление отчета** на вкладке **Главная**.

Теперь записи сначала сортируются по фамилиям, затем по именам. После этого для каждой новой записи таблицы **Карточка сотрудника** появляется заголовок группы (так как

значения поля **Код** для всех записей различны). Ниже заголовка группы из таблицы **Родственники сотрудника** выводятся все родственники, соответствующие данному значению величины **Код**, то есть данному сотруднику.

Заголовки тех групп, для которых нет ни одной записи в таблице **Родственники сотрудника**, в отчет не выводятся.

11. Закройте отчет и сохраните все изменения.

#### 5.4. ПОДВЕДЕНИЕ ПРОМЕЖУТОЧНЫХ ИТОГОВ

Пусть нам необходимо создать отчет, содержащий:

- список сотрудников по подразделениям;
- фонд заработной платы по подразделениям;
- среднюю зарплату по подразделениям;
- фонд заработной платы предприятия.

1. Выполните на вкладке **Создание** группы **Отчеты** команду **Мастер отчетов**.

2. Перенесите в список **Выбранные поля** все поля из запроса **Зарплата**, а из таблицы **Штатное расписание** – поле **Подразделение** и нажмите кнопку **Далее**.

3. В следующем окне предлагается выбрать вид представления данных. По заданию данные должны быть сгруппированы по подразделению. Поэтому в списке источников данных слева выберите таблицу **Штатное расписание**, при этом в окне справа появится уровень группировки по полю **Подразделение**. Нажмите кнопку **Далее**.

4. В следующем окне предлагается добавить дополнительные уровни группировки. Оставьте данные окна без изменения и нажмите **Далее**.

5. Затем выберите сортировку по полям **Должность**, **Фамилия**, **Имя** и **Отчество** по возрастанию. В этом же окне нажмите кнопку **Итоги...** Для поля **Зарплата** поставьте галочки в столбцах, озаглавленных **Sum** (для вычисления суммы) и **Avg** (для вычисления среднего значения) (рис. 5.6), выберите опцию **Показать: данные и итоги**. Нажмите **ОК** и **Далее**.

6. На следующем экране мастера укажите макет **Структура**, **Ориентация** – **книжная** и нажмите **Далее**. Выберите понравившийся стиль, например **Стандартная** и нажмите **Далее**. Сохраните отчет под именем **Зарплата сотрудников по подразделениям**.

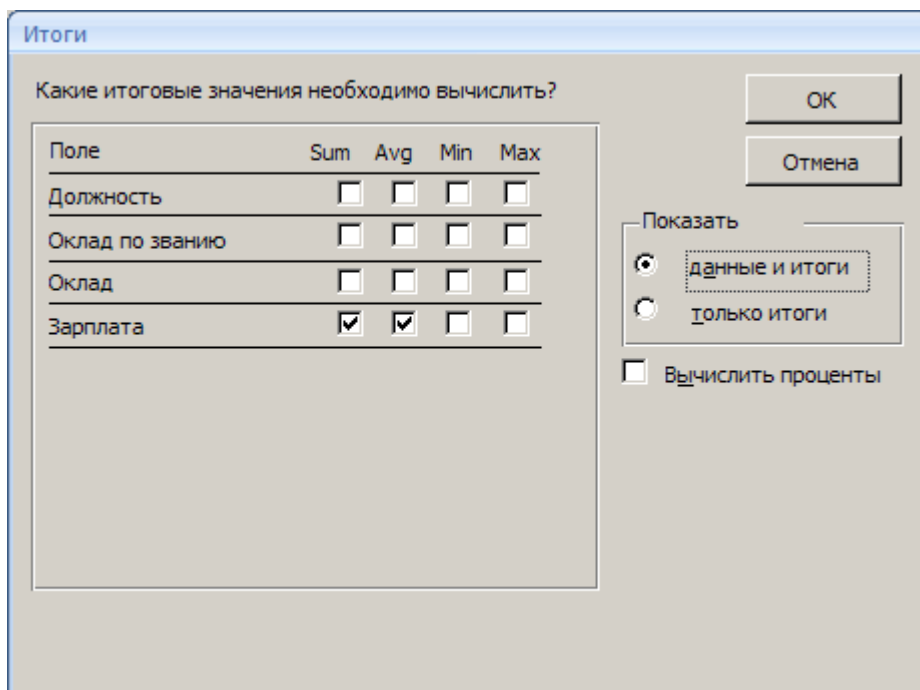


Рис.5.6. Окно выбора итогов

Теперь придадим отчету более привлекательный вид.

1. Перейдите в режим конструктора отчета (рис.5.7). В области заголовка отчета измените название на **Список сотрудников по подразделениям**.

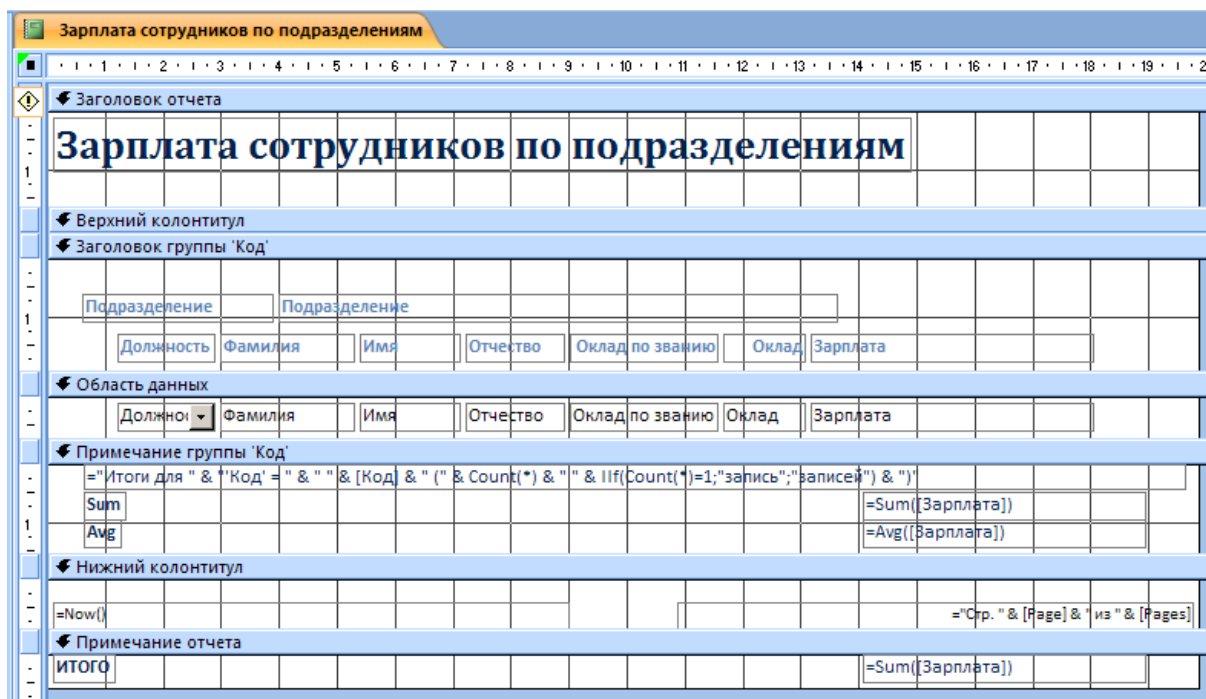


Рис.5.7. Конструктор отчета **Зарплата сотрудников по подразделениям**

2. Передвиньте элементы отчета к левому краю. Увеличьте размер шрифта поля Подразделение. Содержимое и подписи полей **Должность, Фамилия, Имя, Отчество** выровняйте по левому краю, а поля **Оклад по званию, Оклад, Зарплата** – по правому.

3. Перейдите в область примечания группы '**Код**'. Удалите первую надпись "**Итоги для....**". Надписи **Sum** и **Avg** переименуйте соответственно на: **Суммарная зарплата по подразделению** и **Средняя зарплата по подразделению**.

4. Для полей **Sum-Зарплата** и **Avg-Зарплата** в **Окне свойств** справа от отчета, на вкладке **Макет** установите поочередно свойство **Формат поля – Денежный** (рис. 5.8).

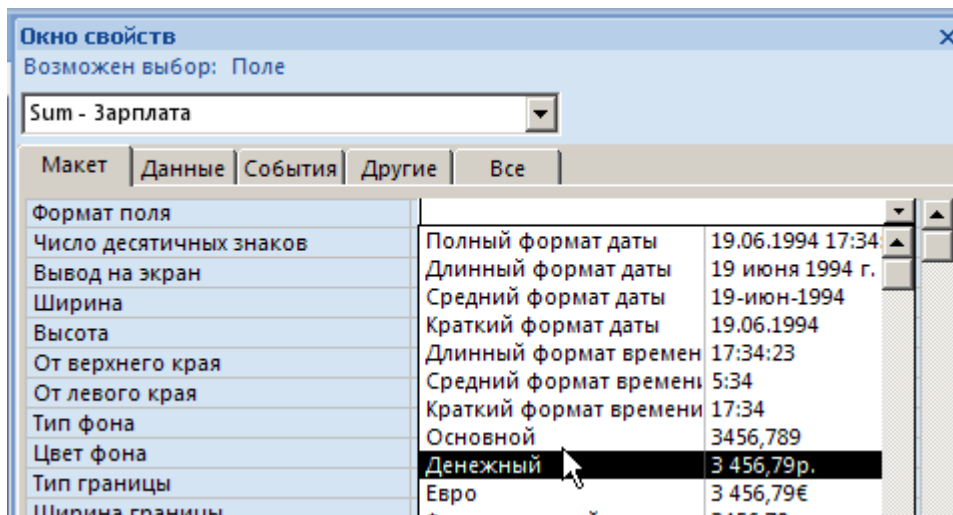


Рис.5.8. Выбор формата поля для полей **Sum-Зарплата** и **Avg-Зарплата**

5. Область примечаний группы сделайте по вертикали немного больше, чтобы одна группа отделялась от другой пустыми строками.

6. Перейдите в область примечаний отчета. Переименуйте надпись **ИТОГО** в **ФОНД ЗАРАБОТНОЙ ПЛАТЫ ПРЕДПРИЯТИЯ**.

7. В области примечаний отчета для поля **Зарплата ИТОГО Sum** также установите **Формат поля – Денежный**.

8. На вкладке **Конструктор** в группе **Режимы** выберите из списка команду **Представление отчета** (рис. 5.9).

## Список сотрудников по подразделениям

### Подразделение Следствие

Должность	Фамилия	Имя	Отчество	Оклад по званию	Оклад	Зарплата
следователь	Петров	Сергей	Александро	3200	4000	9360
Суммарная зарплата по подразделению						9 360,00р.
Средняя зарплата по подразделению						9 360,00р.

### Подразделение Уголовный розыск

Должность	Фамилия	Имя	Отчество	Оклад по званию	Оклад	Зарплата
оперуполн	Сидоров	Михаил	Юрьевич	3000	3900	8970
Суммарная зарплата по подразделению						8 970,00р.
Средняя зарплата по подразделению						8 970,00р.

### Подразделение Дежурная часть

Должность	Фамилия	Имя	Отчество	Оклад по званию	Оклад	Зарплата
водитель	Карамазов	Владимир	Андреевич	2300	3500	7540
Суммарная зарплата по подразделению						7 540,00р.
Средняя зарплата по подразделению						7 540,00р.

### Подразделение Отдел кадров

Должность	Фамилия	Имя	Отчество	Оклад по званию	Оклад	Зарплата
инспектор	Иванов	Иван	Иванович	2000	3600	7280
Суммарная зарплата по подразделению						7 280,00р.
Средняя зарплата по подразделению						7 280,00р.

ФОНД ЗАРАБОТНОЙ ПЛАТЫ ПРЕДПРИЯТИЯ						33 150,00р.
-----------------------------------	--	--	--	--	--	-------------

Рис.5.9. Окончательный вид отчета **Зарплата сотрудников по подразделениям**

## 6. МАКРОСЫ И МОДУЛИ


### 6.1. МАКРОСЫ

Microsoft Office Access позволяет пользователю разрабатывать собственные процедуры работы с данными. Для этого используется алгоритмический язык Visual Basic. Ряд часто выполняемых действий может быть оформлен в виде макрокоманд и назначен командным кнопкам. Рассмотрим процесс создания кнопки в форме ввода, нажатие на которую открывает таблицу запроса **Родственники**.

*Макросы* — это последовательности команд, соответствующие выполнению пользователем ряда пунктов меню. Каждой последовательности может быть присвоено имя. Некоторые последовательности могут содержать параметры, определяемые значениями полей таблиц.

Самые простые действия — это открытие (закрытие) форм ввода-вывода данных или выполнение запросов. Ранее вами был создан запрос **Родственники**. Создадим макрос с одноименным названием, который бы открывал таблицу данного запроса.

Перед началом работы проверим разрешения безопасности Microsoft Access запускать макросы. Для этого нажмите на

кнопку Office  в левом верхнем углу экрана и выберите команду **Параметры Access**. Слева в списке команд выберите **Центр управления безопасностью**, а справа в окне нажмите кнопку **Параметры центра управления безопасностью**. Затем в списке выберите **Параметры макросов** и справа отметьте **Включить все макросы (не рекомендуется, возможен запуск опасной программы)**. Подтвердите все свои действия кнопкой **ОК**.

1. Откройте вкладку **Создание** и в группе **Другие** нажмите на кнопку **Макрос** ⇨ **Макрос**.

2. В поле со списком **Макрокоманда** выберите значение **ОткрытьЗапрос** (рис.6.1). В области **Аргументы макрокоманды** задайте имя запроса **Родственники**, режим **Таблица**, режим данных **Только чтение**.

3. Закройте окно конструктора и сохраните макрос под именем **Родственники**.

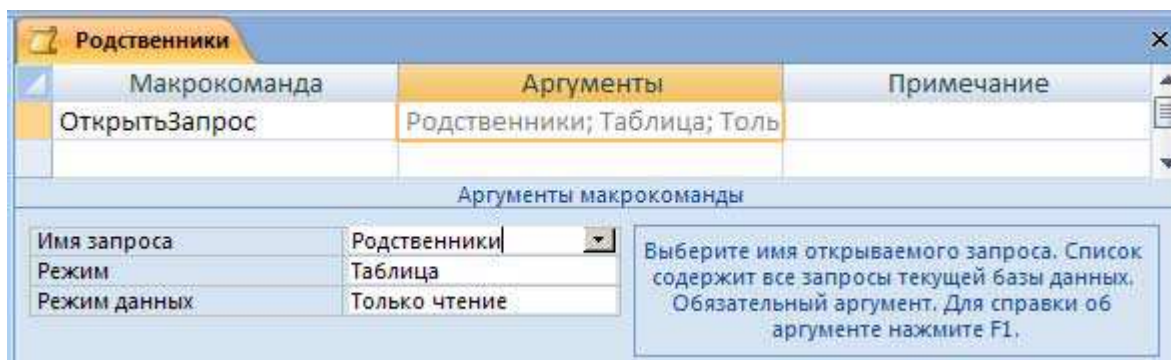

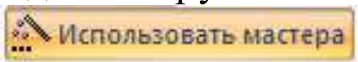


Рис.6.1. Окно конструктора макроса

Для выполнения разработанных макросов могут быть созданы специальные формы или включены элементы управления в уже существующие. Предположим, что при работе с формой **Карточка сотрудника** у пользователя возникает потребность просмотреть список родственников. Для этой цели разместим в форме командную кнопку и свяжем её нажатие с выполнением макроса **Родственники**.

1. В области переходов перейдите в группу **Формы**. Выберите форму **Карточка сотрудника** и в контекстном меню выполните команду **Конструктор**.

2. Наведите курсор на нижнюю границу формы (курсор при этом принимает вид двунаправленной стрелки ) и, нажав левую клавишу мыши, протяните ее немного вниз, увеличив тем самым область примечания формы.

3. На вкладке **Создание** группы **Элементы управления** убедитесь, что кнопка  **Использовать мастера** активирована. Затем щелкните по объекту **Кнопка** и укажите положение кнопки, щелкнув мышью в области примечания формы.

4. В окне **Создание кнопок** выберите категорию **Разное**, действия – **Выполнить макрос**. Следуя инструкциям **Мастера**, назначьте кнопке выполнение макроса **Родственники**. На кнопке поместите текст **Родственники**. Если **Мастер создания кнопок** не загружается, то, щелкнув правой кнопкой мыши по созданной кнопке, выберите в контекстном меню **Свойства**. Перейдите в **Окне свойств** на вкладку **События**, выберите событие **Нажатие кнопки**, в раскрывающемся списке укажите макрос **Родственники**.

5. Общий вид формы в режиме конструктора представлен на рисунке 6.2.

Рис.6.2. Конструктор формы **Карточка сотрудника**


6. Закройте и сохраните форму **Карточка сотрудника**.

Теперь при работе с формой **Карточка сотрудника** пользователь может в любой момент открыть запрос **Родственники**, нажав на командную кнопку.

## 6.2. МОДУЛИ

*Модули* — это программные элементы на языке Visual Basic, выполнение которых связывают с элементами управления форм. Для написания модулей необходима специальная подготовка по программированию. В качестве примера рассмотрим модуль, проверяющий, правильно ли введено значение в поле **Дата приема на работу**. Поскольку при приеме на работу сотрудников МВД существует ограничение по возрасту, то предлагается создать процедуру, которая позволяла бы отследить прием на работу лиц, возраст которых превышает 35 лет.

1. Используя контекстное меню, откройте форму **Карточка сотрудника** в режиме **Конструктор**.

2. Щелкнув правой кнопкой мыши по полю **Дата приема на работу**, выберите в контекстном меню опцию **Свойства**. На вкладке **События** для свойства **До обновления** задайте значение **Процедура обработки события**. Нажмите кнопку построения процедуры  (рис 6.3).

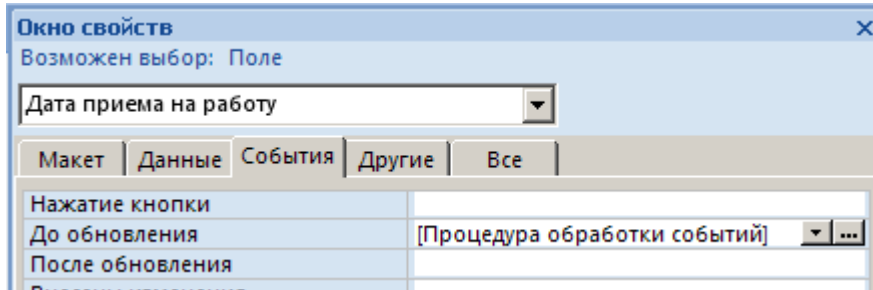


Рис.6.3. Свойства поля **Дата приема на работу**

3. Вставьте в предложенный шаблон (рис 6.4) текст программы

```
If int(([Дата приема на работу]-[Дата рождения])/365)
>= 35 Then
    MsgBox ("Возраст принимаемого сотрудника превышает 35 лет! Повторите ввод!")
    Cancel = True
End If
```

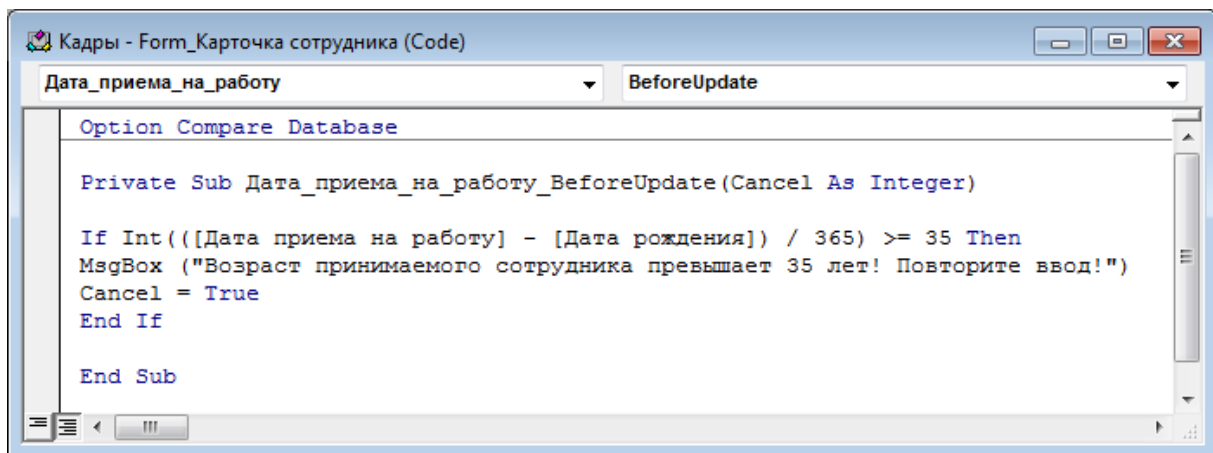


Рис.6.4. Окно редактора Microsoft Visual Basic

Для проверки синтаксиса введенных выражений выберите в меню Microsoft Visual Basic опцию **Debug** ⇔ **Compile Кадры**. В случае обнаружения ошибки строка, содержащая ее, будет подсвечена желтым цветом.

Закройте окно редактора Microsoft Visual Basic. Изменения в модуле сохраняются автоматически после закрытия формы **Карточка сотрудника**.

Приведенная процедура выполняется при вводе значений в поле **Дата приема на работу**. Процедура вычисляет возраст сотрудника в день приема на работу, и если он превышает 35 лет, то на экран выдается сообщение о неверном вводе даты и необходимости ее изменения. Проверьте правильность работы программы. Для этого внесите в форму **Карточка сотрудника** для любой записи в поле **Дата приема на работу** такую дату, чтобы возраст сотрудника превышал 35 лет.

Приведенный пример носит демонстрационный характер. Разработка собственных процедур позволяет выполнять сложнейшие операции с базами данных, но требует специальных познаний в языке программирования.

## 7. ИЗМЕНЕНИЕ БАЗЫ ДАННЫХ

Иногда в процессе эксплуатации базы данных необходимо внести незначительные изменения в структуру таблиц для добавления новых полей и расширения хранимой информации. Это в свою очередь приводит к редактированию форм ввода и возможно других связанных объектов базы данных. Данная ситуация является исключительной и должна учитываться на стадии проектирования, но, ввиду изменяющихся требований пользователей, может иметь место.

Предположим нам необходимо добавить к имеющейся базе данных **Кадры** дополнительные поля **Фотография** и **Телефон**. При этом поле **Фотография** будет добавлено в таблицу **Карточка сотрудника**, а поле **Телефон** в таблицу **Адреса сотрудников**.

1. Откройте с помощью контекстного меню в области переходов таблицу **Карточка сотрудника** в режиме **Конструктор**.

2. Добавьте в конец списка имеющихся полей новое поле с именем **Фотография**.

3. В колонке **Тип данных** укажите тип **Вложение**. Как уже упоминалось ранее тип поля **Вложение** позволяет сохранять в базе данных различные файлы рисунков, документов и т.п. Согласно рекомендациям компании Microsoft данный тип поля наиболее подходит для хранения фотографий.

4. Закройте таблицу **Карточка сотрудника** и сохраните сделанные изменения.

5. Следующим шагом откройте таблицу **Адреса сотрудников** в режиме **Конструктор**.

6. Добавьте поле **Телефон**. В колонке **Тип поля** укажите **Текстовый**. В разделе **Свойства поля** задайте для свойства **Размер поля** – 12.


7. Закройте таблицу **Адреса сотрудников** и сохраните сделанные изменения.

Чтобы обеспечить возможность ввода данных во вновь созданные поля необходимо добавить их в существующую форму **Карточка сотрудника**, предварительно обновив источник записей формы.

1. Откройте форму **Карточка сотрудника** в режиме **Конструктор**.

2. Освободите область данных для размещения новых полей, выровняв и распределив имеющиеся поля по левой стороне формы.

3. Щелкните мышкой вне светлой области поля формы. В ленте на вкладке **Конструктор** в группе **Сервис** выполните команду **Страница свойств**. На экране появится **Окно свойств** (рис. 7.1).

4. На вкладке **Данные** в свойстве **Источник записей** нажмите на кнопку мастера  справа от поля.

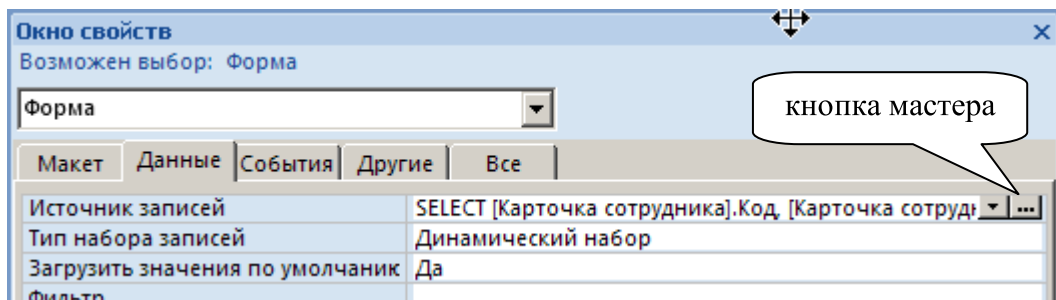


Рис.7.1. Окно свойств формы **Карточка сотрудника**

5. Появится окно **Карточка сотрудника: построитель запросов** (рис. 7.2).

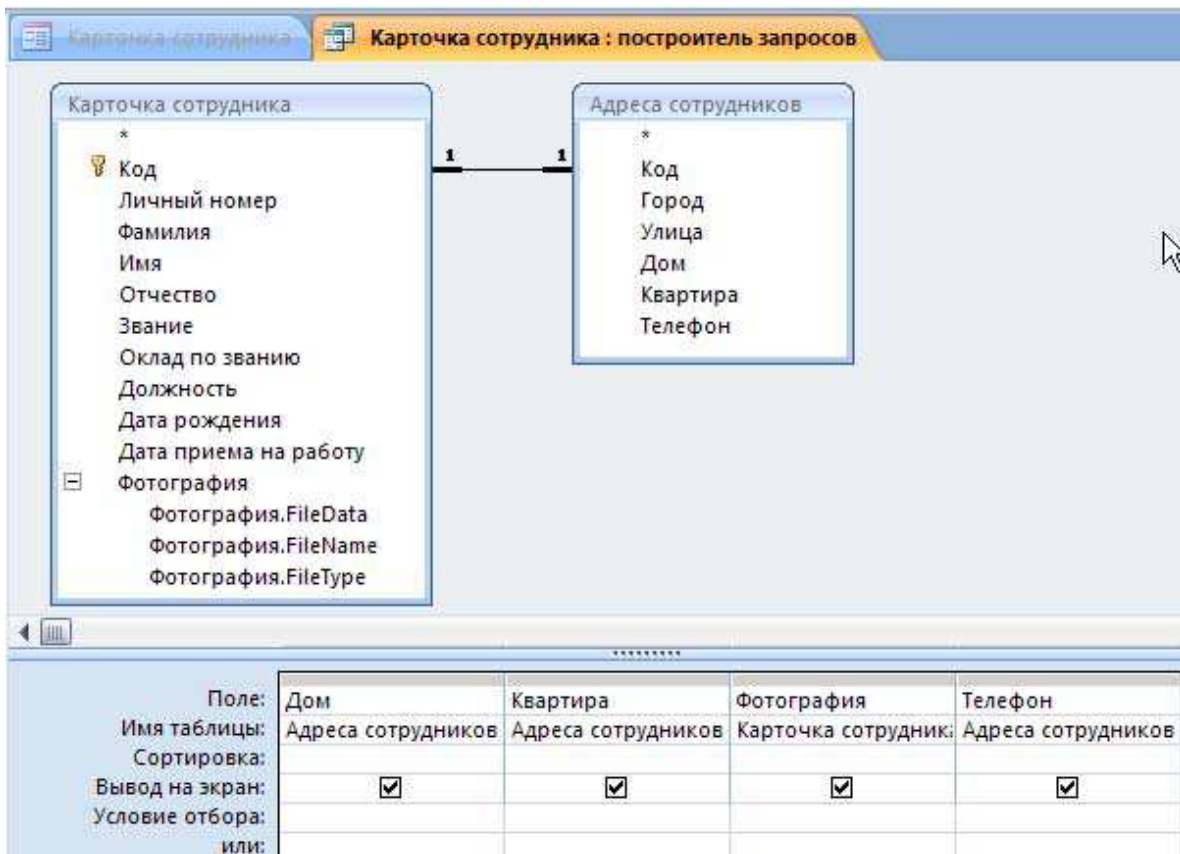


Рис.7.2. Окно построителя запросов

6. Щелкните по полю **Фотография** таблицы **Карточка сотрудника** и, удерживая нажатой левую кнопку мыши, перенесите его в последний пустой столбец конструктора.

7. Прделайте такую же операцию с полем **Телефон** таблицы **Адреса сотрудников**.

8. Закройте построитель запросов, подтвердив сделанные изменения. Закройте **Окно свойств**.

9. В ленте на вкладке **Конструктор** в группе **Сервис** выполните команду **Добавить поля**. На экране появится окно **Список полей** (рис. 7.3).

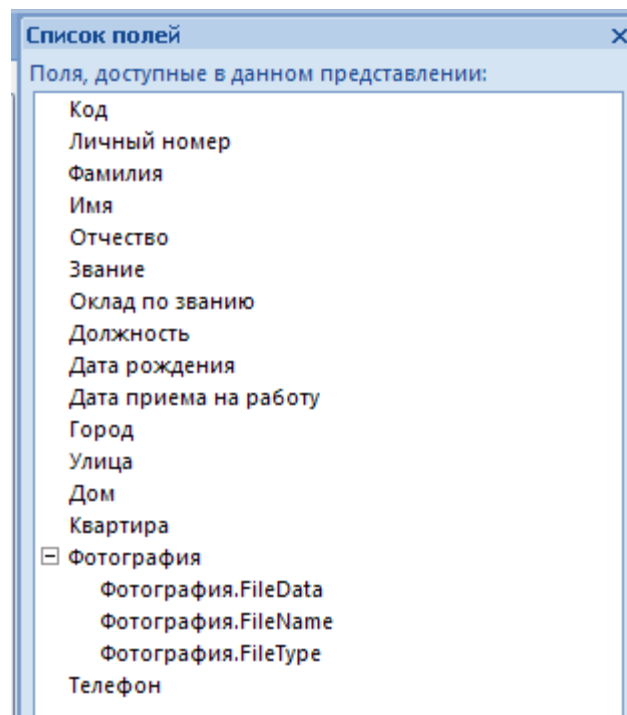


Рис.7.3. Окно **Список полей** формы **Карточка сотрудника**

10. Удерживая нажатой левую кнопку мыши на имени поля **Фотография** перенесите его из списка полей в область данных формы. Прделайте ту же операцию с полем **Телефон**.

11. Измените размеры поля **Фотография** и скомпонуйте имеющиеся поля в форме согласно рисунку 7.4.

12. В ленте на вкладке **Упорядочить** группы **Макет элемента управления** выполните команду **Переходы**.

13. В окне **Последовательность перехода** переместите поле **Телефон** с последнего места перед полем **Фотография**. Подтвердите изменения нажатием кнопки **ОК**.

14. Закройте форму и сохраните результаты.

Карточка сотрудника

## Карточка сотрудника

Код	<input type="text" value="1"/>	Личный номер	<input type="text" value="A-123546"/>	Фотография
Фамилия	<input type="text" value="Иванов"/>	Имя	<input type="text" value="Иван"/>	
		Отчество	<input type="text" value="Иванович"/>	
Звание	<input type="text" value="лейтенант"/>			
Оклад по званию	<input type="text" value="2000"/>			
Должность	<input type="text" value="инспектор"/>			
Дата рождения	<input type="text" value="01.01.1980"/>			
Дата приема на работу	<input type="text" value="01.02.2000"/>	Выслуга	<input type="text" value="10"/>	
Город	<input type="text" value="Красноярск"/>			
Улица	<input type="text" value="Ленина"/>	Дом	<input type="text" value="120"/>	Квартира <input type="text" value="25"/>
Телефон	<input type="text" value="+79831534545"/>			

Рис.7.4. Форма **Карточка сотрудника** после добавления новых полей

15. Откройте форму **Карточка сотрудника** в режиме ввода данных. Заполните для всех записей пустые поля **Телефон** и **Фотография**. Для добавления файла с фотографией используйте контекстное меню поля **Фотография**.

## 8. СОЗДАНИЕ КНОПОЧНОЙ ФОРМЫ

Обычно для придания базе данных законченного вида создают главную кнопочную форму, предоставляющую пользователю прямой доступ к основным задачам приложения: открытию форм, отчетов, выполнению макросов и т.п. **Главная кнопочная форма** может быть простой формой с логотипом, заголовком и несколькими кнопками, открывающими основные объекты приложения. Процесс создания таких форм аналогичен построению форм в режиме конструктора, однако это может оказаться трудоемким делом. Специальная надстройка Microsoft Access – **диспетчер кнопочных форм** поможет вам выполнить эту работу.

Для создания главной формы приложения воспользуемся диспетчером кнопочных форм:

1. Выберите в ленте на вкладке **Работа с базами данных** группы **Работа с базами данных** команду **Диспетчер кнопочных форм**. Диспетчер кнопочных форм сначала проверит, есть ли в базе данных кнопочная форма и таблица Switchboard Items, и в случае их отсутствия выведет окно сообщения (рис. 8.1). В ответ на сообщение нажмите кнопку **Да**.

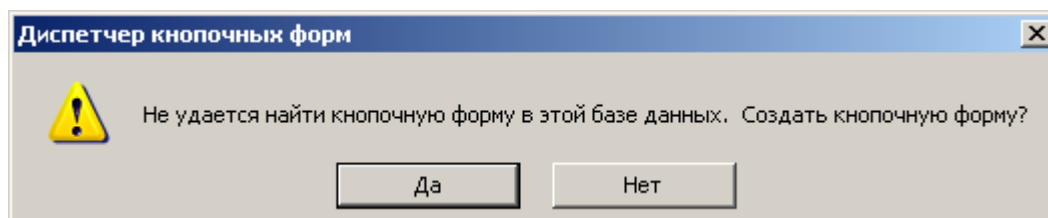


Рис.8.1. Окно сообщения диспетчера кнопочных форм

2. Построив скелет главной кнопочной формы, диспетчер кнопочных форм выведет на экран свое основное окно (рис. 8.2). В этом окне вы имеете возможность определить новые страницы кнопочных форм и указать их элементы. По умолчанию доступна **Главная кнопочная форма**.

3. Для добавления элементов к **Главной кнопочной форме** нажмите кнопку **Изменить** в окне **Диспетчер кнопочных форм**. Становится доступным окно **Изменение страницы кнопочной формы**, позволяющее определить новый элемент формы, отредактировать существующий или изменить его порядок (рис. 8.3).

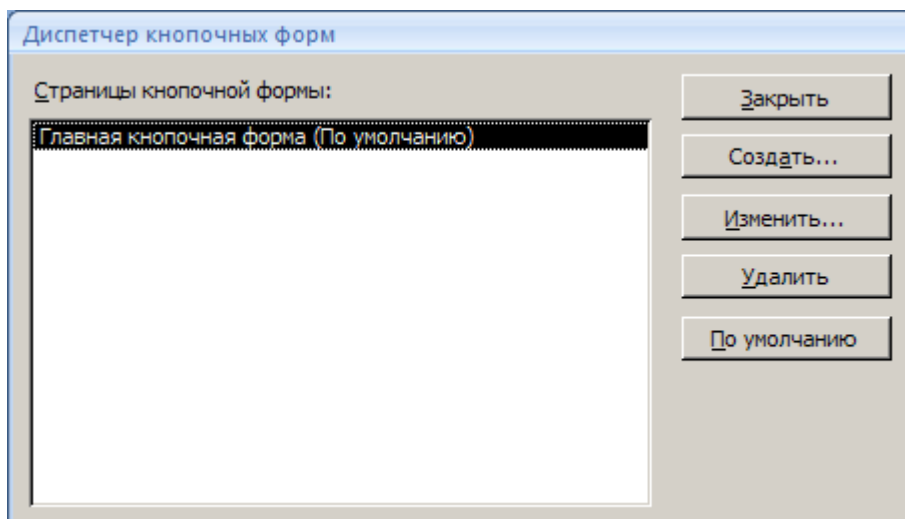


Рис.8.2. Окно Диспетчер кнопочных форм

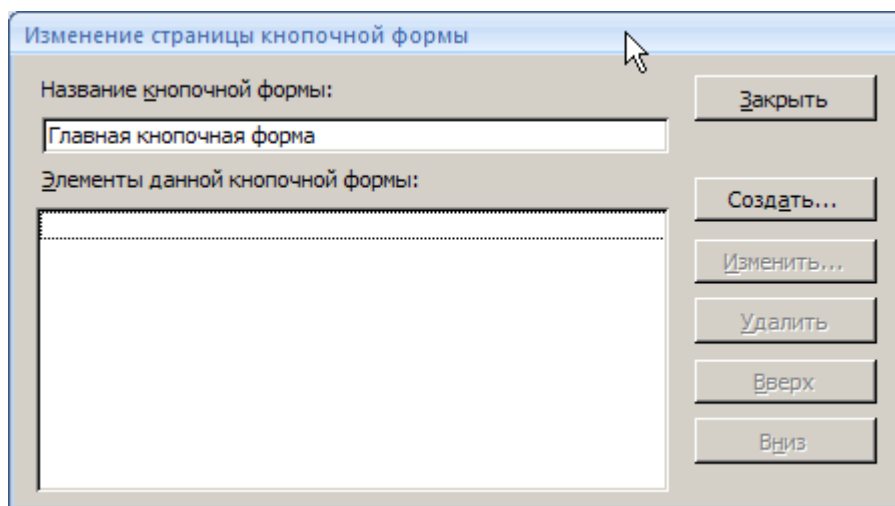


Рис.8.3. Окно Изменение страницы кнопочной формы

4. В окне **Изменение страницы кнопочной формы** нажмите кнопку **Создать**. Введите в поле **Текст:** название элемента **Карточка сотрудника**, в поле **Команда:** укажите из списка **Открыть форму для изменения** и в поле **Форма:** выберите имя формы **Карточка сотрудника** (рис. 8.4). Нажмите кнопку **ОК**.

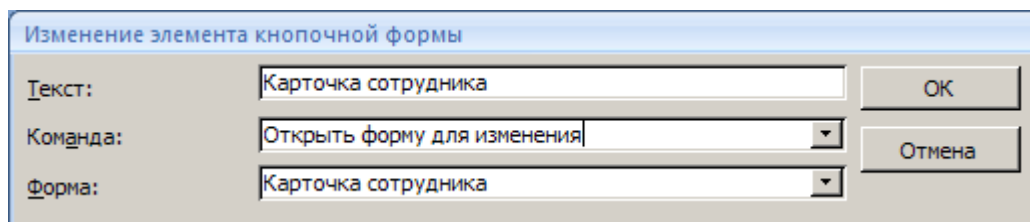


Рис.8.4. Окно Изменение страницы кнопочной формы

5. Аналогичным образом добавьте в **Главную кнопочную форму** элемент с именем **Родственники сотрудника**, указав команду **Открыть форму для изменения**, в поле **Форма: Родственники сотрудника**.

6. Следующим шагом добавьте элементы с именем **Отчет родственники сотрудника** и **Отчет зарплаты сотрудников по подразделениям**. Укажите в поле **Команда: Открыть отчет**.

7. Последним элементом добавьте кнопку **Выход**, указав в поле **Команда: Выйти из приложения**.

8. По завершению всех операций закройте окно диспетчера отчетов. Перейдите на вкладку **Формы** области переходов и откройте форму **Кнопочная форма**. Внешний вид формы представлен на рисунке 8.5.

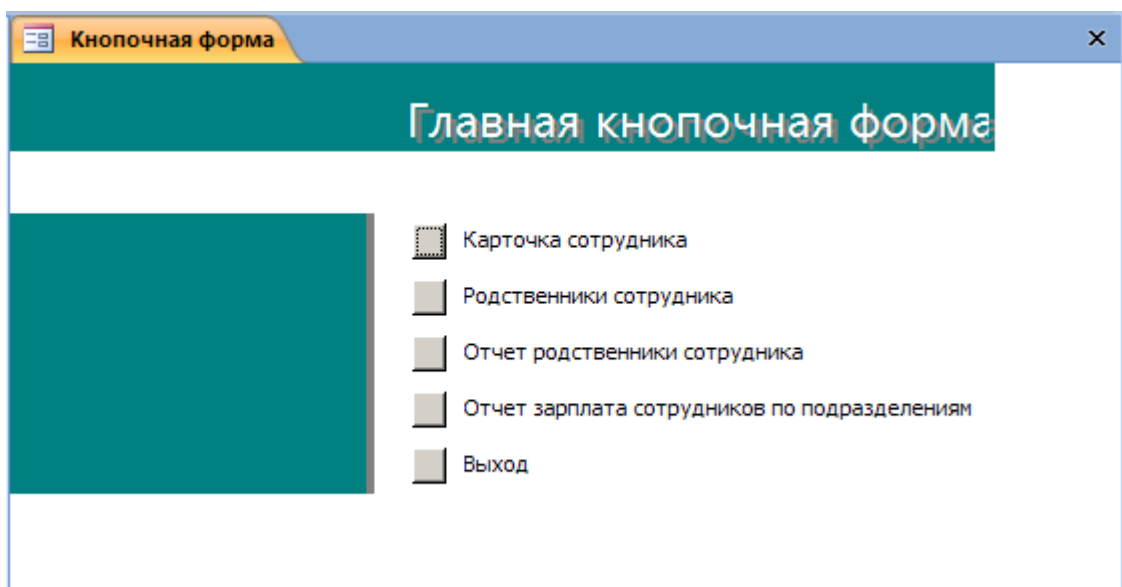



Рис.8.5. Форма **Кнопочная форма**

9. Для того чтобы **Кнопочная форма** загружалась автоматически после запуска файла **Кадры.accdb** нажмите на

кнопку **"Office"**  и выберите команду **Параметры Access**. В окне **Параметры Access** в списке слева выберите опцию **Текущая база данных** и в поле **Форма просмотра** выберите **Кнопочная форма** (рис. 8.6). Нажмите кнопку **ОК**. Теперь при каждом запуске файла базы данных **Кадры** на экране автоматически будет загружаться форма **Кнопочная форма**.

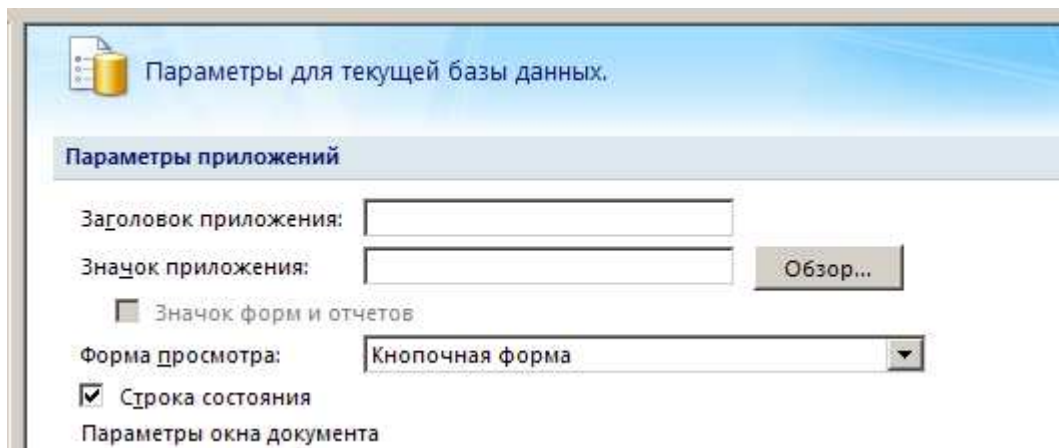


Рис.8.6.Окно **Параметры Access**

## 9. ВЫВОД РЕЗУЛЬТАТОВ

Несмотря на многообразие объектов баз данных, все они печатаются с помощью одних и тех же приемов. Вид печатной копии отчета или формы определяется расположением элементов управления в окне конструктора. Таблицы и запросы вообще практически не имеют параметров форматирования. Ширина и высота ячеек, а также шрифт их содержимого настраиваются непосредственно в режиме просмотра таблицы или запроса с помощью команд вкладки **Формат**. Перед печатью данных полезно вывести отчет в режиме предварительного просмотра, после чего можно распечатывать его на бумагу.

### 9.1. ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР

Перед выводом объекта базы данных на принтер нужно настроить параметры страницы и проверить правильность форматирования в режиме предварительного просмотра. Давайте выполним все эти действия на примере таблицы **Карточка сотрудника**.

1. В области переходов перейдите в группу **Таблицы**.
2. Двойным щелчком на значке **Карточка сотрудника** откройте эту таблицу в режиме редактирования данных.
3. Увеличьте ширину столбцов перемещая мышкой границу между столбцами таким образом, чтобы информация полностью входила в ячейки поля.
4. Выделите столбец **Код**, щелкнув на его заголовке. Нажмите правую кнопку мыши для вызова контекстного меню и выберите в списке команду **Скрыть столбцы**. Информацию служебного столбца нет надобности выводить на печать.
5. Откройте кнопку **"Office"** и выберите команду меню **Печать** ⇒ **Предварительный просмотр**.
6. На вкладке **Предварительный просмотр** группы **Разметка страницы** выполните команду **Параметры страницы**. В появившемся окне на вкладке **Страница** установите альбомную ориентацию; в разделе **Поля** настройте поля по своему усмотрению. Нажмите кнопку **ОК**.
7. Примерный вид печатного варианта таблицы **Карточка сотрудника** представлен на рисунке 9.1.

The screenshot shows a window titled 'Карточка сотрудника' with a table containing employee information. The table has 9 columns: 'Личный номер', 'Фамилия', 'Имя', 'Отчество', 'Звание', 'Оклад по', 'Должность', 'Дата рождения', and 'Дата приема на рабо'. There are four rows of data.

Личный номер	Фамилия	Имя	Отчество	Звание	Оклад по	Должность	Дата рождения	Дата приема на рабо
A-123546	Иванов	Иван	Иванович	лейтенант	2000	инспектор	01.01.1980	01.02.2000
B-289487	Сидоров	Михаил	Юрьевич	капитан	3000	оперуполномоченный	05.07.1978	07.03.1991
и-747474	Карамазов	Владимир	Андреевич	сержант	2300	водитель	20.12.1980	02.09.2002
B-888222	Петров	Сергей	Александрович	майор	3200	следователь	28.10.1977	20.08.1992

Рис.9.1. Окно предварительного просмотра печати

Режим предварительного просмотра является основным вариантом демонстрации отчетов, поэтому он активизируется двойным щелчком на значке отчета в окне базы данных.

## 9.2. ФАЙЛОВЫЕ ОПЕРАЦИИ

Чтобы поддерживать базу данных в надлежащем порядке, необходимо периодически выполнять некоторые служебные процедуры. Вероятно, наиболее важной из них является резервное копирование данных, которое выполняется с помощью стандартных средств Windows для работы с файлами или средствами самого Microsoft Office Access. Не менее существенной процедурой является сжатие базы данных. По мере обработки таблиц и добавления объектов файл базы данных увеличивается в объеме. При удалении части информации не происходит полного ее удаления из файла. В результате объем, занимаемый базой данных на жестком диске, может оказаться неоправданно большим. Кроме этого выполняется процедура исправления ошибок и восстановление.

Чтобы сжать базу данных нажмите на кнопку **"Office"** и в списке меню выполните команду **Управление ⇒ Сжать и восстановить базу данных**. После этой операции база данных будет автоматически перегружена.

Сравните размер файла базы данных до и после сжатия. Вы увидите, насколько эффективно сжатие экономит дисковое пространство.

Команда **Управление ⇒ Сжать и восстановить базу данных** также выполняет восстановление базы данных, поврежденной в результате внезапного сбоя электропитания. Если в процессе работы вы заметили, что база данных ведет себя как-то странно, не дожидаясь предупреждения, выполните команду кнопки **"Office" ⇒ Управление ⇒ Сжать и восстано-**

**вить базу данных.** Microsoft Office Access по возможности восстановит нарушения, обнаруженные в структуре данных.

Если база данных содержит конфиденциальные сведения, воспользуйтесь возможностями Microsoft Access по ограничению доступа и шифрованию данных. Чтобы задать пароль на открытие базы данных, выполните следующие шаги:

1. Закройте базу данных кнопкой **"Office"** ⇒ **Закрывать базу данных.**

2. Щелкните на кнопке **Другие...** списка **Открыть последнюю базу данных.**

3. В окне открытия файла выделите значок вашей базы данных **Кадры.**

4. В списке поля **Открыть** выберите **Монопольно.**

5. В ленте на вкладке **Работа с базами данных** в группе **Работа с базами данных** выберите команду **Задать пароль базы данных.**

6. В поле **Пароль** открывшегося окна диалога введите пароль.

7. Для проверки повторно введите тот же пароль в поле **Подтверждение.**

8. Щелкните на кнопке **ОК.** Теперь для следующего открытия базы данных потребуется вводить пароль.

9. Чтобы удалить пароль для доступа к базе данных в ленте на вкладке **Работа с базами данных** в группе **Работа с базами данных** выберите команду **Удалить пароль базы данных,** введите старый пароль и щелкните на кнопке **ОК.**

## 10. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

### Вариант 1

1. Создайте таблицу, содержащую следующие данные:

Код товара	Наименование программного продукта	Цена, \$	Наличие на складе	Фирма
001	1С. Анализ финансового состояния	160	склад	ЦИНТ
002	Аналитик. Промышленность 4.2	1100	склад	Оптимальный вариант
003	Галактика. Анализ деятельности	424,8	2 недели	ЭТИК
004	1С. Бухгалтерия. Проф. for Windows XP	700	склад	«Интанк» Красноярск
005	БЭСТ - 4	600	склад	Оптимальный вариант

В качестве ключевого поля выбрать Код товара.

2. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания. Включите в нее цену в рублях (курс доллара установите самостоятельно).

3. Выполните запросы:

- найти все программные продукты, начинающиеся на 1С;
- найти все программные продукты с ценой от \$500 до \$1000.

Запросы сохраните под именем «Запрос 1» и «Запрос 2».

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет вставьте столбец расчета рублевых цен.

4.1. Подведите итоги по каждому продукту и общий итог по всем продуктам в целом.

### Вариант 2

1. Создайте таблицу, содержащую следующие данные:

Код товара	Модель	Год выпуска	Пробег, тыс. км.	Цена, тыс. руб.	Автосалон
003	ВАЗ - 2107	1994	83	22	КРЕПОСТЬ
004	ВАЗ - 21093	1997	0	48	ЗАПАДНЫЙ

005	TOYOTA - CALDINA	1994	80	67	ЗАПАДНЫЙ
001	ВАЗ - 2101	1972	200	7	КРАСНОДАРСКИЙ
002	ВАЗ - 21011	1981	115	11	КРАСНОДАРСКИЙ

В качестве ключевого поля выбрать Код товара.

2. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания. Подсчитайте размер комиссионных автосалону (12% или 5% от цены).

3. Выполните запросы:

– найти все модели, начинающиеся с ВАЗ - 2101;  
– найти все автомобили с ценой от 20 до 50 тыс. рублей включительно.

Запросы сохраните под именем «Запрос 1» и «Запрос 2».

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет вставьте столбец расчета комиссионных.

4.1. Итоги подводить по каждому салону и общий итог по всем салонам.

### Вариант 3

1. Создайте таблицу, содержащую следующие данные:

Код вклада	Вид вклада	Длительность (дней)	Сумма вклада, тыс. руб.	Период начисления %	Годовая % ставка	Банк
001	Срочный	31	10	в конце срока	14	КЕДР
002	Срочный	31	5	в конце срока	18	РАДУГА
003	Срочный	91	10	в конце срока	21	КЕДР
004	Срочный	95	5	в конце срока	24	РАДУГА
005	Новый рубль	31	10	в конце срока	18	КЕДР

В качестве ключевого поля выбрать Код вклада.

2. Выполните запросы:

– найти все срочные вклады на 31 день;  
– найти все вклады от 18% до 25% годовых.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания. Рассчитайте процентную ставку за период по формуле:  $\text{ГОДОВАЯ ПРОЦЕНТНАЯ СТАВКА} * \text{ДЛИТЕЛЬНОСТЬ ПЕРИОДА В ДНЯХ} / 365$ . Точность вычисления – 2 знака после запятой.

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет включите столбец расчета процентной ставки за период.

4.1. Итоги подводить для каждого банка и для всех банков.

#### Вариант 4

1. Создайте таблицу, содержащую следующие данные:

№ счета	Дата	Наименование товара	Цена, руб.	Продавец
100	5.5.2010	Электрочайник	210	Иванова
101	5.5.2010	Электроутюг	200	Петрова
102	5.5.2010	Электрокофеварка	300	Иванова
107	10.5.2010	Кухонный процессор	1000	Смирнова
115	20.5.2010	Набор ножей	35	Петрова

В качестве ключевого поля выбрать № счета.

2. Выполните запросы:

- найти все товары, начинающиеся со слова Электро;
- найти все товары, ценой от 300 до 1000 руб. включительно.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Подсчитайте размер комиссионных продавца (8% от цены товара).

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет вставьте столбец расчета размера комиссионных продавца.

4.1. Итоги подводить для каждого продавца и для всего отдела в целом.

## Вариант 5

1. Создайте таблицу, содержащую следующие данные:

№ накладной	Фирма	Дата	Цена, руб.	Накладные расходы, %
1	КРОКУС	25.10.2010	175	15
2	ФИКУС	25.10.2010	295	12,5
3	Светотехника	30.10.2010	324	15
4	Комплимент	30.10.2010	75	12,5
5	КРОКУС	2.11.2010	100	10

В качестве ключевого поля выбрать № накладной.

2. Выполните запросы:

– найти все товары, проданные фирмой КРОКУС;

– найти все товары с ценой от 100 до 300 рублей включительно.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Подсчитайте итоговую цену товаров с учетом накладных расходов.

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет вставьте столбец расчета итоговой цены товара, с учетом накладных расходов.

4.1. Итоги подводить по каждой фирме и для всех фирм в целом.

## Вариант 6

1. Создайте таблицу, содержащую следующие данные:

Индекс комплектующей	Наименование комплектующего	Количество, штук	Стоимость единицы, руб.
122	А - 234	4	200
123	Ф - 235	5	150
706	С - 200	1	600
504	К - 101	2	100
321	Н - 203	1	1000

В качестве ключевого поля выбрать Индекс комплектующей.

2. Выполните запросы:

– найти все комплектующие с литерой «А» в наименовании;

– найти все комплектующие, входящие в изделие в одном экземпляре.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Подсчитайте стоимость комплектующих каждого наименования, входящих в изделие.

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет включить столбец расчета стоимости комплектующего каждого наименования.

4.1. Итоги подводить по каждому комплектующему и по всем комплектующим в целом.

### Вариант 7

1. Создайте таблицу, содержащую следующие данные:

№ клиента	Фамилия клиента	Наименование работы	Дата окончания	Стоимость, руб.
1	Иванов	чистка костюма	5.10.2010	250
2	Петров	чистка платья	7.10.2010	100
3	Сидорова	чистка пальто	3.10.2010	300
4	Александрова	чистка скатерти	2.10.2010	50
5	Иванов	чистка костюма	10.10.2010	250

В качестве ключевого поля выбрать № клиента.

2. Выполните запросы:

– найти все записи с работой «чистка костюма»;  
– найти все записи со стоимостью от 100 до 250 рублей включительно.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Подсчитайте зарплату в каждом заказе (25 % от стоимости заказа).

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В от-

чет включите столбец определения величины зарплаты по каждому заказу.

4.1. Итоги подводить по видам работы и по всем работам.

### Вариант 8

1. Создайте таблицу, содержащую следующие данные:

Код группы	Номенклатурный номер	Наименование товара	Количество	Цена за единицу, руб.
01	К - 001	Гарнитур кухонный	100	2000
02	Пл - 004	Ковровое покрытие «Лес»	800	37,5
03	М - 001	Набор мягкой мебели	20	1647,5
04	Пл - 003	Ковровое покрытие	380	34,31
05	О - 001	Обивочная ткань	90	29,97

В качестве ключевого поля выбрать Код группы или Номенклатурный номер.

2. Выполните запросы:

– найти все Ковровые покрытия;

– найти все товары, ценой свыше 1 тыс. руб. за шт.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Подсчитайте стоимость товаров каждого вида.

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет введите столбец расчета стоимости товаров.

4.1. Итоги подводить по каждой группе товаров и по всем товарам в целом.

### Вариант 9

1. Создайте таблицу, содержащую следующие данные:

Дата операции	№ документа	Код операции	От кого получено или кому отпущено	Приход	Расход
10.10.2010	0001	03	ООО «АГРО»	30	
10.10.2010	0002	01	Комиссионный	20	15
11.10.2010	0010	01	Универмаг №1	25	15
11.10.2010	0011	03	ООО «АГРО»	10	
15.10.2010	0020	03	ООО «Факел»	20	

В качестве ключевого поля выбрать № документа.

2. Выполните запросы:

– найти все записи по расходу товаров;

– найти все записи в интервале времени операций с 10.10.2010 по 11.10.2010 включительно.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания .

3.1. Рассчитайте остаток товаров для каждой записи по формуле  $ОСТАТОК = ПРИХОД - РАСХОД$  (считать начальный остаток равным нулю).

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет включите столбец расчета остатков товара на складе .

4.1. Итоги подводить по каждой дате и по всем датам в целом.

### Вариант 10

1. Создайте таблицу, содержащую следующие данные:

Группа товаров	Наименование товара	Единица измерения	Цена за единицу, руб.	Количество, шт.
005	Коврик прикроватный	шт.	37,50	30
003	Ковровое покрытие «Лес»	метр	10,00	800
001	Гарнитур кухонный	комплект	1647,70	93
004	Набор мягкой мебели	комплект	2500,00	20
002	Обивочная ткань	метр	29,97	90

В качестве ключевого поля выбрать Группа товаров.

2. Выполните запросы:

– найти все записи с мебелью;

– найти все записи с ценой свыше 1000 руб.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания .

3.1. Подсчитайте сумму по каждому товару.

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обя-

зательно включить № группы, ФИО полностью, дату создания. Вставьте столбец расчета сумм по каждому товару.

4.1. Итоги подводить по группам товаров и общие итоги в целом.

### Вариант 11

1. Создайте таблицу, содержащую следующие данные:

Код группы	Наименование группы	Первоначальная стоимость, руб.	Сумма износа
201	Гидротехнические сооружения	1000000,00	10000,00
301	Трубопроводы	100000,00	5000,00
480	Вычислительная техника	30000,00	5000,00
490	Спортивное оборудование	50000,00	5000,00
550	Инвентарь	75000,00	5000,00

В качестве ключевого поля выбрать Код группы.

2. Выполните запросы:

- найти запись по вычислительной технике;
- найти все записи с первоначальной стоимостью от 50 до 500 тысяч включительно.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные данные. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Подсчитайте остаточную стоимость по каждой группе по формуле: **ОСТАТОЧНАЯ СТОИМОСТЬ = ПЕРВОНАЧАЛЬНАЯ СТОИМОСТЬ - СУММА ИЗНОСА.**

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. Вставьте столбец расчета остаточной стоимости.

4.1. Итоги подводить по первой цифре группы и по всем группам.

### Вариант 12

1. Создайте таблицу, содержащую следующие данные:

Код товара	Наименование программного продукта	Тип программного продукта	Цена, \$ США	Наличие на складе	Фирма-поставщик
001	Access 2007 RUS	Базы данных	110,00	склад	СТЕК

002	STYLUS for Windows XP	Переводчик	290,00	1 неделя	Максофт
003	Консультант плюс. Версия проф.	Законодательство	950,00	склад	Искра
004	Консультант плюс. Российское законодательство	Законодательство	450,00	склад	Искра
005	Access for Windows XP	Базы данных	193,00	1 неделя	СТЕК

В качестве ключевого поля выбрать Код товара.

2. Выполните запросы:

– найти все программные продукты, посвященные законодательству;

– найти все программные продукты, которые можно приобрести немедленно.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Для каждого программного продукта рассчитайте цену в рублях. Коэффициент конвертации задайте самостоятельно.

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет включить столбец расчета рублевых цен.

4.1. Итоги подводить по каждому типу программных средств и общий итог.

### Вариант 13

1. Создайте таблицу, содержащую следующие данные:

Код товара	Наименование товара	Тип товара	Цена, \$	Наличие на складе	Фирма-поставщик
001	CITIZEN SB-745P	Бизнес-папка	11,80	склад	АЛВИ
002	AURORA B85A	Калькулятор	27,50	склад	ЭТиК
003	CITIZEN SDC-8780	Калькулятор	20,80	склад	АЛВИ
004	CASIO LS-700	Калькулятор карманный	3,60	склад	АЛВИ
005	CITIZEN RX-4110	Записная книжка	35,50	склад	СТИЛЬ

В качестве ключевого поля выбрать Код товара.

2. Выполните запросы:

– найти все модели, выпускаемые фирмой CITIZEN;

– найти все товары, с ценами от 10 до 30 долларов.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Рассчитайте рублевые цены на каждый вид товара. Коэффициент конвертации ввести самостоятельно.

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. Вставьте столбец расчета рублевых цен на товары.

4.1. Итоги подводить по каждому типу товаров.

#### Вариант 14

1. Создайте таблицу, содержащую следующие данные:

Табельный номер	ФИО	Должность	Разряд	Часовая тарифная ставка, руб.	Отработано, часов
101	Михайленко	повар	6	60,82	175
123	Зинченко	повар	5	50,96	175
183	Сидоренко	повар	5	50,96	150
105	Капроненко	повар	5	50,96	80
134	Сычева	повар	4	50,39	120

В качестве ключевого поля выбрать табельный номер.

2. Выполните запросы:

– найти всех поваров пятого разряда;

– найти всех поваров, у которых часовая тарифная ставка находится в интервале от 50,5 до 60 рублей включительно.

Запросы сохранить под именем "Запрос 1" и "Запрос 2".

3. Создайте форму, содержащую все исходные данные. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Рассчитайте заработную плату за отработанное время.

4. Разработайте отчет в табличном виде, используя вид отчета "Группировка данных и вычисление итогов" В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В от-

чет включите столбец расчета заработной платы за отработанное время

4.1. Итоги подводить по каждому разряду и в целом по всей бригаде.

### Вариант 15

1. Создайте таблицу, содержащую следующие данные:

№ выдачи дела	№ личного дела	Дата выдачи	Кому выдано	№ телефона	Дата возврата
2	А - 101	10.10.2010	Иванов	227-75-24	17.10.2010
4	Б - 203	10.10.2010	Петров	227-75-18	10.10.2010
1	С - 407	12.10.2010	Сидоров	221-21-21	15.10.2010
3	В - 103	12.10.2010	Смирнов	221-75-18	15.10.2010
10	А - 121	13.10.2010	Иванов	227-75-24	15.10.2010

В качестве ключевого поля выбрать № выдачи дела.

2. Выполните запросы:

- найти все личные дела запрошенные 12.10.2010;
- найти все личные дела с датами возврата от 10.12.2010 до 15.10.2010.

Запросы сохранить под именем «Запрос 1» и «Запрос 2».

3. Создайте форму, содержащую все исходные поля. В заголовок формы обязательно включить № группы, ФИО полностью, дату создания.

3.1. Подсчитайте ДЛИТЕЛЬНОСТЬ РАБОТЫ с личными делами по формуле: ДАТА ВОЗВРАТА - ДАТА ВЫДАЧИ + 1

4. Разработайте отчет в табличном виде, используя вид отчета «Группировка данных и вычисление итогов». В заголовок отчета обязательно включить № группы, ФИО полностью, дату создания. В отчет включите столбец расчета длительности работы с личными делами.

4.1. Итоги подводить по каждой дате и по всем датам в целом.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Белянин, М.В. Создание баз данных в Access 2007. Ваша первая база данных в Access 2007 / М.В. Белянин. – Издательство: НТ Пресс, 2008. – 281 с. ISBN: 9785477012275.
2. Введение в системы баз данных / К. Дж. Дейт. – Издательство: Вильямс, 2006. – 328 с. ISBN 5-8459-0788-8, 0-321-19784-4.
3. Днепров, А. Видеосамоучитель. Microsoft Access 2007. / А. Днепров. – СПб.: Питер, 2008. – 240 с. ISBN 978-5-388-00139-9.
4. Информатика и математика для юристов : учебник для вузов: рекомендовано Министерством образования РФ / под ред. С.Я. Казанцева, Н.М. Дубинина. – ЮНИТИ-ДАНА, 2010. – 560 с. ISBN: 5-238-00928-3, 978-5-238-00928-5.
5. Сергеев, А.П. Microsoft Office 2007. Самоучитель / А.П. Сергеев. – Издательство: Диалектика, 2008. – 416 с. ISBN 978-5-8459-1226-8.
6. Тимошок, Т.В. Microsoft Office Access 2007. Самоучитель. / Т.В. Тимошок. – Издательство: Диалектика, 2008. – 464 с. ISBN 978-5-8459-1341-8.
7. Microsoft Office Access 2007 для «чайников» / Лори Ульрих Фуллер, Кен Кук, Джон Кауфельд. – Издательство: Вильямс, 2007. – 384 с. ISBN 978-5-8459-1239-8, 0-470-04612-0.

**Сергей Николаевич Ефимов  
Вячеслав Витальевич Молоков  
Александра Сергеевна Шерстяных**

## **Основы работы с базами данных Access 2007**

Учебное пособие

Печатается в авторской редакции

Технический редактор М.Н.Соколова

Подписано в печать 20 января 2011 г.  
Формат Р 60x84. Бумага типографская. Гарнитура Таймс.  
Печать офсетная. Уч.изд.л. 2,8 (5,75 усл.п.л.).  
Тираж 60 экз. Заказ \_\_\_\_\_.

Организационно-научный и редакционно-издательский отдел.  
Сибирский юридический институт МВД России.  
660131, г. Красноярск, ул. Рокоссовского, 20.

---

Отпечатано на участке оперативной полиграфии  
Сибирского юридического института МВД России.  
660050, г. Красноярск, ул. Кутузова, 6.