

**Министерство образования и науки Российской Федерации
Федеральное агентство по образованию**

**Санкт-Петербургский государственный университет
информационных технологий, механики и оптики**

А.А. Бобцов, В.В. Шиегин

**Банки и базы данных.
Основы работы с MS Access.
Часть 1 (для пользователей)**

Учебное пособие

Санкт-Петербург

2005

УДК 681.3

Бобцов А.А., Шиегин В.В. Банки и базы данных. Основы работы с MS Access. Часть 1 (для пользователей). Учебное пособие. – СПб., 2005. - 93 с.

Рецензенты: Л.С. Лисицына, к.т.н., доцент, зав. каф. КОТ СПбГУ ИТМО
А.В. Лямин, к.т.н., доцент директор ЦДО СПбГУ ИТМО

Учебное пособие предназначено для дисциплины ОПД.Ф.14 “Банки и базы данных” для специальности 073700 – «Информационные технологии в образовании», а также для курсов повышения квалификации работников образования по программам соответствующего содержания.

Учебное пособие позволит читателю получить представление о банках и базах данных, а также научиться самостоятельно работать и разрабатывать базы данных на основе офисного пакета MS Access.

Печатается по решению УМС факультета ИТиП СПбГУ ИТМО

© Санкт-Петербургский государственный университет информационных технологий механики и оптики, 2005

© Кафедра компьютерных образовательных технологий, 2005

© Бобцов А.А., Шиегин В.В., 2005

Содержание

Введение	4
1. Введение в системы управления базами данных	5
1.1. Данные	5
1.2. Базы данных.....	6
1.3. Модели представления данных.....	6
1.4. Реляционные базы данных	8
1.5. Системы управления базами данных	10
2. СУБД Microsoft Access 2000. Знакомство	12
2.1. Инструменты для работы с данными в Access	12
2.2. Работа с готовой базой данных	16
2.3. Интеграция с приложениями MS Office.....	27
3. Разработка базы данных.....	30
3.1. Таблицы и связи между ними.....	30
3.2. Запросы	38
3.3. Формы и отчеты.....	55
Приложение	70
Задание 1. Работа с готовой базой данных. Приемы работы	70
Задание 2. Создание и редактирование таблиц.	
Работа со схемой данных	73
Задание 3. Создание и редактирование запросов	78
Задание 4. Интеграция Access с приложениями MS Office.	
Экспорт и импорт данных	82
Задание 5. Создание и редактирование форм и отчетов	85
Литература.....	91

Введение

Цель данного учебного пособия – получить представление о банках и базах данных, а также научиться самостоятельно работать и разрабатывать базы данных на основе офисного пакета MS Access. За время обучения будут рассмотрены следующие темы:

- какие модели представления данных известны;
- что представляют собой и для чего предназначены реляционные базы данных;
- какие задачи выполняют системы управления базами данных (СУБД);
- какие разновидности СУБД существуют;
- для решения каких задач предназначена СУБД MS Access, какие инструменты для работы с данными она предоставляет и какого рода данные способна хранить;
 - как работать с базой данных в среде СУБД MS Access, как создавать и редактировать структуру таблиц базы данных и связей между ними;
 - как выполнять экспорт и импорт данных.

Для освоения материала, изложенного в предлагаемом пособии, необходимы знания персонального компьютера, операционной системы *Windows* и ее приложений. Если же у вас по ряду причин возникнут трудности с освоением материала, мы советуем вам посмотреть некоторые из книг, входящих в список литературы.

1. Введение в системы управления базами данных

Для понимания термина «Система Управления Базами Данных» (СУБД) попробуем выяснить сначала:

1. Что такое данные?
2. Что такое базы данных?

1.1. Данные

Похоже, всякая попытка дать четкое определение термина «данные» не лишена тавтологичности. Рассмотрим несколько таких определений:

Данные – это факты, характеризующие свойства объектов и явлений из некоторой предметной области. Но что такое факты, как не разнообразные данные?

Данные – это информация, предназначенная к обработке. А информация, в свою очередь – это знания о фактических данных.

Данные – это сведения, полученные путем измерения, наблюдения и т.п. Не вдаваясь в дальнейшие семантические изыскания, примем за основу: **данные** – это **информация, факты и сведения**. Теперь разберемся, что представляют собой данные с практической точки зрения.

Посмотрим по сторонам: предположим, что мы видим три книжных полки. В первой – художественные произведения, во второй – техническая литература, в третьей спит кошка. Что можно сказать об увиденном, где здесь данные? Жанр, наименование, и имена авторов книг, местоположение каждой книги (на первой или второй полке, или вообще вне полок), общее их количество. Стоимость каждой книги, количество страниц. Материал, из которого изготовлена каждая полка, ее габариты. Порода, возраст, вес, имя кошки. Марка ее любимого корма, способность к ловле мышей. На какой из полок предпочитает отдыхать. Данные можно найти везде, они вокруг нас, но не все они представляют интерес с точки зрения их сбора, хранения и обработки. Итак, множество данных описывает свойства каких-либо объектов, принадлежащих выбранной предметной области и взаимоотношения между этими объектами. Для каждого объекта можно определить список его свойств, которые являются существенными. Например, объект «налогоплательщик» с точки зрения налоговых служб может обладать следующими свойствами: имя, дата рождения, доход за предыдущий год и т.п. Конкретный человек, к которому эти данные относятся, имеет гораздо больше свойств – цвет глаз, размер обуви, любимый вид спорта и т.д., но вряд ли они представляют интерес для решения задачи сбора налогов. Тому же самому человеку с точки зрения стоматолога соответствует совершенно иной перечень свойств. Каждое свойство конкретного объекта имеет некоторое значение (цвет глаз – голубой, вес – 78кг, доход – неизвестен). Набор значений свойств формирует некоторую **абстракцию**, упрощение этого объекта – мы манипулируем не самим объектом, а ограниченным набором представлений о нем.

1.2. Базы данных

База данных (БД) – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными.

Данные, относящиеся к некоторой предметной области, объединяются и структурируются таким образом, чтобы над ними можно было выполнять необходимые операции: обновлять, добавлять новые данные, удалять ненужные, извлекать данные, отвечающие условиям поиска. Простейший пример базы данных – записная книжка с номерами телефонов. Данные упорядочены по первым буквам имен абонентов, что позволяет быстро найти нужный телефон. При необходимости можно добавить информацию о новом абоненте, вычеркнуть или изменить записанную ранее. Другой пример – база данных торговой организации. БД содержит сведения о клиентах и товарах. Объект «клиент» вступает во взаимоотношение с объектом «товар», оформляя заказ. Данные должны быть организованы так, чтобы допускать их обновление, добавление сведений о новых клиентах, поступивших товарах, оформление заказов, а также извлечение данных для формирования различных отчетов – поступившие товары, заказы за последний месяц и т.п.

Банк данных – база данных и система управления ею.

1.3. Модели представления данных

Модель – способ структурирования данных, описания взаимосвязей между данными.

Очевидные требования к модели:

- Модель должна быть достаточно универсальной, позволяя работать с данными различной структуры и сложности.
- Модель должна допускать автоматическую обработку данных, т.е. должна быть реализуема программными средствами.
- Модель должна быть наглядной, «прозрачной». Поскольку задача описания структуры данных средствами выбранной модели возлагается на разработчика (человека), чем сложнее модель – тем труднее избежать ошибок при проектировании.

Ниже перечислены основные разновидности моделей представления данных, используемых или использовавшихся в прошлом.

Иерархическая модель. Модель (как видно из названия) представляет данные в виде иерархии (рис. 1.1). Модель ориентирована на описание объектов, находящихся между собой в отношении подчинения. Например, структура кадров некоторой организации. Организация состоит из отделов, каждый отдел имеет руково-

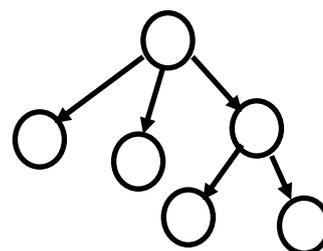


Рис. 1.1. Иерархическая модель

дителя и сотрудников. Другой пример: объект «колесо» является составной частью объекта «автомобиль». Между автомобилем и колесом имеется связь, смысл которой можно озвучить следующим образом: объект «автомобиль» включает в себя несколько объектов «колесо».

Сетевая модель. Сетевая модель (рис. 1.2) представляет собой развитие иерархической. Модель позволяет описывать более сложные виды взаимоотношений между данными. Однако расширение возможностей достигается за счет большей сложности реализации самой модели и трудности манипулирования данными.

Реляционная модель. В реляционной модели (рис. 1.3) данные представляются в виде таблиц, состоящих из строк и столбцов. Каждая строка таблицы – информация об одном конкретном объекте, столбцы содержат свойства этого объекта. Взаимоотношения между объектами задаются с помощью связей между столбцами таблиц. Реляционная модель на сегодняшний день наиболее распространена. Она достаточно универсальна и проста в проектировании.

Ниже будут более подробно рассмотрены принципы построения баз данных, основанных на реляционной модели.

Объектная модель. В этой модели данные представляются в форме объектов. Объект имеет набор свойств, называемых атрибутами, и может включать в себя также процедуры для обработки данных, которые называют методами. Объекты, имеющие одинаковые наборы атрибутов и различающиеся только их значениями, образуют некоторый класс объектов. Например, класс «клиент» может иметь следующие атрибуты: «фамилия», «имя», «отчество», «номер кредитной карты». Для каждого объекта из этого класса определены конкретные значения перечисленных атрибутов. Говорят, что объект является экземпляром класса. На основе существующего класса могут создаваться новые, наследующие свойства исходного. При этом исходный класс именуется родителем нового класса. Производный класс называют потомком исходного. При этом объекты – экземпляры класса-потомка принадлежат также и родительскому классу, поскольку обладают всеми его атрибутами. Пример: на основе класса «клиент» может быть определен класс «постоянный клиент». Новый класс наследует атрибуты родительского и при этом добавляется новый атрибут – «процент скидки». Все постоянные клиенты продолжают оставаться клиентами в прежнем понимании, но о них имеется еще и дополнительная информация. В настоящее время не существует единого подхода к реализации объектных баз данных. Объектный подход – набор общих принципов, которые могут применяться при проектировании различных приложений.

Гибридные модели. В некоторых приложениях предпринимаются попытки смешения различных моделей представления данных. Пример такого смешения

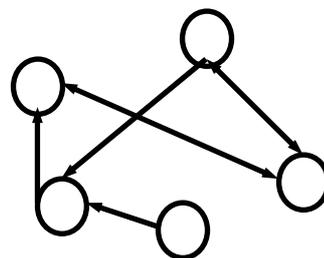


Рис. 1.2. Сетевая модель

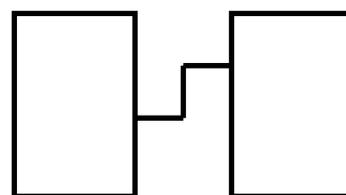


Рис. 1.3. Реляционная модель

– объектно-реляционная модель. В ней использовано некоторое сходство между реляционной и объектной идеологией. Строки таблиц реляционной модели соответствуют объектам объектной модели, столбцы таблиц – атрибутам объектов. Таблицы в целом являются аналогом классов. Отсюда вытекает возможность введения наследования при определении таблиц – таблица-потомок содержит те же столбцы, что и родительская, и, кроме того – дополнительные, определенные при наследовании. По идее создателей, объектно-реляционная модель должна унаследовать от реляционной легкость описания и манипулирования данными, а от объектной – возможность определения более сложных взаимоотношений между объектами.

1.4. Реляционные базы данных

Базы данных, хранение информации в которых основано на реляционной модели, называют **реляционными базами данных**. Как было сказано ранее, реляционная модель предполагает организацию данных в виде таблиц. Строки таблиц называют **записями**, столбцы – **полями**.



Рис. 1.4. Таблицы в реляционной модели

В примере (рис. 1.4) показано, как могут быть представлены данные и связи между ними в БД некоторой торговой организации. Таблица «Клиент» содержит данные о каждом клиенте – фамилию, имя и отчество. В таблице «Товар» хранятся сведения об имеющихся товарах. Когда клиент вступает во взаимоотношения с некоторым товаром (делает заказ), этот факт фиксируется в таблице «Заказ». Для того чтобы можно было сослаться на отдельную запись (строку) в некоторой таблице, каждая запись этой таблицы должна содержать уникальный идентификатор. В данном примере роль таких идентификаторов выполняют поля «Id_кл», «Id_тов» и «Id_зак». Поле таблицы, значения которого гаранти-

роvanно уникальны для каждой записи этой таблицы, называют **ключевым полем** или **ключом**. Ключ не обязательно должен быть числовым. Иногда уникальным идентификатором может служить не одно поле, а комбинация полей. При этом сочетание значений этих полей должно быть уникальным. Такие поля образуют **составной ключ** таблицы.

Вернемся к примеру. Обратим внимание, что поля «Клиент» и «Товар» содержат значения, совпадающие со значениями ключей таблиц «Клиент» и «Товар» (совпадение имен полей и названий таблиц не является обязательным). Каждая запись таблицы «Заказ» содержит информацию о заказе клиентом некоторого товара – дату заказа и количество единиц товара. В заказе также содержатся идентификаторы клиента и заказа, что позволяет однозначно определить – кем и на какой товар сделан заказ. Так, клиент Иванов заказал один шкаф, а позднее – двенадцать стульев. Клиент Николаев заказал только один шкаф, а Петров пока не заказал ничего.

Для изображения таблиц и связей между ними обычно применяется более компактная форма, в которой указываются имена полей, без данных. На рис. 1.5 приведена структура рассмотренной ранее базы данных.

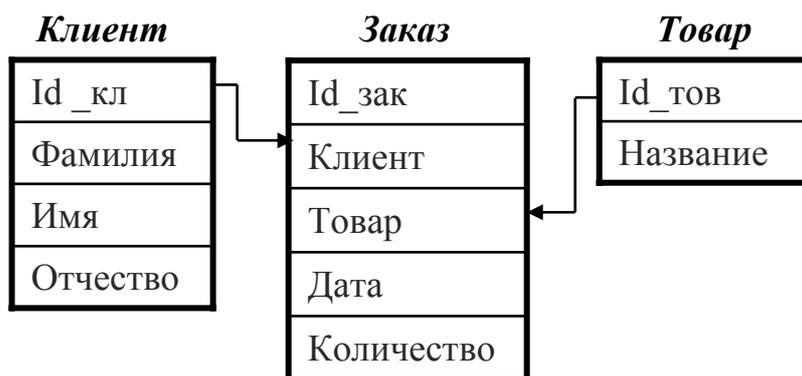


Рис. 1.5. Изображение таблиц и связей

Стрелки указывают «направление», в котором используются ключевые поля. Значения из поля «Id_кл» помещаются в поле «Клиент», а значения из поля «Id_тов» - в поле «Товар». Очевидно, что одному клиенту может соответствовать несколько заказов, но каждому заказу соответствует только один клиент. С товарами – аналогичная ситуация. Один товар может входить в несколько заказов, но в каждый заказ входит только один товар.

Вероятно, оформление заказов было бы более удобным, если бы в одном заказе можно было указывать сразу несколько товаров. Для этого структуру данных следует изменить.

На рис. 1.6 приведена новая структура БД. Таблица «Заказ» содержит для каждого заказа его уникальный идентификатор, идентификатор клиента, сделавшего этот заказ и дату. Перечень и количество товаров, входящих в заказ, хранится в таблице «Состав заказа». Одной записи в таблице «Заказ» может соответствовать несколько записей в таблице «Состав заказа».

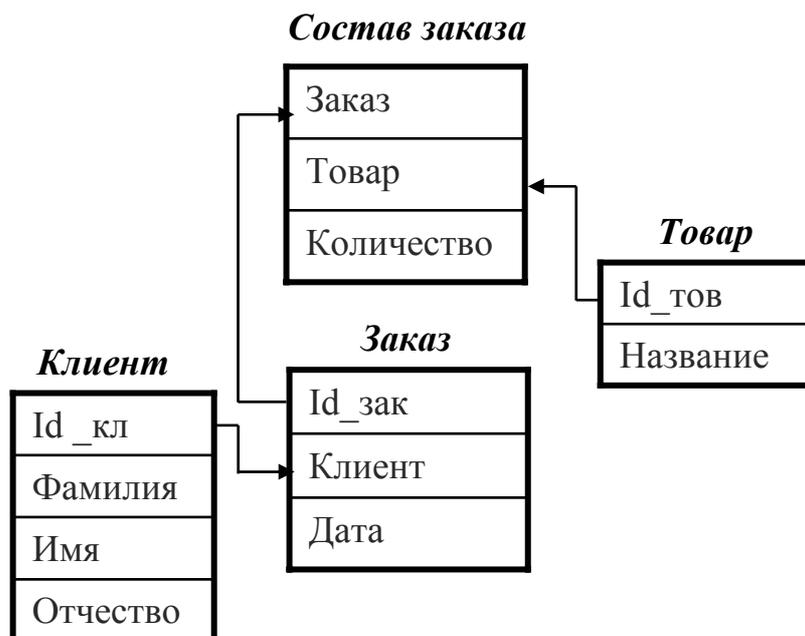


Рис. 1.6. Преобразованная структура БД

1.5. Системы управления базами данных

Система управления базами данных (СУБД) – комплекс программ и языковых средств, предназначенных для создания, ведения и использования баз данных. В англоязычной терминологии СУБД именуется DBMS (Database Management System).

Ведение базы данных – действия по добавлению, удалению и изменению хранимых данных.

Как видно из рис. 1.7, СУБД играет роль посредника между клиентом (человеком или программой) и данными.

Главные задачи, которые решает СУБД:

- Хранение данных.
- Поддержание целостности данных.
- Предоставление инструментария для манипулирования данными.

Задача хранения подразумевает размещение данных на некотором носителе информации (например, диске). Должен быть определен формат данных, обеспечивающий не только статичное хранение, но и эффективное манипулирование данными.

СУБД должна контролировать любые изменения в данных и гарантировать, что при любых действиях по манипулированию данными не возникнет противоречий в структуре или в самих данных (при условии, что такие противоречия не были заложены разработчиком). СУБД должна обеспечивать интерфейс (функции, программы) для доступа к манипулированию данными.

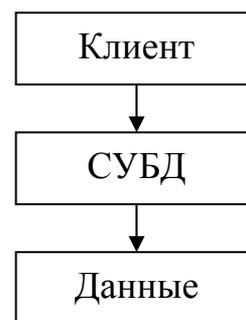


Рис. 1.7. Роль СУБД

Виды СУБД:

- Локальная СУБД.
- Система «клиент-сервер».

Различие заключается в степени «удаленности» данных и СУБД от потребителя этих данных – **клиента** (человека или программы).

СУБД, организованная по технологии «клиент-сервер» (рис. 1.8), предоставляет средства, позволяющие работать с данными клиентам, имеющим различное местоположение и использующим различное программное обеспечение. При этом клиентское программное обеспечение не отвечает за хранение данных и поддержание их целостности, оно лишь отправляет запросы к СУБД и получает на них ответы.

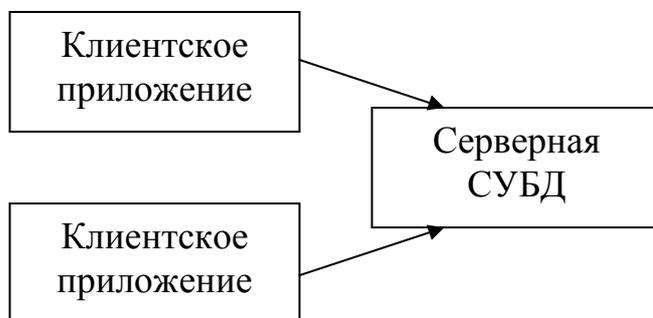


Рис. 1.8. Архитектура «клиент-сервер»

СУБД является **сервером**, хранящим данные, который обеспечивает некоторый интерфейс и при этом не отвечает за то, как эти данные могут быть использованы клиентами (внешними приложениями).

Локальные СУБД (рис. 1.9) интегрированы с конечным приложением – потребителем данных (программой автоматической обработки или интерфейсом пользователя). Соответственно, если с одними и теми же данными работает несколько приложений, каждое приложение должно включать в себя собственную СУБД.

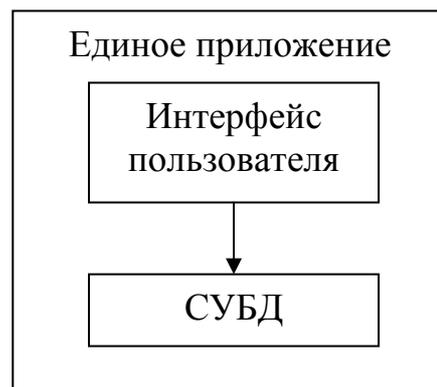


Рис. 1.9. Локальная СУБД

Многие локальные СУБД внутри организованы по технологии «клиент-сервер», но с точки зрения пользователя представляют собой единое приложение. В этом случае разделение скорее логическое – одни процедуры внутри программы отвечают только за доступ к данным и их целостность, другие выполняют только обработку и взаимодействие с пользователем, обращаясь к данным не напрямую, а лишь с помощью процедур первой группы.

Ниже перечислены некоторые из известных СУБД, относящиеся к первому и второму виду.

Серверные СУБД:

- Oracle
- Microsoft SQL
- MySQL
- mSQL
- PostgreSQL
- Caché

Локальные СУБД:

- Microsoft Access
- FoxPro
- Clarion
- Clipper

2. СУБД Microsoft Access 2000. Знакомство

Access – приложение, входящее в состав пакета Office, разработанного компанией Microsoft. Microsoft продвигает на рынок две различные СУБД – SQL Server и Access.

SQL Server предназначен для реализации крупных проектов, с большим числом пользователей, одновременно работающих с БД. Он является (как следует из названия) серверным приложением и способен обрабатывать запросы от удаленных клиентов.

Access представляет собой **локальную СУБД** и ориентирован на разработку баз данных малой и средней сложности, с которыми одновременно могут работать один или несколько пользователей. Приложение обращается непосредственно к файлу базы данных, доступ к файлу ограничен пределами одного компьютера или локальной сети. Данные не кэшируются в оперативной памяти (при каждом запросе выполняется обращение к файлу), что гарантирует устойчивость БД по отношению к аппаратным сбоям, но негативно влияет на производительность. Для защиты целостности данных при обращении к ним производится блокировка (два пользователя не могут одновременно изменить содержимое одной и той же ячейки). Файлы баз данных Access имеют расширение «.MDB».

Access предоставляет простые и удобные возможности для разработки структуры БД и пользовательского интерфейса. Одно и то же приложение служит как инструментом создания баз данных, так и средством работы с готовыми БД. На рис. 2.1 показан вид окна Access с открытой в нем базой данных. В левой части окна базы данных перечислены виды групп объектов, которые могут присутствовать в БД (таблицы, запросы, формы и т.п.). В правой части окна отображается список объектов, принадлежащих конкретной выбранной группе.

2.1. Инструменты для работы с данными в Access

База данных Access может содержать объекты следующих видов:

- Таблицы
- Запросы
- Формы
- Отчеты
- Страницы
- Макросы
- Модули

Все эти объекты являются инструментами, позволяющими выполнять различные действия над данными.

Таблицы. Объекты-таблицы (рис. 2.2) предоставляют пользователю непосредственный доступ к просмотру, изменению и добавлению хранимых данных. В соответствии с требованиями реляционной модели, все данные в Access хранятся именно в таблицах. Остальные объекты – лишь средства работы с этими данными.

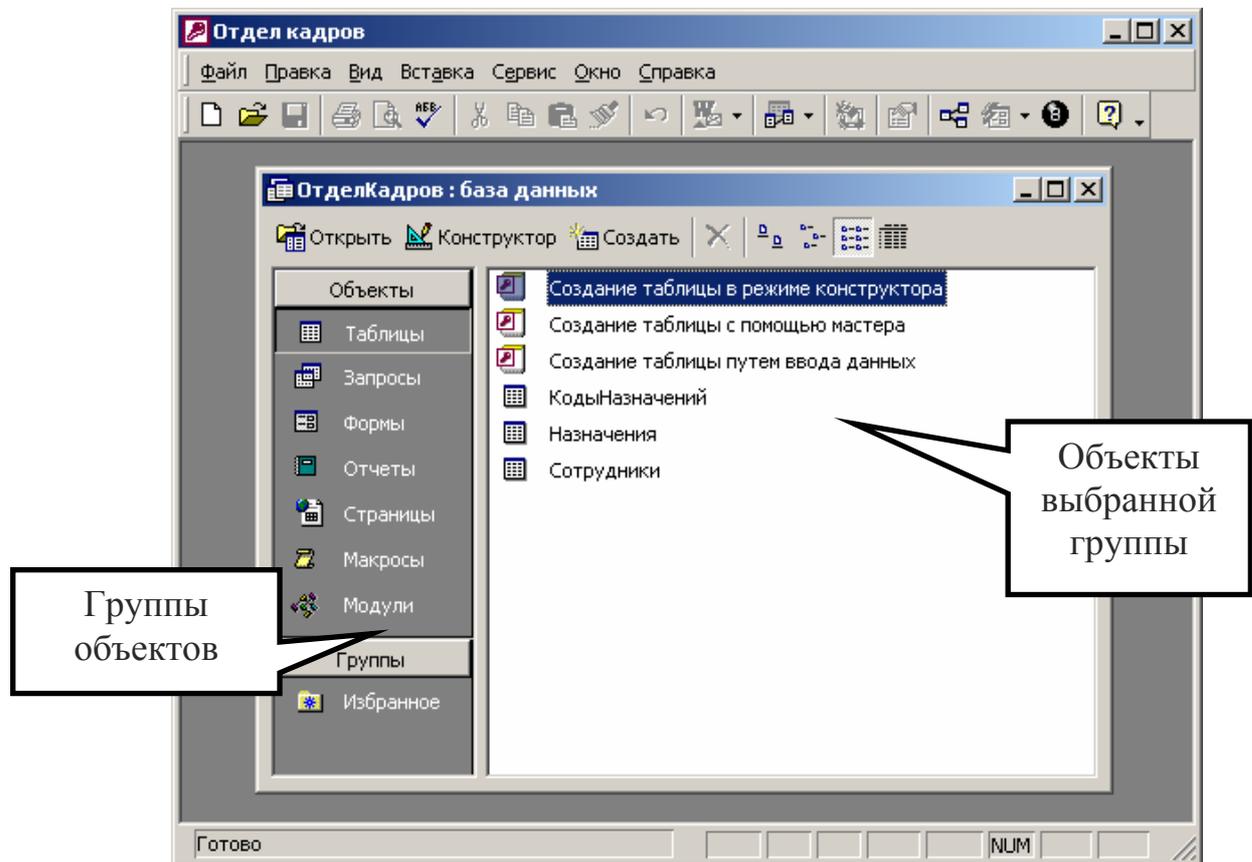


Рис. 2.1. Вид приложения с загруженной БД

The screenshot shows the 'Отдел кадров - [Сотрудники : таблица]' window. The table displays employee data with columns: Код, Ном, Таб, Фамилия, Имя, Отчество, ДатаРождения, and Ит. The status bar at the bottom shows 'Запись: 1 из 22' and 'Режим таблицы'.

	Код	Ном	Таб	Фамилия	Имя	Отчество	ДатаРождения	Ит
+	1			Бобцов	Алексей	Алексеевич	01.01.1973	
+	2			Бойков	Владимир	Иванович	01.01.1955	
+	3			Болтунов	Геннадий	Иванович	02.02.1957	
+	4			Болтунов	Иван	Павлович	03.03.1930	
+	5			Бушуев	Александр	Борисович	04.04.1950	
+	6			Быстров	Сергей	Владимирович	01.01.1967	
+	7			Власенко	Владимир	Андреевич	01.03.1938	
+	8			Григорьев	Валерий	Владимирович	05.03.1942	
+	9			Коровьяков	Анатолий	Николаевич	05.01.1943	
+	10			Котельников	Юрий	Петрович	07.04.1948	
+	11			Мирошник	Илья	Васильевич	07.03.1948	
+	12			Никифоров	Владимир	Олегович	12.05.1962	
+	13			Салмыгин	Юрий	Петрович	14.07.1952	
+	14			Сергеев	Константин	Александрович	11.05.1975	
+	15			Ушаков	Анатолий	Владимирович	30.04.1938	
+	16			Хабалов	Владимир	Викторович	25.06.1944	
+	17			Чежин	Михаил	Сергеевич	12.07.1960	
+	18			Черноусов	Виталий	Витальевич	02.06.1948	
+	19			Руквйжа	Елена	Викторовна	08.06.1966	

Рис. 2.2. Пример таблицы

Запросы. Запросы (рис. 2.3) позволяют извлекать данные из одной или нескольких таблиц в соответствии с заданными условиями отбора, производить при этом обработку и сортировку данных. Визуально запросы напоминают таблицы, однако их принципиальное отличие в том, что отображаются не сами данные, которые хранятся в БД, а то, что является результатом их переработки. Запросы могут быть параметризованными (иметь параметры). При вызове такого запроса пользователь должен задать значения параметров.

Фамилия	ДатаРождения
Волков	03.06.1983
Василенко	06.02.1982
Сергеев	11.05.1975
Бобцов	01.01.1973
Кожевникова	21.05.1967
Быстров	01.01.1967
Рукуйжа	08.06.1966
Никифоров	12.05.1962
Чижин	12.07.1960

Рис. 2.3. Пример запроса

Формы. Формы (рис. 2.4) предоставляют пользователю более удобные средства по работе с данными, чем непосредственное редактирование таблиц. Форма обычно привязана к одному или нескольким источникам данных (таблицам или запросам) и позволяет манипулировать данными из этих источников с помощью визуальных средств. Форма может и не быть связана с источниками данных, а служить пользовательским интерфейсом – содержать кнопки и другие элементы управления.

Код сотрудника	Номер паспорта	Табельный номер
1		

Фамилия	Имя	Отчество
Бобцов	Алексей	Алексеевич

ДатаРождения	Индекс
01.01.1973	

Дата/Время	Назначен	Приказ
20.09.2003	Курьер	
20.09.2003	Менеджер	
* 05.10.2003		

Рис. 2.4. Пример формы

Отчеты. Отчеты (рис. 2.5) представляют данные в виде, пригодном для вывода на печать. Данные в отчеты подставляются напрямую из таблиц или из результатов выполнения запросов. Например, можно создать отчет в форме стандартного бланка организации, в определенные поля которого подставляются соответствующие данные.

Страницы. Страницы (рис. 2.6) – документы в формате HTML, которые могут быть просмотрены в web-обозревателе (точнее, только в MS Internet Explorer). Страницы имеют то же назначение, что и формы – пользовательский интерфейс для доступа к данным. Страницы хранятся за пределами файла базы данных. Для работы с ними не требуется запускать приложение Access.

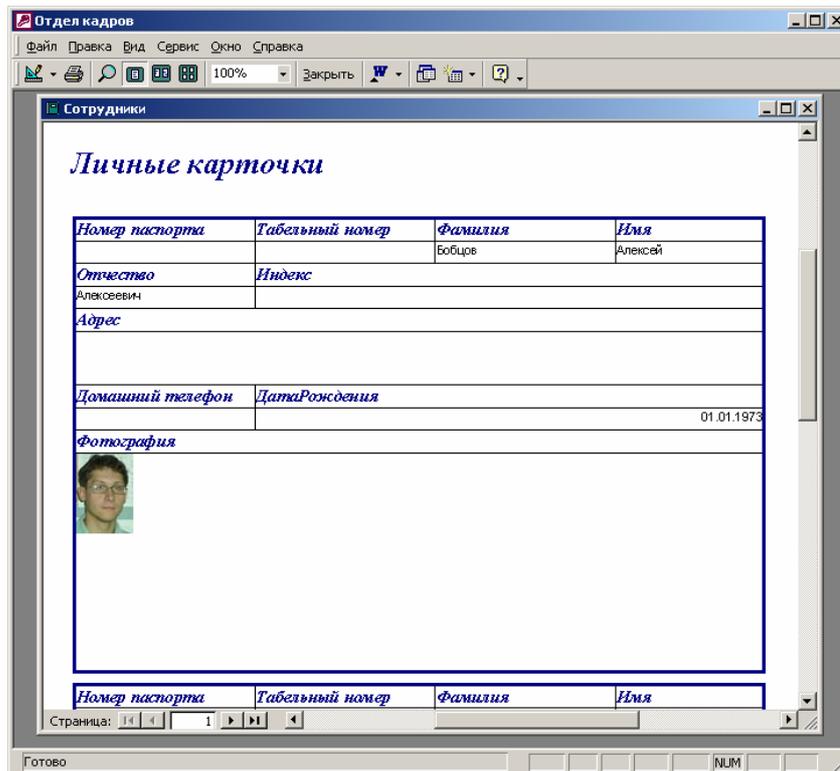


Рис. 2.5. Пример отчета

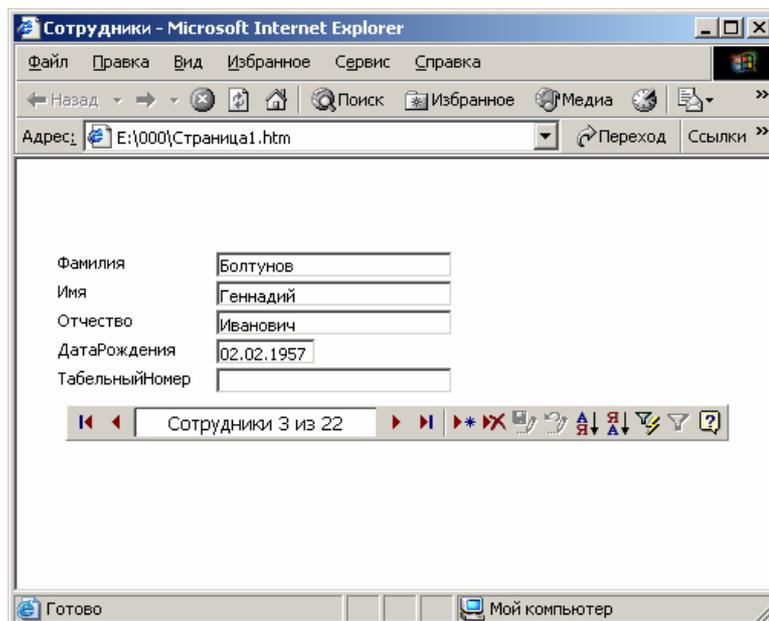


Рис. 2.6. Пример страницы доступа к данным

Макросы и модули. Макрос – определяемая пользователем последовательность действий. Если требуется часто выполнять одни и те же операции (например, открытие и последующую печать отчета), можно их оформить в виде макроса. При запуске макроса выполняются все заданные в нем действия. Модуль – набор подпрограмм на языке Visual Basic, составленных разработчи-

ком БД и предназначенных для выполнения нетривиальных операций над данными, которые сложно реализовать иными средствами.

2.2. Работа с готовой базой данных

Для запуска программы Microsoft Access можно воспользоваться соответствующим ярлыком  Microsoft Access из меню программ Windows. При запуске пользователю будет предложено выбрать одно из действий: создать пустую базу данных, создать БД по шаблону или открыть существующую БД. В последнем случае можно воспользоваться списком ранее открывавшихся файлов или выбрать другой файл, указав его расположение на диске. Ниже будут рассмотрены некоторые приемы работы с готовой базой данных. В качестве демонстрационного примера будет использована учебная БД «Борей», входящая в состав MS Access. БД хранит данные, относящиеся к сфере деятельности воображаемой компании «Борей». Знакомство с базой данных «Борей» может быть полезно как начинающим пользователям, так и разработчикам. Однако будет ошибкой думать, что Access предназначен для работы только с этой базой данных. Это лишь пример. Все рассматриваемые ниже объекты – формы, отчеты, запросы и таблицы были созданы разработчиками базы данных «Борей». Другая БД может содержать другие формы, таблицы и т.п. Какие именно – будет зависть от ее заказчика и разработчика. Способы создания разных объектов БД мы рассмотрим позднее. БД «Борей» размещается в файле «Борей.mdb». Файл расположен в каталоге Samples каталога, в котором установлен Access, например

«E:\Program Files\Microsoft Office\Office\Samples\».

При работе с базами данных пользователю обычно приходится иметь дело с объектами следующих типов:

- Таблицы
- Запросы
- Формы
- Отчеты

Далее будут рассматриваться именно эти объекты из БД «Борей».

Формы. Навигация в формах. При открытии БД «Борей» запускается окно – заставка (рис. 2.7). Его можно закрыть, нажав на кнопку «ОК». Заставка является формой. Ее можно увидеть в перечне всех форм базы данных. Как и любую форму, заставку можно открыть позднее, в процессе работы с БД. После закрытия заставки становится доступным окно базы данных (рис. 2.8). Выберем в левой колонке тип объектов «Формы» и двойным щелчком откроем форму «Главная кнопочная форма» (рис. 2.9). Главная кнопочная форма содержит набор кнопок, предназначенных для выполнения наиболее распространенных действий в БД «Борей». Нажмем кнопку «Типы» для запуска формы, предназначенной для работы с типами (категориями) товаров. В верхней части формы на рис. 2.10 располагаются сведения об одной из категорий товара: название категории, описание категории и некоторая картинка (иллюстрация). В нижней части показан список товаров, относящихся к выбранной категории.

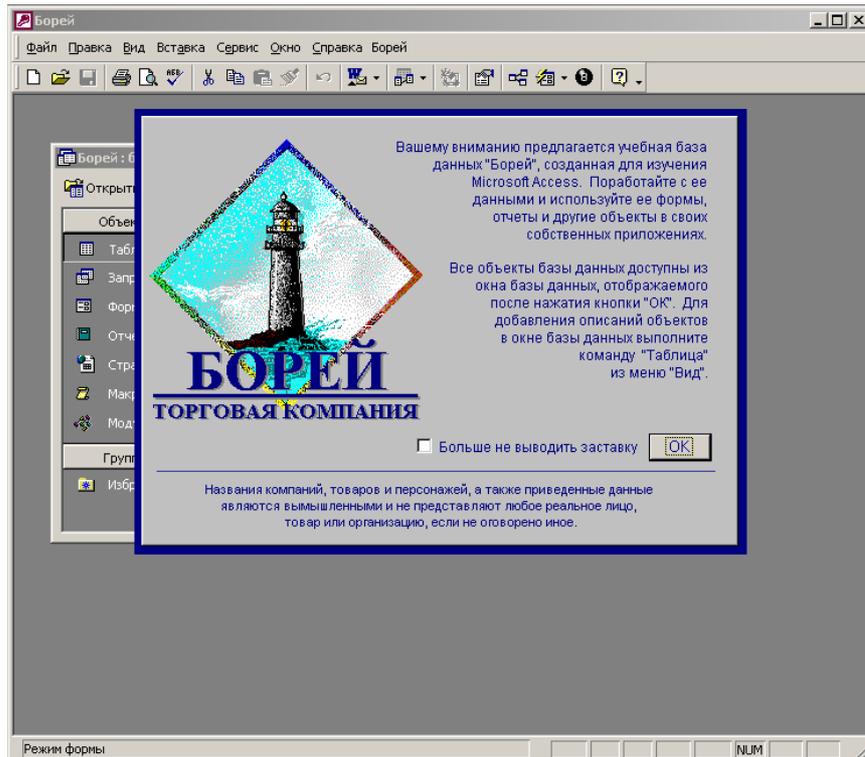


Рис. 2.7. Заставка БД «Борей»

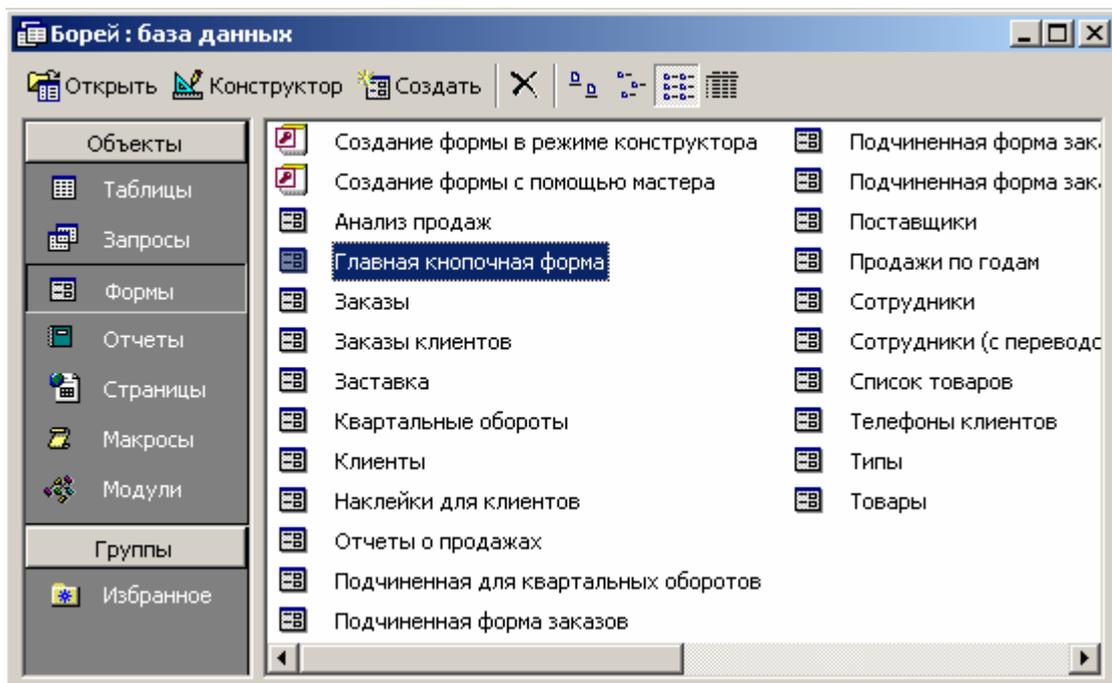


Рис. 2.8. Окно БД «Борей»

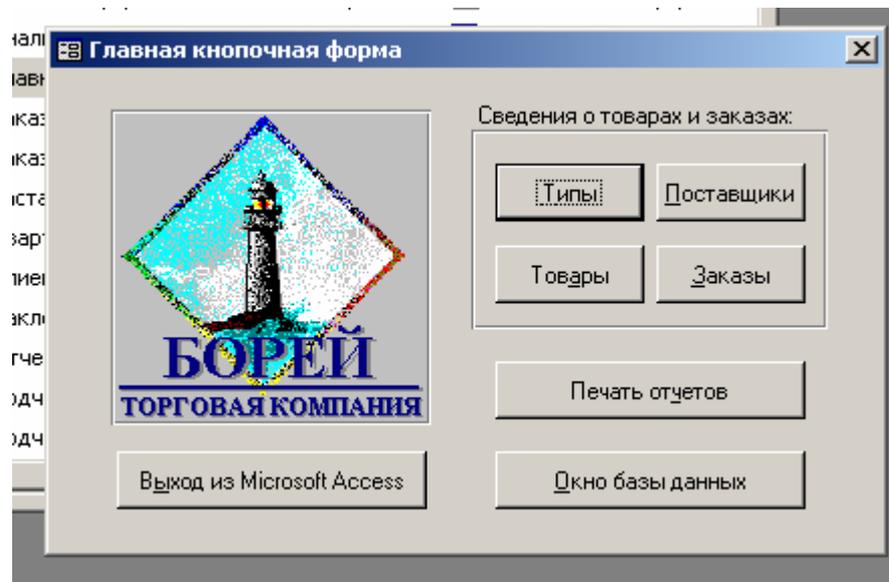


Рис. 2.9. Главная кнопочная форма БД «Борей»

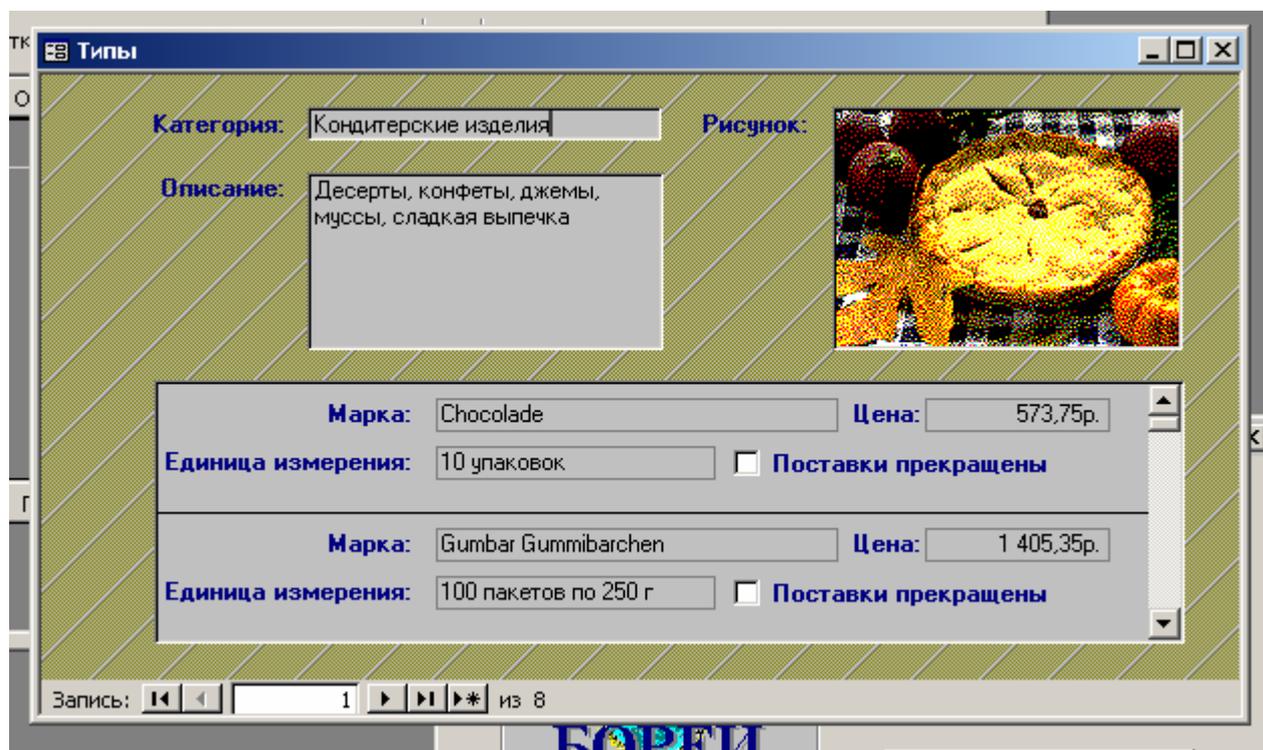


Рис. 2.10. Типы (категории) товаров

Данные, отображаемые в форме, берутся из двух разных таблиц. Для верхней части – из таблицы «Типы», а для нижней – «Товары». Каждый товар в таблице «Товары» связан с определенной категорией из таблицы «Типы». Эта связь используется для отображения в нижней части формы только тех товаров, которые имеют соответствующий тип. Форма позволяет просматривать и добавлять категории товаров и для каждой категории видеть товары, к ней относящиеся. При перемещении к другой категории товара изменяется перечень товаров в нижней части формы. На нижней границе окна формы находятся стан-

дартные элементы навигации – «перелистывания» категорий (рис. 2.11). В каждый момент времени отображается информация только об одной категории – т.е. только одна запись из таблицы «Типы». Элементы навигации позволяют перемещаться от одной записи к другой. Кроме элементов навигации формы также могут быть полезны кнопки «Добавить новую запись»  и «Удалить текущую запись»  на панели инструментов (в верхней части окна Access).

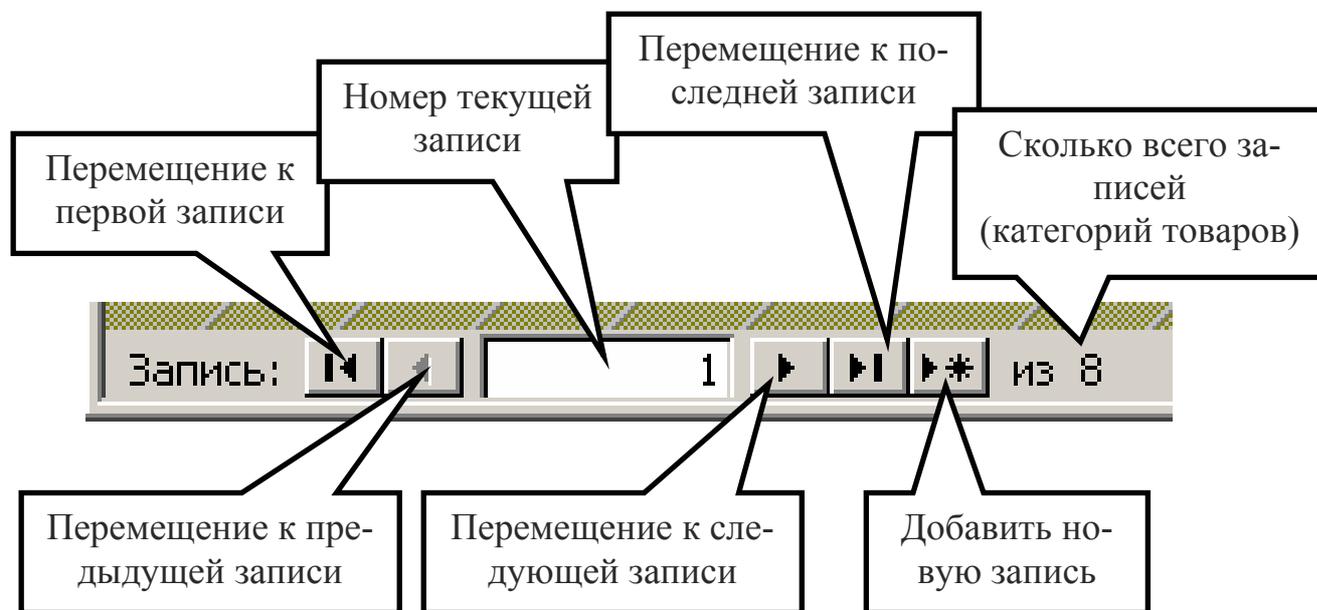


Рис. 2.11. Элементы навигации формы

Сортировка, фильтрация и поиск. Кроме описанных выше средств навигации, при работе с данными могут использоваться также сортировка, фильтрация и поиск. Кнопки **сортировки** на панели инструментов  позволяют упорядочить записи по возрастанию или убыванию значений выбранного поля. Перед нажатием на одну из этих кнопок следует поместить курсор в то поле, по которому должна выполняться сортировка.

Фильтрация дает возможность ограничить множество просматриваемых записей только теми, которые удовлетворяют некоторому условию. Для управления фильтрацией используются три кнопки  на панели инструментов – «Фильтр по выделенному», «Изменить фильтр» и «Применение фильтра/удалить фильтр».

Кнопка «Фильтр по выделенному» позволяет, выделив некоторый фрагмент содержимого какого либо поля, ограничить просмотр только теми записями, которые содержат в этом поле аналогичные фрагменты.

Другой способ – создание определенного фильтра и последующее его применение. Рассмотрим этот способ подробнее на примере формы «Товары» (которая, как и форма «Типы», может быть открыта из главной кнопочной формы).

Форма «Товары» (рис. 2.12) предназначена для просмотра, редактирования и добавления сведений о товарах (хранящихся в таблице «Товары»). Форма в каждый момент времени отображает содержимое одной записи из таблицы.

The screenshot shows a web form titled «Товары» with a blue header bar. The form contains several input fields and buttons. At the top right, there are two buttons: «Просмотр списка товаров» and «Вывод списка в файл HTML». The main form area has the following fields:

- Товар:** 1
- Марка:** Genen Shouyu
- Поставщик:** Мауми's
- Тип:** Приправы
- Единица измерения:** 24 бутылок по 250 мл
- Цена:** 697,50р.
- На складе:** 39
- Ожидается:** 0
- Минимальный запас:** 5
- Поставки прекращены:**

At the bottom left, there is a pagination control: «Запись:» followed by navigation arrows and the number «1» in a box, and «из 77».

Рис. 2.12. Форма «Товары»

Обратим внимание, что в нижней части окна формы показано общее количество различных товаров – 77.

Для определения фильтра нажмем на кнопку «Изменить фильтр» . Окно формы примет вид, позволяющий редактировать фильтр (рис. 2.13).

The screenshot shows a web form titled «Товары: фильтр» with a blue header bar. The form contains several input fields and buttons. At the top right, there are two buttons: «Просмотр списка товаров» and «Вывод списка в файл HTML». The main form area has the following fields:

- Товар:** (empty)
- Марка:** (empty)
- Поставщик:** (empty)
- Тип:** "Фрукты"
- Единица измерения:** (empty)
- Цена:** (empty)
- На складе:** (empty)
- Ожидается:** (empty)
- Минимальный запас:** (empty)
- Поставки прекращены:**

At the bottom left, there is a search control: «Найти» followed by two «Или» buttons.

Рис. 2.13. Редактирование фильтра

В фильтре можно для одного или нескольких полей указать условия фильтрации (какие значения должны содержаться в полях записи, чтобы эта запись соответствовала фильтру). Если фильтр допускает для поля несколько различных значений, их можно задать, используя вкладки «Или» (в нижней части окна).

В приведенном на рисунке 2.13 примере фильтр содержит простое условие: отображать только товары, имеющие тип «Фрукты». Создав фильтр, нажмем на кнопку «Применение фильтра» .

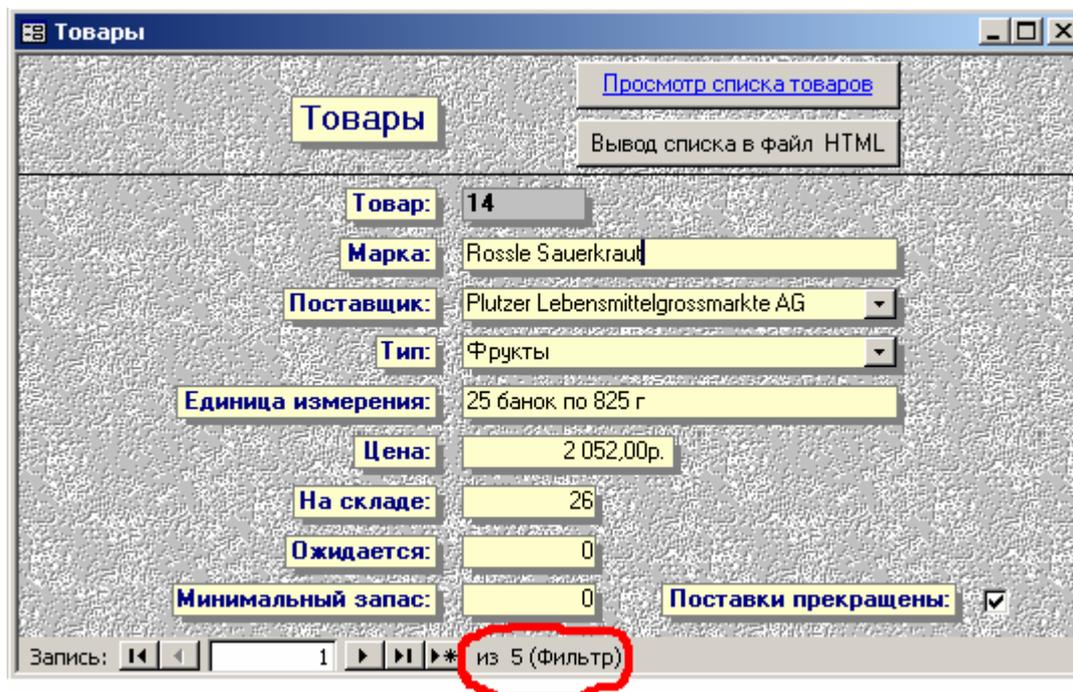


Рис. 2.14. Форма с примененным фильтром

Как видно из рисунка 2.14, после применения фильтра доступными для просмотра остались 5 записей из 77. Именно столько товаров относится к типу «Фрукты». Для отмены фильтрации следует нажать кнопку «Удалить фильтр»

 на панели инструментов. Форма вернется в режим отображения списка всех записей (рис. 2.12).

Функция **поиска** позволяет находить записи, некоторое поле которых совпадает с заданным образцом. Рассмотрим поиск данных на примере той же формы «Товары». Предположим, требуется найти товары, поставляемые в банках. Поместим курсор в поле формы «Единица измерения». Нажмем на панели инструментов кнопку «Найти» .

Откроется диалоговое окно поиска (рис. 2.15). В поле «Образец» введем слово «банок». В поле «Совпадение» выберем вариант «С любой частью поля» и нажмем кнопку «Найти далее». В форме отобразится первая найденная запись, содержащая слово «банок» в поле «Единица измерения» (рис. 2.16). При повторном нажатии на кнопку «Найти далее» будет показана следующая запись, удовлетворяющая заданному условию поиска. Если в диалоговом окне (рис. 2.15) выбрать вкладку «Замена», можно задать значение, которое будет подставляться вместо слова «банок» в найденные записи.

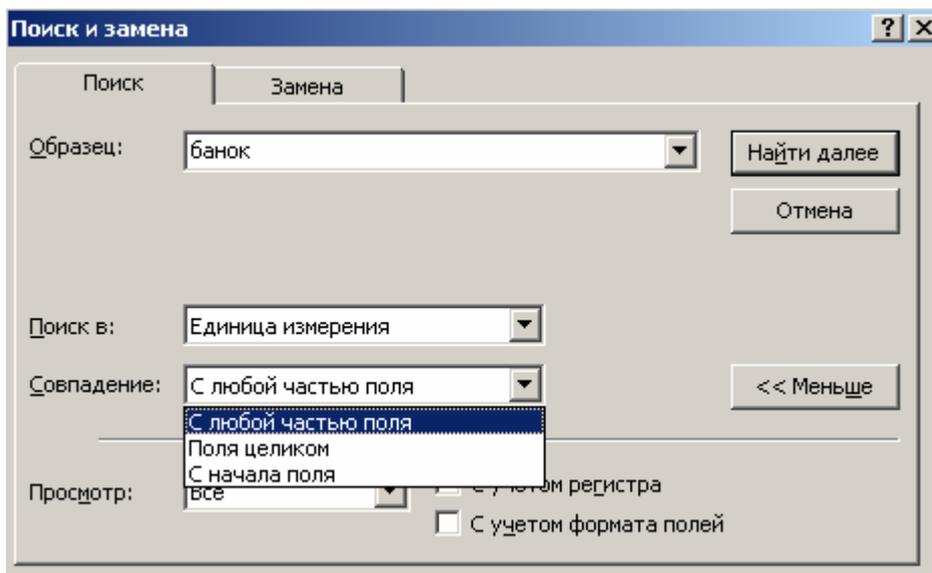


Рис. 2.15. Диалоговое окно поиска

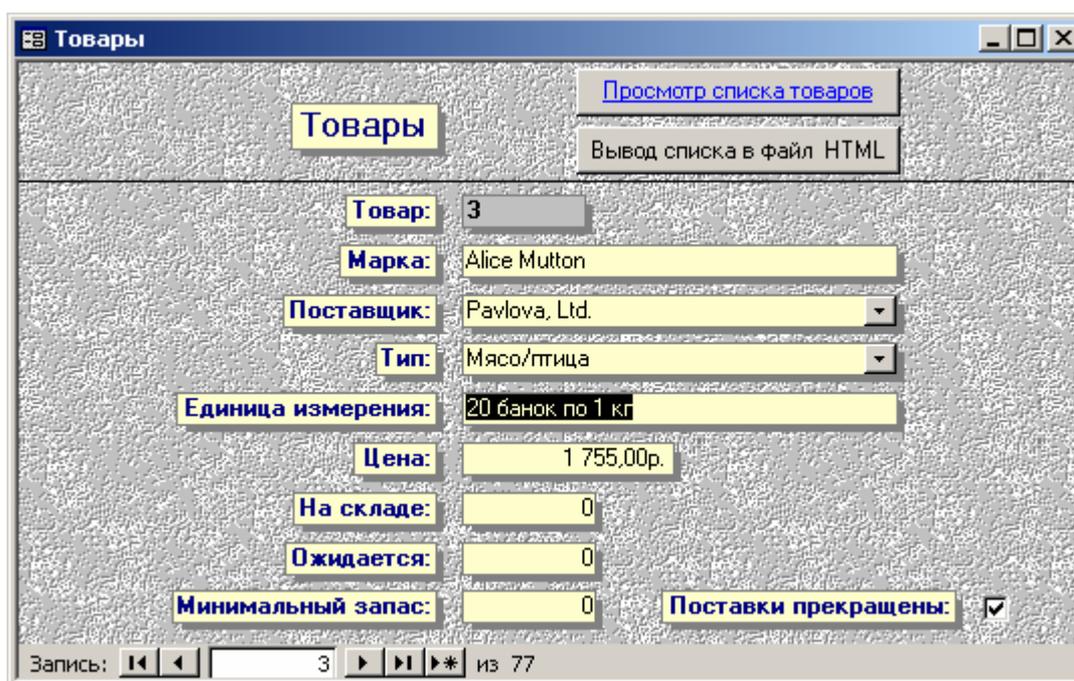


Рис. 2.16. Результат поиска

Отчеты. Вернемся к окну «Главная кнопочная форма» (рис. 2.9). Нажмем в нем кнопку «Печать отчетов». В появившемся окне (которое также является формой) выберем пункт «Итоги продаж по объему» и нажмем кнопку «Просмотр» (рис. 2.17). Откроется окно отчета, данные в котором представлены так же, как они будут выглядеть при печати (рис. 2.18). Отчет разбит на страницы в соответствии с заданным форматом листов и настройками принтера. Кнопка «Печать»  на панели инструментов отправит открытый отчет на принтер.

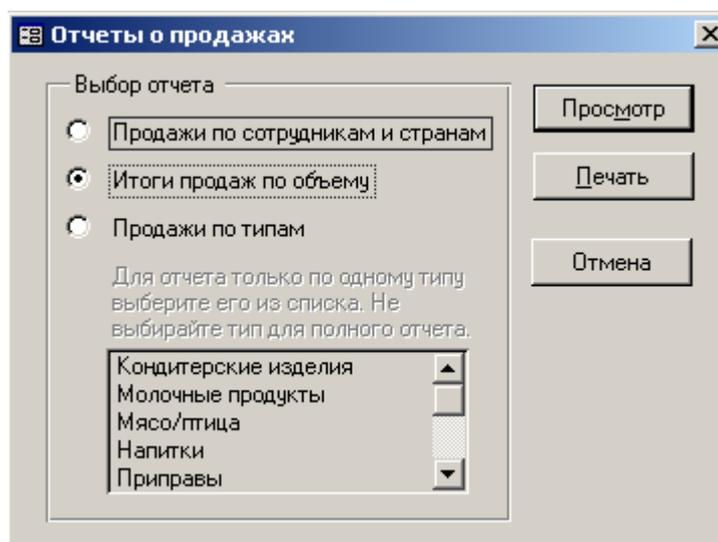


Рис. 2.17. Выбор вида отчета

Рис. 2.18. Отчет «Итоги продаж по объему»

Показанный выше отчет и все остальные отчеты в БД расположены в разделе «Отчеты» окна базы данных (рис. 2.19). Для открытия отчета нужно дважды щелкнуть по его названию мышкой.

Запросы. Запросы, как и остальные объекты, размещаются в соответствующем разделе окна БД (рис. 2.20). Двойной щелчок мышью по имени запроса вызывает открытие его окна. Запросы извлекают данные из БД, отображая их в табличном виде. Откроем запрос «Запрос Заказы» (рис. 2.21). Запрос отображает данные из таблицы «Заказы», дополняя их данными о клиентах, разместивших заказы (из таблицы «Клиенты»). Т.е. данные о заказах и клиентах хранятся в отдельных таблицах, но при выполнении запроса они отображаются совместно, также в форме таблицы. В этом отличие между двумя видами объектов базы

данных – запросами и таблицами: запросы не хранят данные, а лишь извлекают их из таблиц, и выполняют обработку.

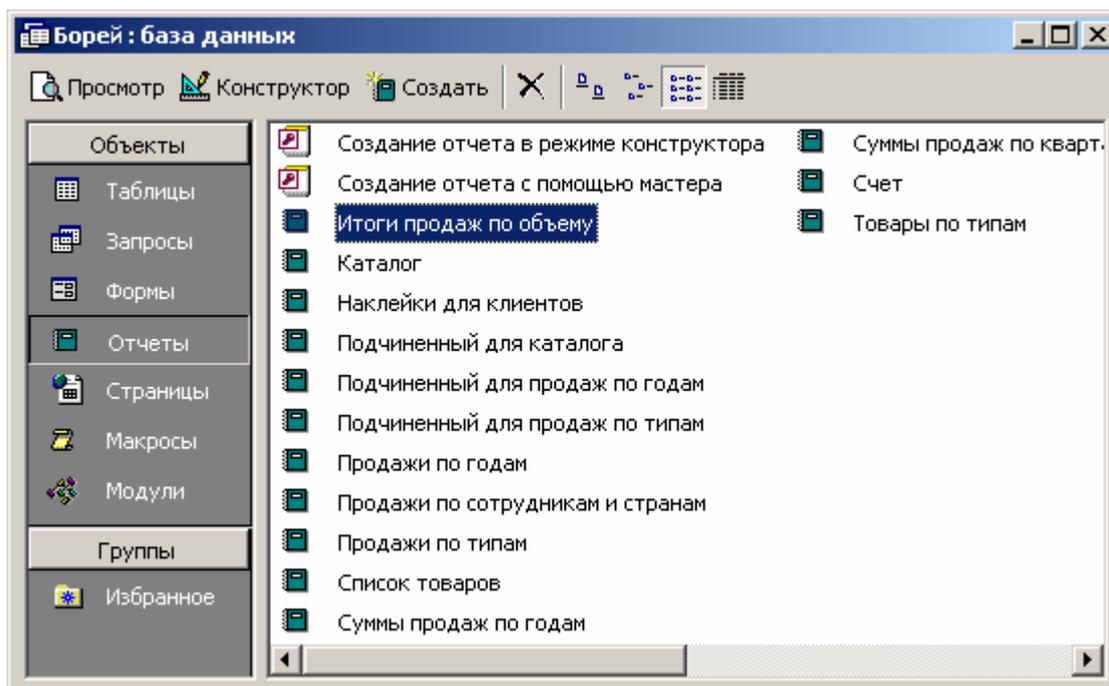


Рис. 2.19. Список отчетов

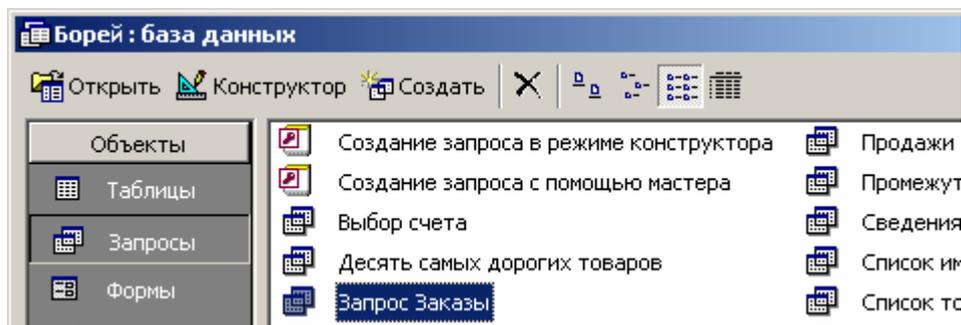


Рис. 2.20. Список запросов

The screenshot shows a data table window titled 'Запрос Заказы : запрос на выборку'. The table contains the following data:

Код заи	Клиент	Сотрудник	Дата размещ	Дата на
10702	Alfreds Futterkiste	Воронова, Дарья	13-10-1997	24-
10952	Alfreds Futterkiste	Белова, Мария	16-03-1998	27-
11011	Alfreds Futterkiste	Бабкина, Ольга	09-04-1998	07-
10308	Ana Trujillo Emparellados	Кралев, Петр	18-09-1996	16-
10625	Ana Trujillo Emparellados	Бабкина, Ольга	08-08-1997	05-
10759	Ana Trujillo Emparellados	Бабкина, Ольга	28-11-1997	26-
10926	Ana Trujillo Emparellados	Воронова, Дарья	04-03-1998	01-
10365	Antonio Moreno Taqueria	Бабкина, Ольга	27-11-1996	25-
10507	Antonio Moreno Taqueria	Кралев, Петр	15-04-1997	13-
10535	Antonio Moreno Taqueria	Воронова, Дарья	13-05-1997	10-
10573	Antonio Moreno Taqueria	Кралев, Петр	19-06-1997	17-
10677	Antonio Moreno Taqueria	Кралев, Петр	22-09-1997	20-
10682	Antonio Moreno Taqueria	Бабкина, Ольга	25-09-1997	23-
10856	Antonio Moreno Taqueria	Бабкина, Ольга	28-01-1998	25-

At the bottom of the window, it indicates 'Запись: 1 из 830'.

Рис. 2.21. Результат запроса

Как и при работе с формами, к данным можно применять операции сортировки , фильтрации  и поиска . В нижней части окна находятся элементы навигации (к первой записи, к последней и т.п.)

Таблицы, схема данных. В разделе «Таблицы» базы данных «Борей» отображается список всех таблиц, имеющих в БД (рис. 2.22). Как уже было сказано, все данные в БД хранятся именно в таблицах, а остальные объекты (запросы, формы, отчеты) используются для работы с этими данными.

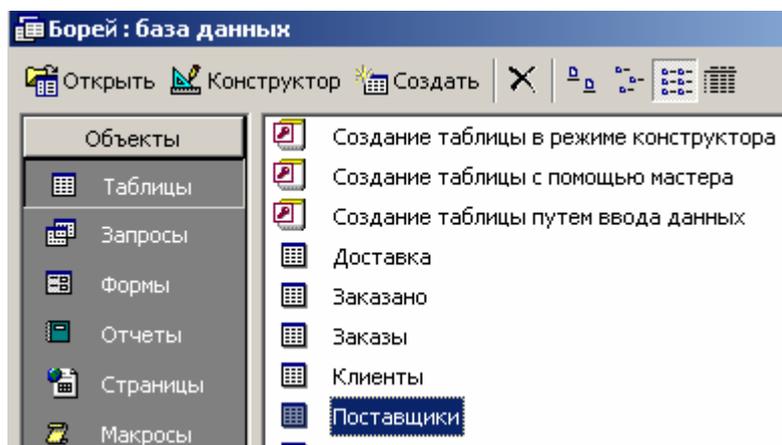


Рис. 2.22. Список таблиц

Однако можно просматривать и редактировать данные и непосредственно в таблицах. Иногда такой способ может оказаться даже более удобным, чем работа с формами. Но при этом следует действовать более осторожно: разработчики БД, создавая пользовательский интерфейс в виде форм, закладывают в него возможность выполнять только те операции, которые, по их мнению, необходимы. При работе же напрямую с таблицами ограничений нет и пользователю легче совершить непоправимые действия.

В таблице «Поставщики» (рис. 2.23) содержатся сведения о поставщиках товаров, с которыми сотрудничает компания «Борей». Каждому поставщику присвоен уникальный числовой код (*ключ*), отличающий его от всех остальных. По внешнему виду таблицы похожи на запросы (точнее – наоборот, запросы представляются в форме таблиц). В нижней части окна таблицы расположены средства навигации. При работе с таблицами также доступны функции сортировки, фильтрации и поиска. Таблица «Товары» (рис. 2.24) хранит сведения о товарах.

Для каждого товара, в частности, указана информация о его поставщике. В поле с заголовком «Поставщик» хранится код поставщика из таблицы «Поставщики». При просмотре таблицы вместо кода поставщика отображается его наименование (как это делается, мы выясним позднее). Поле «Поставщик», используя значения из **ключевого поля** таблицы «Поставщики», реализует **связь** между таблицами. В это поле можно поместить только значение, которому соответствует некоторая запись в таблице «Поставщики», при попытке указать некорректное значение будет выдано сообщение об ошибке.

	Код по	Название	Обращаться к	Должность
+	1	ООО Экзотика	Вероника Кудрявцева	Менеджер по закупкам
+	2	New Orleans Cajun Delights	Shelley Burke	Координатор
+	3	Grandma Kelly's Homestead	Regina Murphy	Представитель
+	4	Tokyo Traders	Yoshi Nagase	Главный менеджер
+	5	Cooperativa de Quesos 'Las C	Antonio del Valle Saavedra	Специалист по закупкам
+	6	Mayumi's	Mayumi Ohno	Директор
+	7	Pavlova, Ltd.	Ian Dewling	Главный менеджер
+	8	Specialty Biscuits, Ltd.	Peter Wilson	Представитель
+	9	PB Knackebrod AB	Lars Peterson	Продавец
+	10	Refrescos Americanas LTDA	Carlos Diaz	Главный менеджер
+	11	АО Германия-Россия	Петр Моргунов	Менеджер по закупкам
+	12	Plutzer Lebensmittelgrossmar	Martin Bein	Менеджер по закупкам
+	13	Nord-Ost-Fisch mbH	Sven Petersen	Внешний координатор
+	14	Formaggi Fortini s.r.l.	Elio Rossi	Представитель
+	15	Norske Meierier	Bente Wiland	Главный менеджер

Рис. 2.23. Окно таблицы «Поставщики»

	Код тов	Марка	Поставщик	Тип
+	1	Genen Shouyu	Mayumi's	Приправы
+	2	Pavlova	Pavlova, Ltd.	Кондитерские изделия
+	3	Alice Mutton	Pavlova, Ltd.	Мясо/птица
+	4	Carnarvon Tigers	Pavlova, Ltd.	Рыбные продукты
+	5	Teatime Chocolate Biscuit	Specialty Biscuits, Ltd.	Кондитерские изделия
+	6	Sir Rodney's Marmalade	Specialty Biscuits, Ltd.	Кондитерские изделия
+	7	Sir Rodney's Scones	Specialty Biscuits, Ltd.	Кондитерские изделия
+	8	Gustafs Knackehrd	PB Knackehrd AB	Хлебобулочные изделия

Рис. 2.24. Таблица «Товары»

Связи между таблицами можно задать и просмотреть с помощью инструмента «Схема данных» (кнопка  на панели инструментов). Рассмотрим схему данных БД «Борей» (рис. 2.25).

Между полем «КодПоставщика» таблицы «Поставщики» (выделение названия поля означает, что оно является ключевым) и полем «КодПоставщика» таблицы «Товары» (отметим, что названия полей не совпадают с заголовками, отображаемыми при просмотре таблиц) изображена линия, обозначающая наличие связи. Символы 1 и ∞ на концах линии связи показывают, что одной записи из таблицы «Поставщики» может соответствовать много (или ни одной) записей из таблицы «Товары», но каждой записи в таблице «Товары» соответствует только одна запись из таблицы «Поставщики», определяемая уникальным кодом поставщика (ключом).

Вернемся теперь к таблице «Поставщики» (рис. 2.23). В каждой строке таблицы слева расположен символ «+». Если щелкнуть по нему мышкой, в окне этой таблицы откроется подтаблица, отображающая записи из таблицы «Товары», связанные с выбранным поставщиком (рис. 2.26).

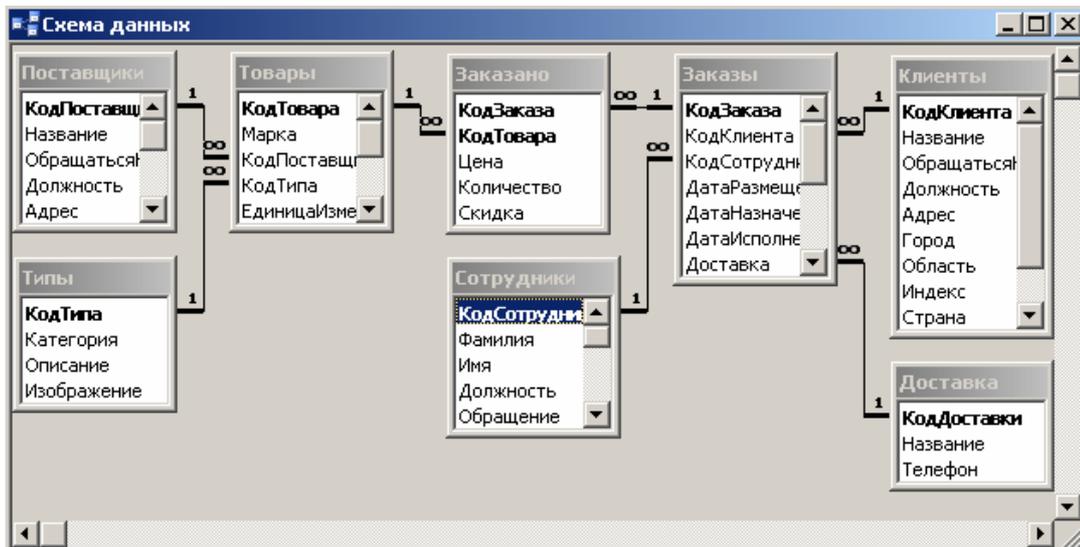


Рис. 2.25. Схема данных

Поставщики : таблица					
	Код по	Название	Обращаться к	Должн	
+	1	ООО Экзотика	Вероника Кудрявцева	Менеджер по	
+	2	New Orleans Cajun Delights	Shelley Burke	Координатор	
+	3	Grandma Kelly's Homestead	Regina Murphy	Представите.	
▶	-	4	Tokyo Traders	Yoshi Nagase	Главный мен
		Код товара	Марка	Поставщик	Тип
▶	+	28	Longlife Tofu	Tokyo Traders	Фрукты
	+	72	Mishi Kobe Niku	Tokyo Traders	Мясо/птица
	+	73	Ikura	Tokyo Traders	Рыбопродук
	*	(Счетчик)		Tokyo Traders	
+	5	Cooperativa de Quesos 'Las C	Antonio del Valle Saave	Специалист г	
+	6	Mayumi's	Mayumi Ohno	Директор	

Рис. 2.26. Отображение Таблицы с подтаблицей

Подтаблицы позволяют представить данные иерархически, отображая в одном окне содержимое нескольких таблиц, между которыми существуют связи. Такая возможность может оказаться удобной для просмотра и редактирования взаимосвязанных данных.

2.3. Интеграция с приложениями MS Office

Каждое из приложений, входящих в Microsoft Office, ориентировано на определенный класс выполняемых задач:

- **MS Word** предназначен для работы с текстом. Он предоставляет пользователю широкие возможности по форматированию и печати текстовых документов.
- **MS Excel** выполняет, в основном, обработку числовых данных, производя над ними достаточно сложные вычисления.
- **MS Access** используется для хранения различных данных в виде связанных таблиц и выполнения манипуляций над ними.

Способность приложений к взаимодействию позволяет передавать данные из одного приложения в другое, выбирая наиболее подходящий способ их обработки.

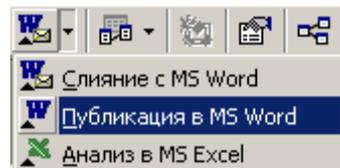
Экспорт объекта в формате .RTF. RTF – формат хранения текстовых документов. После экспорта данных из объекта базы в файл .RTF этот файл может быть открыт в MS Word. Для выполнения экспорта следует в окне базы данных выбрать объект (таблицу, запрос, форму или отчет) и в его контекстном меню выбрать пункт «Экспорт», выбрать в качестве типа файла «Файлы RTF» и задать имя файла. При экспорте таблицы, запросы и формы сохраняются в виде таблиц, а отчеты – в виде форматированных документов.

Пример. Выполним экспорт таблицы «Сотрудники» и откроем созданный файл в MS Word:

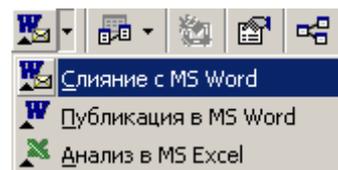
а	Табельный номер	Фамилия	Имя	Отчество
		Бобцов	Алексей	Алексеевич
		Бойков	Владимир	Иванович
		Болтунов	Геннадий	Иванович
		Болтунов	Иван	Павлович
		Бушуев	Александр	Борисович
		Быстров	Сергей	Владимиров
		Власенко	Владимир	Андреевич
		Григорьев	Валерий	Владимиров
		Коровьяков	Анатолий	Николаевич
		Котельников	Юрий	Петрович

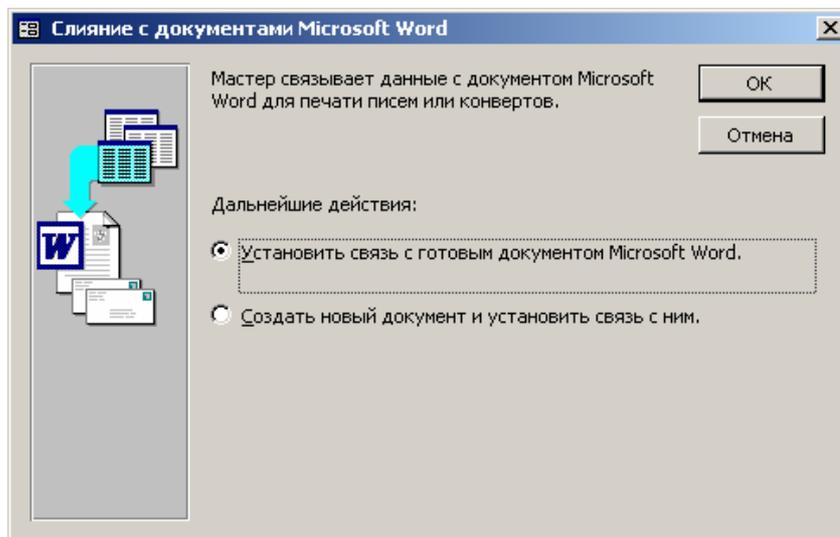
На основе таблицы Access сформировался текстовый документ, содержащий таблицу.

Публикация в MS Word. Для публикации объекта в MS Word следует выделить его мышью и выбрать пункт «Публикация в MS Word» в меню «Сервис→Связи с Office» или использовать соответствующую кнопку на панели инструментов. Публикация объекта в MS Word аналогична экспорту документа в формате .RTF, только при публикации для просмотра нового документа MS Word запускается автоматически.



Слияние с MS Word. Слияние с MS Word может быть выполнено на основе таблицы или запроса. Слияние позволяет выполнить подстановку данных из БД Access в созданный с помощью Word бланк документа. Для выполнения слияния следует выделить таблицу или запрос мышью и выбрать пункт «Публикация в MS Word» в меню «Сервис→Связи с Office» или использовать соответствующую кнопку на панели инструментов. При нажатии на кнопку вызывается мастер, предлагающий установить связь с уже существующим документом Word или создать новый.



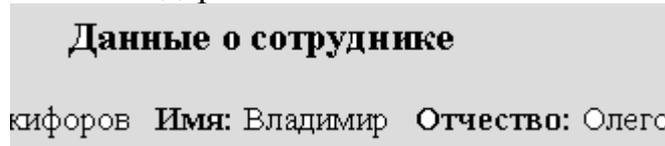


После выбора открывается документ MS Word, в котором доступна панель инструментов «Слияние».



Пункт «Добавить поле слияния»  позволяет разместить поля таблицы в нужном месте бланка документа.

Кнопка «Поля/данные»  переключает режим просмотра документа – видны имена полей:

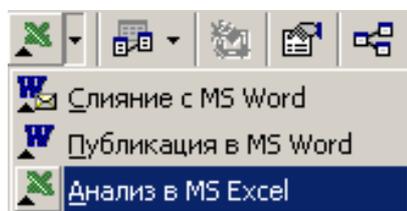


Кнопки навигации  позволяют сделать текущей другую запись, данные которой должны отображаться в бланке. Можно выбрать определенную запись и вывести документ на печать – будет распечатан бланк с подставленными в него данными из текущей записи. Составленный документ можно сохранить. При этом он остается связанным с БД и при последующем его использовании из БД будут подставляться актуальные (свежие) данные. Если требуется, чтобы создаваемый документ не зависел от базы данных, нужно объединить бланк и данные. Можно воспользоваться кнопками «Объединить»

 или «Слияние в новый документ» . При этом создается документ, в котором бланк тиражируется по количеству записей таблицы (если таблица БД содержит 100 записей, в созданном документе будет 100 разделов). При выполнении объединения можно также отобразить только те записи, которые нужно включить в результирующий документ. Процедура слияния может быть ини-

цирована также и со стороны MS Word. При этом нужно будет указать источник данных для слияния (базу данных Access) и выбрать таблицу или запрос.

Анализ в MS Excel. Операция «Анализ в MS Excel» позволяет поместить данные из выбранной таблицы, запроса, формы или отчета в документ Excel.



D	E	F	G
Фамилия	Имя	Отчество	Инд
Бобцов	Алексей	Алексеевич	
Бойков	Владимир	Иванович	
Болтунов	Геннадий	Иванович	
Болтунов	Иван	Павлович	
Бушуев	Александр	Борисович	

Действия, которые при этом следует выполнить, аналогичны публикации в MS Word.

3. Разработка базы данных

Выше, при изучении работы с готовой базой данных в первую очередь рассматривались формы, поскольку именно с ними пользователю приходится взаимодействовать чаще всего. Создание же БД начинается со структуры данных – с таблиц и связей, т.к. именно в таблицах хранятся все данные.

3.1. Таблицы и связи между ними

Создать таблицу – значит определить имена и типы данных для всех ее полей. В Access есть инструмент «Мастер», позволяющий создавать таблицы по предопределенным образцам. При запуске мастера предлагается выбрать наиболее подходящий образец («Контакты», «Сотрудники», «Поставщики», «Студенты» и т.п.), после чего нужно отобрать из списка имена необходимых полей (рис. 3.1). Очевидно, что невозможно предусмотреть образцы таблиц на все случаи жизни. Кроме того, не стоит полагаться на то, что создатели образцов имели в виду то же самое, что и вы.

Конструктор таблиц. Для создания и редактирования структуры любых таблиц в Access используется инструмент «Конструктор таблиц». Рассмотрим вид окна конструктора при редактировании структуры таблицы (рис. 3.2). В верхней части перечислены имена полей таблицы их типы данных. Поле «Код сотрудника» помечено слева символом , который означает, что поле является ключевым. В нижней части окна определяются дополнительные свойства для поля, которое выделено в данный момент в списке сверху.

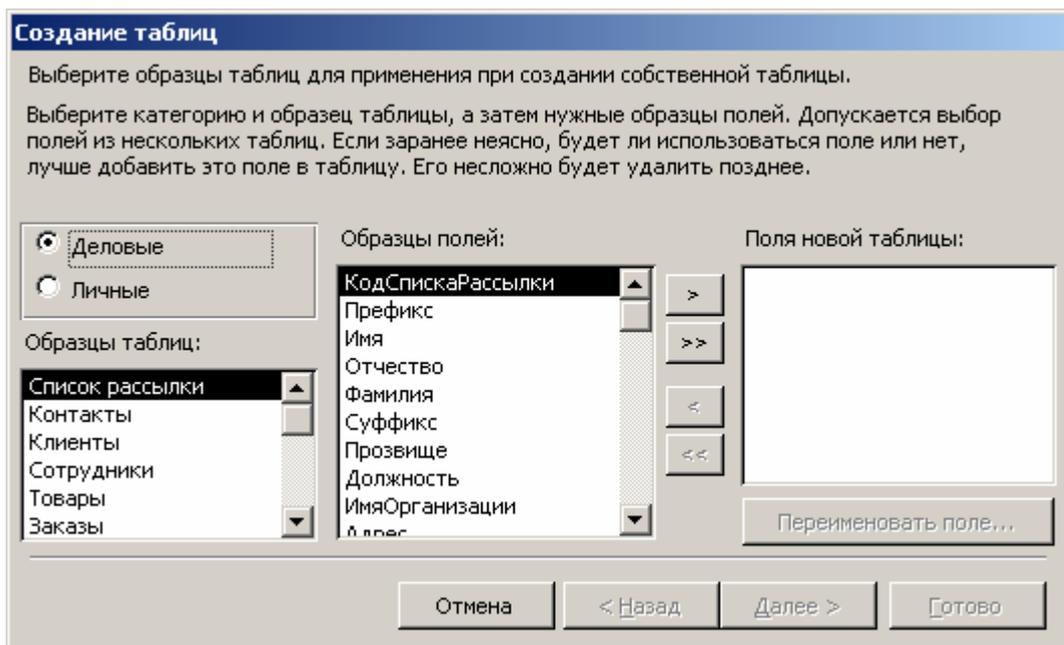


Рис. 3.1. Окно мастера создания таблиц

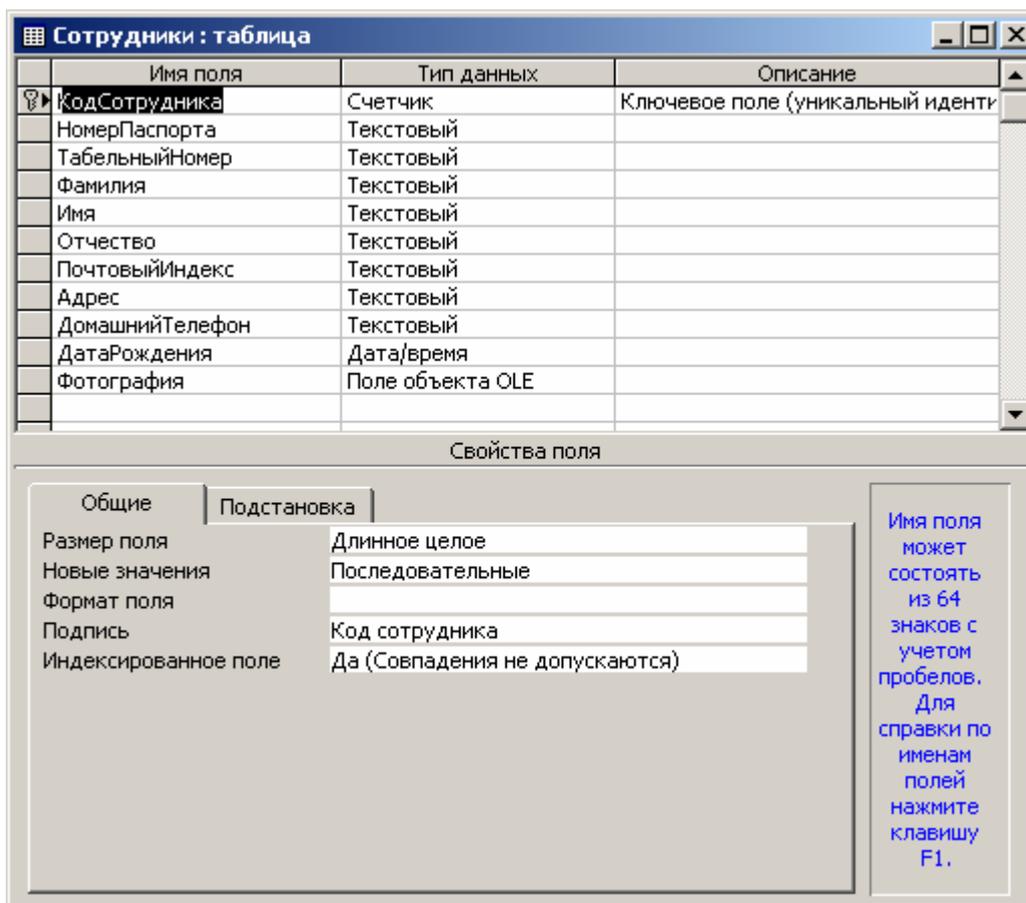


Рис. 3.2. Окно конструктора таблиц

При работе с таблицей можно выполнять переключение между двумя режимами – режимом конструктора и режимом таблицы. Для переключения используется кнопка на панели инструментов или соответствующие опции в разделе меню «Вид» (рис. 3.3). В режиме конструктора выполняется просмотр и редактирование структуры таблицы (названий и типов ее полей), а в режиме таблицы – просмотр и редактирование данных, хранящихся в этой таблице.

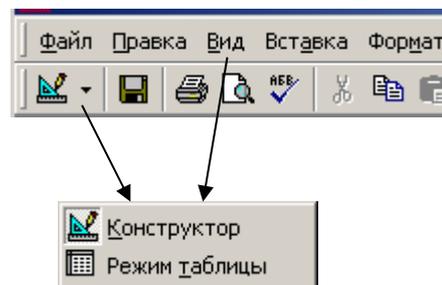


Рис. 3.3. Режимы работы с таблицей

Ниже перечислены типы данных Access, которые могут использоваться при определении полей таблиц:

- Текстовый (до 255 символов)
- Поле МЕМО (большой текст)
- Числовой (целые или вещественные значения)
- Дата/время
- Денежный
- Счетчик (для каждой добавляемой строки автоматически назначается уникальное целое значение)
- Логический (может принимать только два значения – “да” и “нет”)
- Поле объекта OLE (Object Linked Embedding) – некие объекты, для работы с которыми должно использоваться стороннее приложение. Например – графическое изображение или документ MS Word.

Тип данных определяет допустимое множество значений, которые могут быть размещены в поле. Кроме значений, специфичных для каждого типа данных, в Access (как и в большинстве других СУБД), существует специальное значение **Null**, универсальное для всех типов данных. Оно рассматривается как «никакое», «не присвоенное» значение. Если поле содержит значение Null, значит, оно не содержит никакого значения. Null отличается от значения «0» для числового поля и от значения «пустая строка» для текстового.

Для объявления поля ключевым, следует в режиме конструктора поместить курсор в строку, описывающую это поле и нажать на панели инструментов кнопку «Ключевое поле» . Объявление поля ключевым гарантирует поддержание уникальности его значений для каждой записи (при попытке внести повторяющиеся данные, фиксируется ошибка). Значения из ключевого поля таблицы могут использоваться в других таблицах для ссылки на конкретные записи (строки) этой таблицы.

Для заполнения ключевого поля гарантированно уникальными значениями может оказаться удобным использовать для него тип данных «Счетчик». При добавлении новой записи в таблицу, полю этого типа автоматически присваивается очередное числовое значение – на единицу больше предыдущего. Для определения составного ключа нужно выделить несколько полей, образующим ключ (при этом может потребоваться удерживать нажатой клавишу **Ctrl**). Обь-

явление составного ключа гарантирует поддержание уникальности комбинации значений полей, входящих в этот ключ.

В примере на рис. 3.4 ключ состоит из полей «НомерЭтажа» и «НомерКомнаты».

The screenshot shows a window titled 'Окно Справка' with a toolbar. Below it, a table structure is displayed with columns 'Имя поля' and 'Тип данных'. The fields are 'НомерЭтажа' (Числовой), 'НомерКомнаты' (Числовой), and 'Стоимость' (Личный). Below this, a data table is shown with columns 'НомерЭтажа', 'НомерКомнаты', and 'Стоимость'.

Имя поля	Тип данных
НомерЭтажа	Числовой
НомерКомнаты	Числовой
Стоимость	Личный

НомерЭтажа	НомерКомнаты	Стоимость
1	1	100,00
1	2	150,00
2	1	120,00
2	2	100,00

Рис. 3.4. Пример составного ключа

При этом несколько записей могут содержать один и тот же номер этажа и несколько – номер комнаты, но комбинация этих двух значений не может повторяться.

Задание свойств полей в конструкторе. Кроме выбора типа данных для полей, конструктор позволяет настраивать их свойства более детально. Вид списка свойств зависит от типа данных поля (рис. 3.5, 3.6).

The screenshot shows a 'Клиент : таблица' window with a table structure. The 'Свойства поля' (Field Properties) dialog box is open, showing a list of data types on the left and a list of properties on the right. The 'Длинное целое' (Long Integer) type is selected, and the 'Основной' (Basic) property is highlighted.

Имя поля	Тип данных
КодКлиента	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый

Свойства поля	
Общие	
Подстановка	
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)

Рис. 3.5. Свойства числового поля

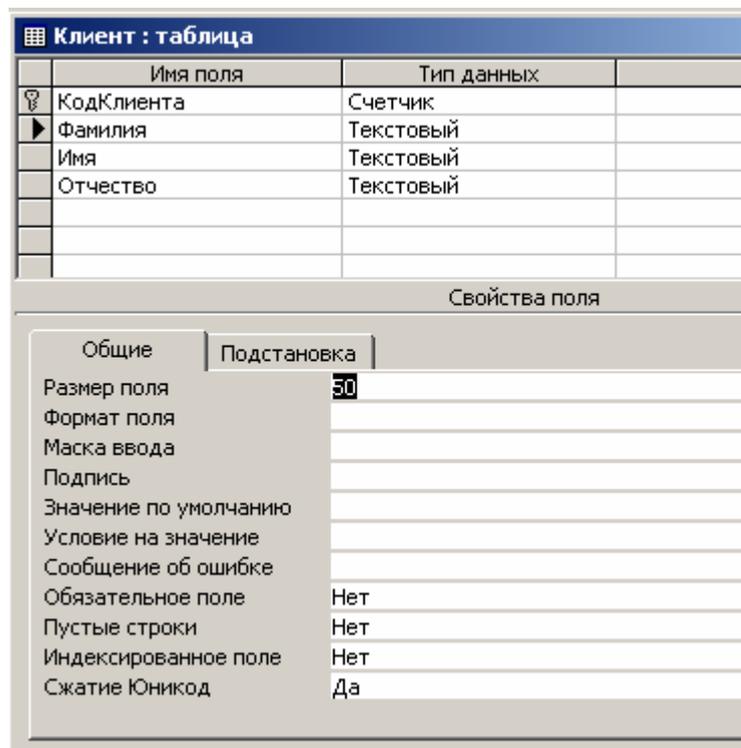


Рис. 3.6. Свойства текстового поля

Свойство «Размер поля» для числового поля уточняет тип числа (целое или вещественное) и размер места, отводимого под его хранение. Для текстового поля – это допустимое количество размещаемых в поле символов.

Свойство «Подпись» задает текст, который будет отображаться в качестве заголовка столбца таблицы. Если свойство не задано, в качестве заголовка будет использовано имя поля.

Свойство «Обязательное поле» определяет, допустимы ли в этом поле пустые значения (Null). Если свойство установлено в «Да», то поле не должно содержать значения Null.

Свойство «Значение по умолчанию» определяет значение, которое будет помещаться в поле новой записи, если при этом никакое другое значение не будет присвоено явно. Если свойство не задано, то значение по умолчанию – Null.

Свойство «Пустые строки» (для текстового поля) определяет, можно ли считать строку с нулевым числом символов строковым значением. В противном случае она будет рассматриваться как не присвоенное значение (Null).

Свойство «Индексированное поле» указывает, следует ли для данного поля строить индекс. Наличие индекса позволяет ускорить выборку и сортировку данных из этого поля, но может несколько замедлить их обновление. Кроме того, размещение индекса может вызвать увеличение размера файла БД. При определении индекса следует указать, допускаются ли в этом поле повторяющиеся значения. При объявлении поля ключевым, для него автоматически строится уникальный (без повторяющихся значений) индекс.

Вкладка «Подстановка» позволяет настраивать свойства столбцов таким образом, чтобы при просмотре таблицы отображались не те данные, которые хранятся в этих столбцах (если эти данные малоинформативны), а связанные с

ними данные из другого источника. Подробнее настройка подстановки рассматривается в практическом задании № 2.

Связи между таблицами. Для задания связей между таблицами в Access используется средство «Схема данных» (рис. 3.7).

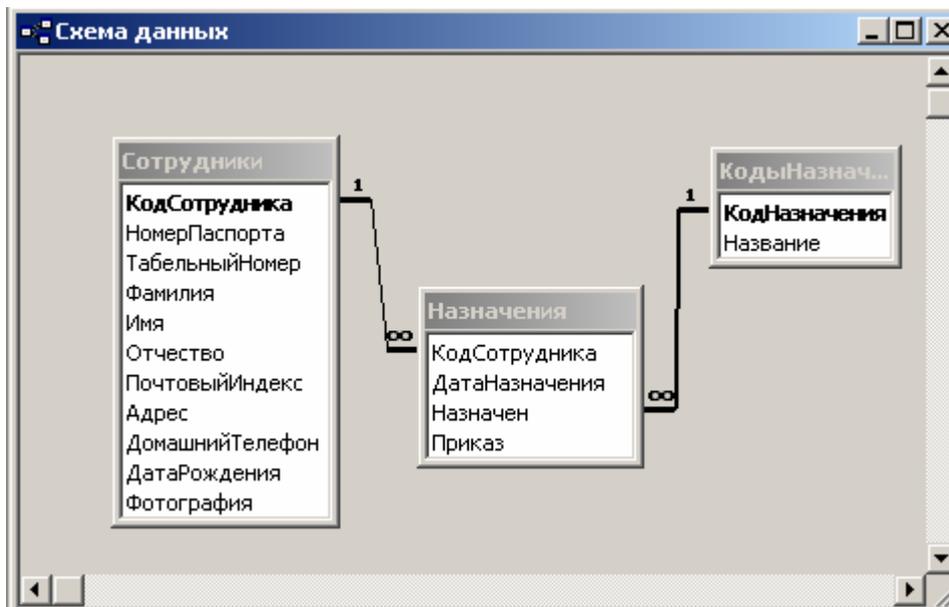


Рис. 3.7. Пример схемы данных

Схема данных – важный инструмент построения баз данных. Она позволяет наглядно представить связи между таблицами и обеспечить поддержание целостности данных в этих таблицах. Как будет видно в дальнейшем, явное задание связей между таблицами также облегчает построение запросов на основе этих таблиц.

Для открытия окна схемы данных следует нажать кнопку «Схема данных»  на панели инструментов или выбрать соответствующий пункт в разделе меню «Сервис». При первом обращении к схеме данных откроется диалоговое окно, в котором будет предложено выбрать таблицы для включения в схему (рис. 3.8).

В дальнейшем этот диалог можно вызвать, выбрав пункт «Добавить таблицу» в контекстном меню окна схемы данных или в разделе меню «Связи». Диалог не создает новые таблицы, а включает уже существующие в схему данных. В схеме данных отображаются названия таблиц и их полей. Ключевые поля выделены жирным шрифтом.

Для создания связи между таблицами, включенными в схему данных, следует с помощью мыши «перетащить» название одного связываемого поля на другое поле, с которым строится связь (рис. 3.9).

При этом появится диалоговое окно (рис. 3.10), в котором нужно задать требуемые свойства связи: обеспечение целостности данных, каскадное обновление и удаление, после чего нажать кнопку «Создать».

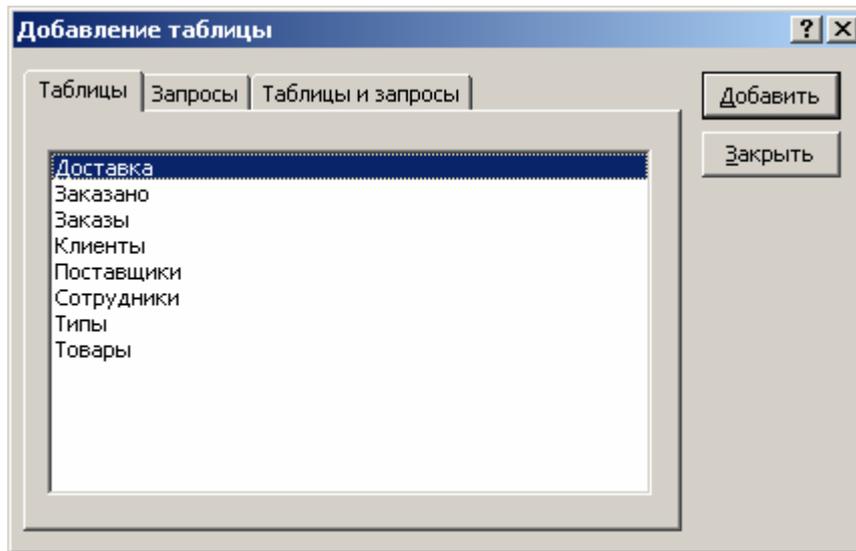


Рис. 3.8. Добавление таблиц в схему данных

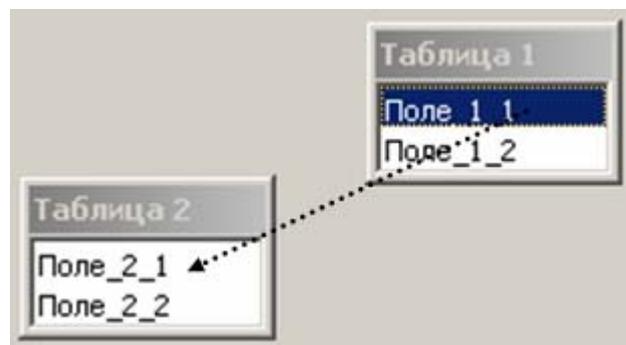


Рис. 3.9. Построение связи

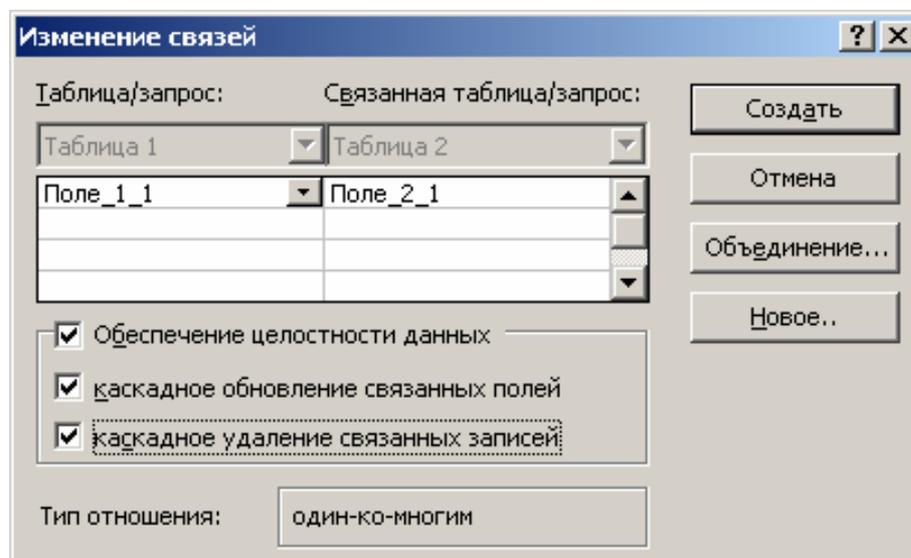


Рис. 3.10. Свойства связи

Вместо перетаскивания полей мышкой, можно вызвать тот же диалог, обратившись к пункту меню «Изменить связь» в разделе «Связи» или сделав двойной щелчок по незанятой области окна схемы данных. В этом случае нужно будет указать имена таблиц и полей, между которыми устанавливается связь.

Свойства ранее созданной связи также можно изменить, выделив ее и выбрав тот же пункт меню или просто щелкнув по ней мышкой.

При построении связи, Access называет одну из таблиц главной, а другую – связанной. Часто он сам может принять решение о том, какую из таблиц считать главной (например, если одно из связываемых полей является ключевым или просто имеет уникальный индекс, главной будет таблица, содержащая это поле). Если это не удастся, то главной считается таблица, с которой было начато прокладывание связи.

Диалог позволяет задать следующие свойства связи:

- **Обеспечение целостности данных.** Включение этого свойства запрещает помещать в поле связанной таблицы значения, которые отсутствуют в поле главной таблицы. При этом поле главной таблицы должно являться ключевым или, по крайней мере, уникальным. Т.е., значения в поле связанной таблицы будут однозначно указывать на конкретные записи в главной таблице.
- **Каскадное обновление связанных полей.** Если задано это свойство, при изменении значения поля в главной таблице будут автоматически меняться соответствующие значения в связанной таблице.
- **Каскадное удаление связанных записей.** Если удаляется запись из главной таблицы, должны автоматически удалиться связанные с ней записи из связанной таблицы.

Каскадное обновление и удаление могут быть заданы только в том случае, если включено обеспечение целостности данных. Если обеспечение целостности не задано, построенная связь никак не будет ограничивать множества допустимых значений связанных полей. В Access такие связи играют роль только при построении запросов (см. ниже). При создании связи между таблицами, в которых уже есть данные, Access проверяет условие целостности и отказывается строить связь, если это условие нарушено.

Построение связи с ограничением целостности создает **взаимоотношение** между таблицами. Если одно из связываемых полей является ключом или уникальным индексом (не допускает повторений), образуется связь вида «**один ко многим**» (рис. 3.11). Здесь «Таблица_1» - главная таблица, «Таблица_2» - связанная. Смысл этой связи заключается в следующем: Каждой записи из таблицы «Таблица_1» может соответствовать несколько записей (или ни одной) в таблице «Таблица_2». Каждой записи из «Таблица_2» может соответствовать только одна запись из «Таблица_1».

Если оба связываемых поля являются уникальными, связь будет иметь вид «**один к одному**» (рис. 3.12). При этом каждой записи в одной из таблиц соответствует только одна за-



Рис. 3.11. Отношение «Один ко многим»

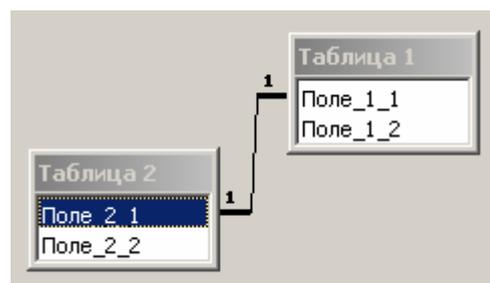


Рис. 3.12. Отношение «Один к одному»

пись в другой таблице. Связи такого вида используются нечасто, поскольку в этих случаях данные из двух таблиц обычно можно объединить в одну таблицу.

Связи, образующие взаимоотношения («один ко многим» или «один к одному»), требуют, чтобы связанное поле главной таблицы было ключевым, или, по крайней мере, уникальным, т.е. каждое значение из этого поля однозначно указывало на одну запись таблицы. Такое поле также принято называть **первичным ключом**. Поле связанной таблицы, которое содержит значения из ключевого поля главной таблицы, называют **внешним ключом**.

Пример. Схема данных на рис. 3.13 содержит три таблицы. Каждая из таблиц «Клиент» и «Журнал» является главной по отношению к таблице «Подписка». Поля «КодКлиента» и «КодЖурнала» являются первичными ключами таблиц «Клиент» и «Журнал» соответственно. Одноименные им поля в таблице «Подписка» являются внешними ключами благодаря наличию связей. При этом они образуют составной ключ (первичный) таблицы «Подписка» (т.е. любая пара значений КодКлиента/КодЖурнала встречается в таблице не больше одного раза).

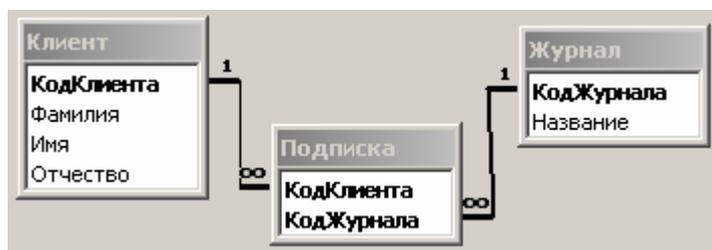


Рис. 3.13. Отношение «Многие ко многим»

На каждый журнал может быть подписано много клиентов. Любой клиент может быть подписан на множество журналов. Таким образом, между таблицами «Клиент» и «Журнал» образуется взаимоотношение вида «**многие ко многим**». Такое взаимоотношение не может быть задано напрямую, созданием одной связи между таблицами. Взаимоотношение «многие ко многим» образуется с помощью двух связей вида «один ко многим» и вспомогательной (связующей) таблицы. Пример на рис. 3.13 – лишь иллюстрация. Эта схема не учитывает срок, на который оформляется подписка и количество выписанных экземпляров.

3.2. Запросы

Запросы позволяют извлекать данные из одной или нескольких таблиц в соответствии с заданными условиями отбора, производить при этом обработку и сортировку данных. Кроме запросов на выборку, встречаются запросы на обновление, добавление и удаление данных, а также на создание таблиц. Наиболее часто применяются запросы на выборку (извлечение данных). Ниже будет рассмотрена задача создания именно таких запросов. Запрос на выборку должен иметь один или несколько источников данных. Источниками могут быть таблицы или другие запросы.

Общие принципы построения запросов. При выполнении запроса на основе нескольких источников, над данными из этих источников выполняются две операции: **умножение** и **сужение**. Первая строит все возможные комбинации из всех записей всех источников. Такое множество называется **декартовым произведением** источников. Вторая операция (сужение) отсекает «лишние» строки, используя заданные ограничения. Первоначальные ограничения образуются связями, определенными между источниками. При этом множество комбинаций всех записей «сужается» до множества комбинаций записей, связанных между собой. Дальнейшее сужение выполняется за счет ограничений, наложенных на значения выбранных полей.

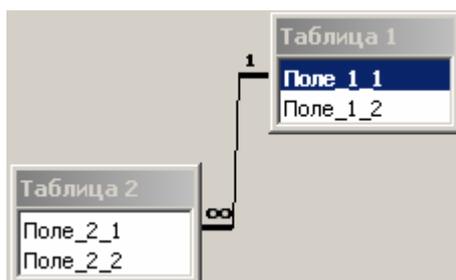


Рис. 3.14. Связанные таблицы

На рис. 3.14 приведены две связанные таблицы, на рис. 3.15 – данные, содержащиеся в этих таблицах. Линиями изображены связи между записями таблиц. Заметим, что первая запись левой таблицы связана с двумя записями правой.

Таблица 1 : таблица	
Поле_1_1	Поле_1_2
1	значение 1-1
2	значение 1-2
3	значение 1-3

Таблица 2 : таблица	
Поле_2_1	Поле_2_2
1	значение 2-1
2	значение 2-2
3	значение 2-3
4	значение 2-4

Рис. 3.15. Данные в таблицах

На рис. 3.16 слева показано декартово произведение таблиц (только вторые поля). Общее число строк в декартовом произведении равно произведению количества записей в каждой из таблиц. Справа – результат сужения (только комбинации записей, связанных между собой). Количество строк в этом наборе равно количеству пар связанных записей (см. линии на рис. 3.15).

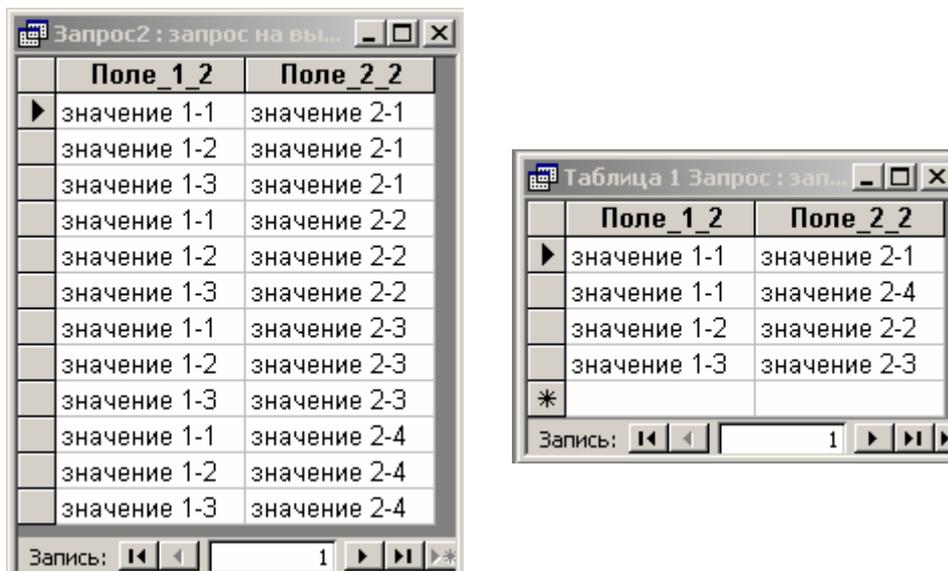


Рис. 3.16. Результаты запроса

Инструменты создания запросов. Для создания запросов в Access используются мастер или конструктор. Мастер (рис. 3.17) позволяет создавать несложные запросы на основе одного или нескольких источников данных. Мастер отказывается строить запросы при отсутствии связей между источниками.

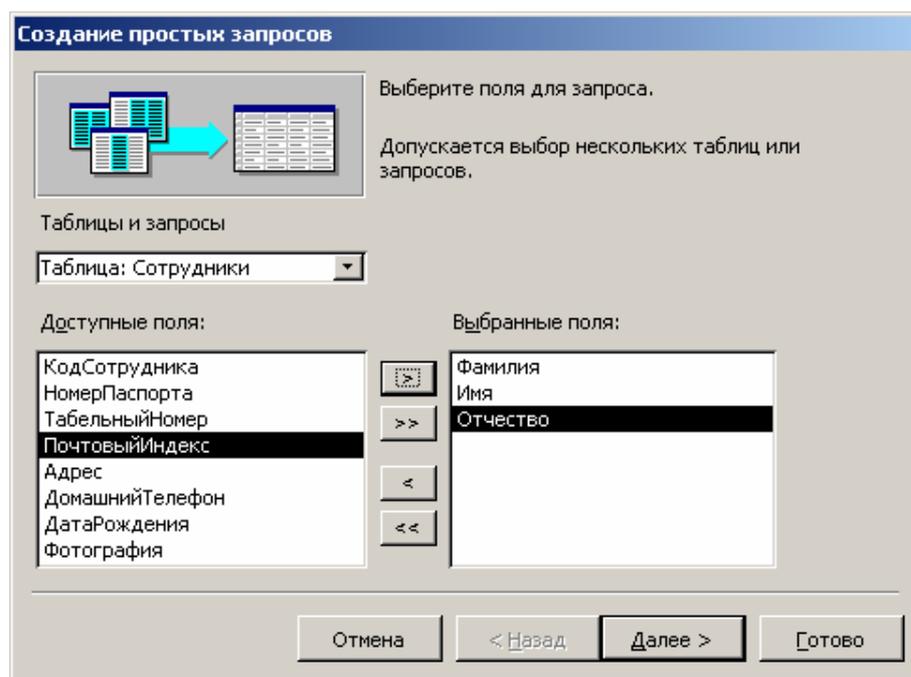


Рис. 3.17. Мастер запросов

Конструктор запросов (рис. 3.18) предоставляет значительно более широкие возможности. В верхней части окна конструктора находится область таблиц. В ней размещаются источники данных, участвующие в запросе. В нижней части – бланк запроса, в который заносятся выбираемые поля, указываются операции над ними (математические, логические, текстовые и т.п.) и задаются ограничения (условия отбора).

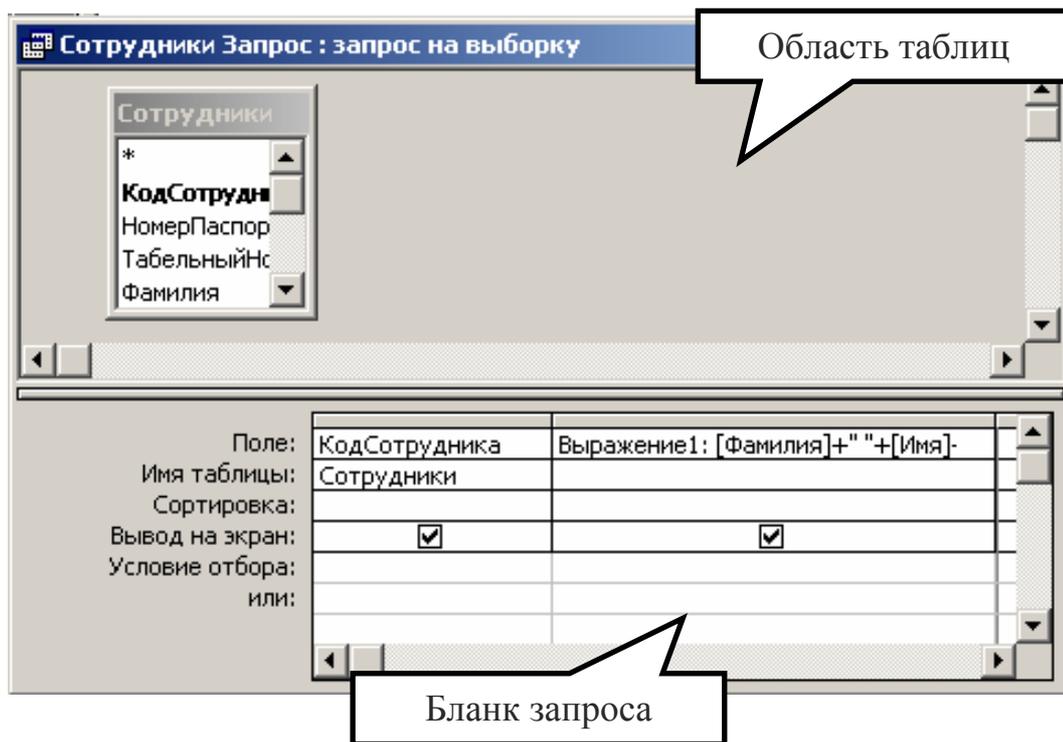


Рис. 3.18. Конструктор запросов

Пример 1. Рассмотрим процедуру построения запроса на примере. Пусть имеются три связанные таблицы (рис. 3.19).

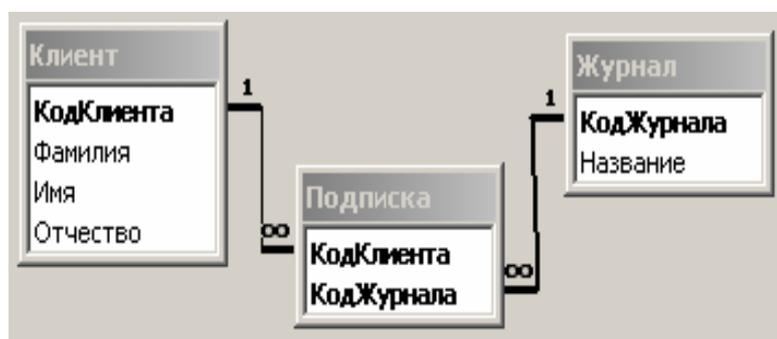


Рис. 3.19. Структура таблиц

На рис. 3.20 показаны данные, содержащиеся в этих таблицах.

Построим простой запрос на выборку: требуется вывести фамилии клиентов и названия журналов, на которые они подписаны. Выберем пункт «Создание запроса в режиме конструктора». Откроется диалоговое окно (рис. 3.21), предлагающее выбрать таблицы в качестве источников данных. Выберем таблицы «Клиент», «Подписка» и «Журнал», затем закроем окно со списком. Откроется окно конструктора (рис. 3.22). В области таблиц изображены таблицы и их связи. Перетащим мышкой поля «Фамилия» и «Название» в бланк запроса, как указывают стрелки.

Клиент : таблица

	КодКлиента	Фамилия	Имя	Отчество
+	1	Иванов	Иван	Иванович
+	2	Петров	Петр	Петрович
+	3	Николаев	Николай	Николаевич
+	4	Федоров	Федор	Федорович
	(Скрыть)			

Запись: 5 из 5

Журнал : таблица

	КодЖурнала	Название
+	1	Журнал 1
+	2	Журнал 2
+	3	Журнал 3
+	4	Журнал 4
	(Скрыть)	

Запись: 5

Подписка : таблица

	КодКлиента	КодЖурнала
	1	1
	1	3
	2	2
	3	1
	(Скрыть)	

Запись: 1

Рис. 3.20. Данные в таблицах

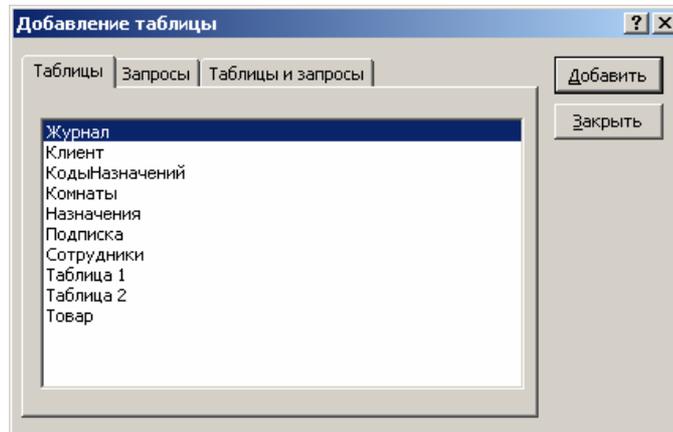


Рис. 3.21. Выбор источников данных

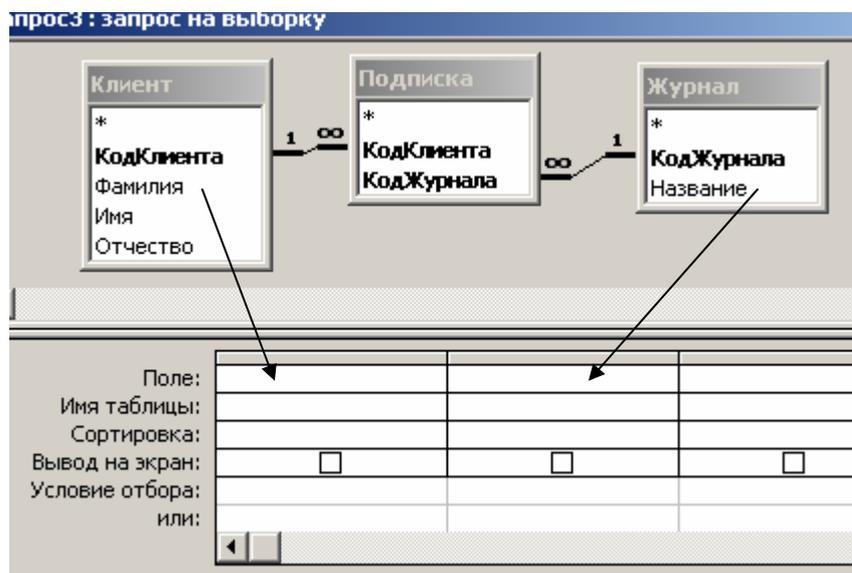


Рис. 3.22. Построение запроса в конструкторе

После этого бланк запроса будет выглядеть, как показано на рис. 3.23.

Поле:	Фамилия	Название	
Имя таблицы:	Клиент	Журнал	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Критерий отбора:			
или:			

Рис. 3.23. Бланк запроса с размещенными полями

Для просмотра результатов работы этого запроса, переключим запрос из режима конструктора в режим таблицы, нажав соответствующую кнопку  на панели инструментов Access. Будет показана таблица (рис. 3.24), содержащая извлеченные данные. При выполнении данного запроса из всех возможных комбинаций записей, содержащихся в трех таблицах, отображены лишь те, которые связаны между собой. Отображаются только поля, для которых в бланке запроса задано свойство «Вывод на экран».

Запрос3 : запрос на выборку		
	Фамилия	Название
▶	Иванов	Журнал 1
	Петров	Журнал 2
	Николаев	Журнал 1
	Иванов	Журнал 3
*		

Рис. 3.24. Результат работы запроса

Созданный запрос следует сохранить (). При первом сохранении откроется диалоговое окно с предложением задать название нового запроса. Сохраняется сам запрос, а не результат его работы. В дальнейшем сохраненный запрос можно вызывать неоднократно, при этом каждый раз результирующий набор будет формироваться заново на основе данных, которые будут находиться в таблицах на тот момент.

Пример 2. В запросах над данными можно выполнять и более сложные действия. Рассмотрим другой пример: требуется построить запрос, подсчитывающий количество журналов, на которые подписан каждый из клиентов. Сначала выберем пункт «Создание запроса в режиме конструктора», в качестве источников данных отберем таблицы «Клиент» и «Подписка» (таблица «Журнал» в данном случае не нужна, поскольку не нужны названия журналов, достаточно их кодов). Перенесем в бланк запроса поля «Фамилия» и «Коджурнала». Получившийся запрос показан на рис. 3.25. На рис. 3.26 можно видеть результат работы этого запроса (который пока далек от желаемого).

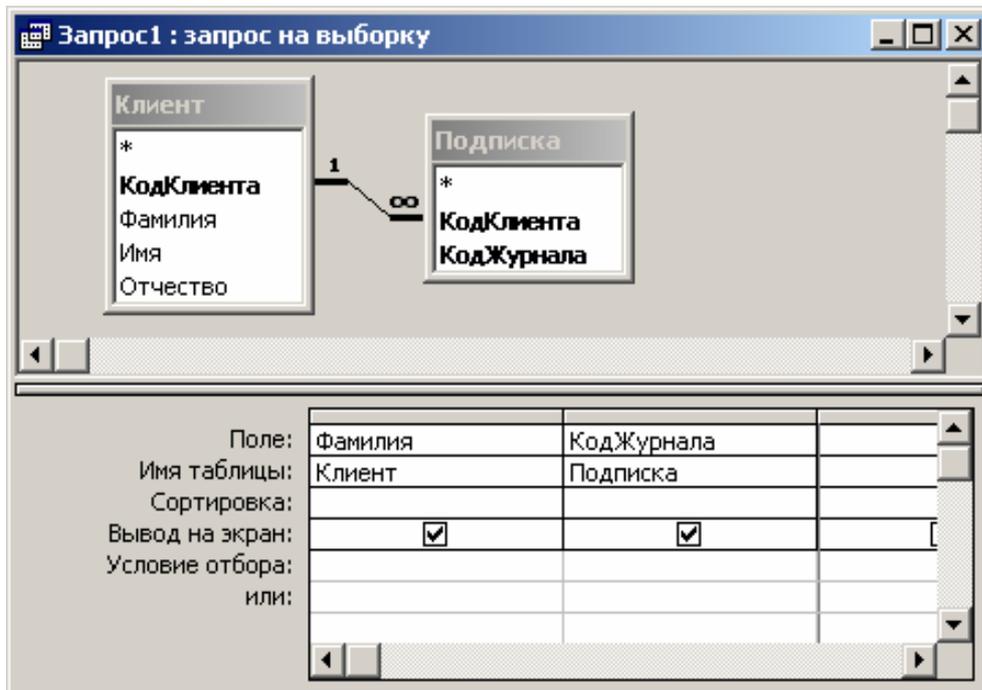


Рис. 3.25. Предварительный вариант запроса

	Фамилия	КодЖурнала
▶	Иванов	1
	Иванов	3
	Петров	2
	Николаев	1
*		

Запись: 1

Рис. 3.26. Результат работы запроса

Запрос выводит фамилии клиентов и коды журналов, на которые эти клиенты подписаны (так, клиент Иванов подписан на журналы с кодами 1 и 3). Для решения поставленной задачи нужно сгруппировать строки с одинаковыми фамилиями и для каждой фамилии подсчитать количество строк в этой группе. Вернувшись в режим конструктора (), нужно разрешить использование в запросе **групповых операций** нажатием соответствующей кнопки Σ на панели инструментов. При этом в бланке запроса появится строка «Групповая операция» (рис. 3.27). Для поля «Фамилия» следует выбрать из выпадающего списка групповую операцию «Группировка», а для поля «КодЖурнала» - операцию «Count» (подсчет количества). Заодно добавим сортировку по фамилии.

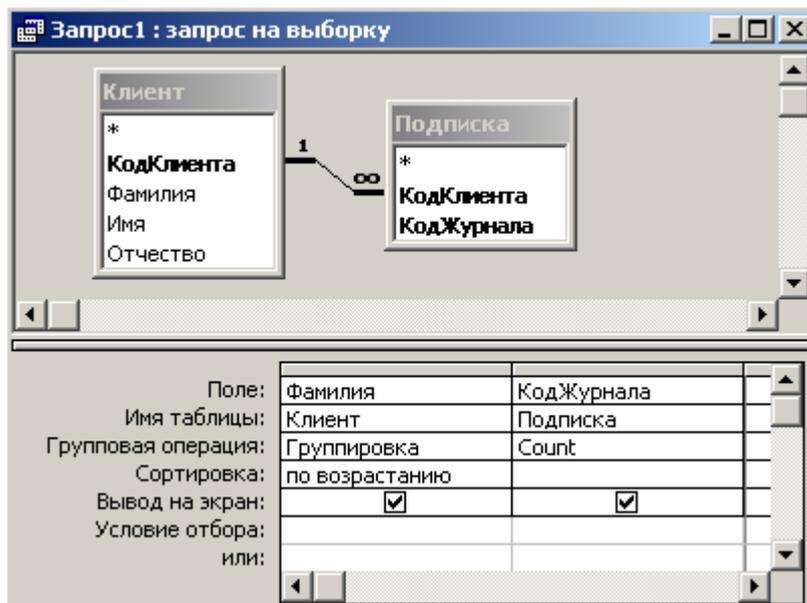


Рис. 3.27. Групповые операции в запросе

На рис. 3.28 приведен результат работы этого запроса. В нем отображаются фамилии только тех клиентов, которые подписаны хотя бы на один журнал.

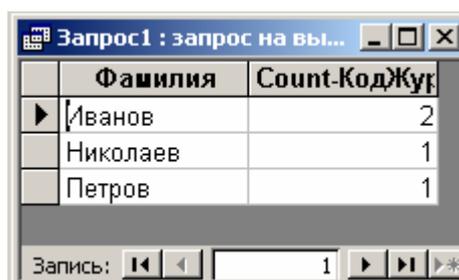


Рис. 3.28. Результат работы измененного запроса

Причина этого – в способе объединения данных. В результирующий набор попадают записи из обеих таблиц, связанные между собой. Поскольку клиент Федоров не связан ни с одной записью в таблице «Подписка», он здесь не показан. Для того чтобы изменить параметры объединения таблиц, нужно щелкнуть мышкой по линии связи между этими таблицами (в области таблиц конструктора запросов). В открывшемся диалоговом окне (рис. 3.29) выберем второй вариант (объединение всех записей из «Клиент» и только тех записей из «Подписка», в которых связанные поля совпадают).

На рис. 3.30 показано, как будет работать запрос после изменения параметров объединения. В созданный запрос можно внести некоторое «эстетическое» улучшение. Как показано на рис. 3.30, в качестве заголовка второго столбца отображается строка «Count-КодЖурнала», сформированная Access автоматически. Чтобы задать «правильный» заголовок, внесем изменение в бланк запроса:

Фамилия	Количество: КодЖурнала
---------	------------------------

(сравните с рис. 3.27).

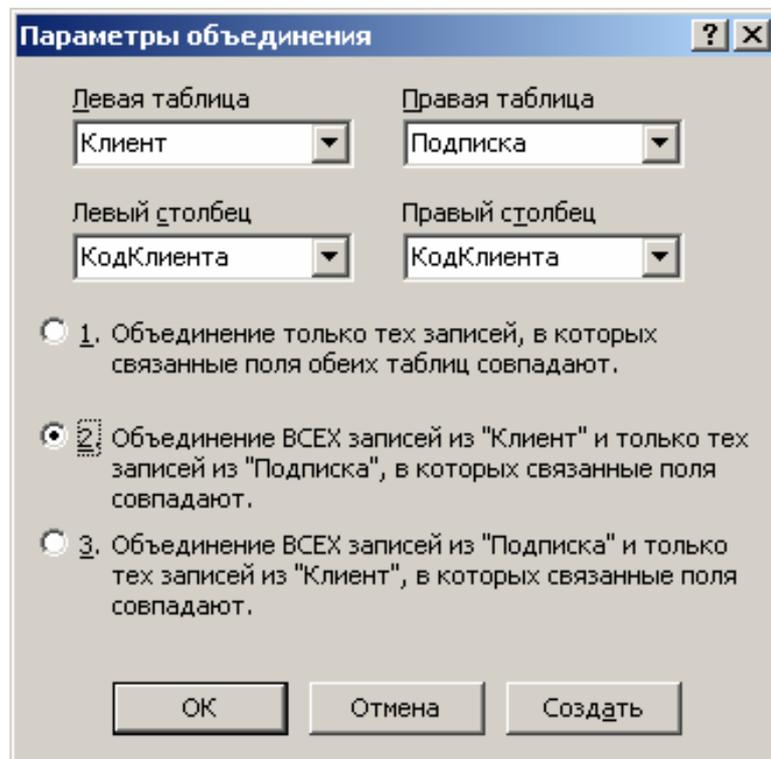


Рис. 3.29. Параметры объединения таблиц

Фамилия	Count-КодЖур
Иванов	2
Николаев	1
Петров	1
Федоров	0

Рис. 3.30. Результат работы окончательной версии запроса

В приведенном примере были затронуты такие вопросы как типы объединений таблиц и групповые операции. Рассмотрим их более подробно.

Типы объединений таблиц. Способ объединения двух связанных таблиц, при котором в результирующий набор поступают только связанные между собой записи обеих таблиц, называют **внутренним объединением**. Этот способ используется по умолчанию. В предыдущем примере было использовано **внешнее левое объединение**, при котором извлекаются все записи из первой из объединяемых таблиц и только связанные с ними записи из второй таблицы. Если для записи первой таблицы нет соответствия во второй, соответствующие поля будут заполнены «не присвоенным значением» (Null).

Внешнее правое объединение симметрично внешнему левому. При таком способе в результирующий набор будут включены все записи из второй объединяемой таблицы и только связанные с ними записи из первой.

Групповые операции. Групповые операции предназначены для статистической обработки данных. Имеется в виду, что одна строка результирующего

набора может быть сформирована на основе нескольких строк исходных данных. Иначе говоря, группа исходных строк порождает одну результирующую строку. Группа формируется на основе равенства значений некоторых полей. Для всех полей, формирующих группу, должна быть задана операция «Группировка». Поскольку значения в этих полях равны между собой, они могут быть представлены единственной строкой. Значения в остальных полях могут не совпадать и для того, чтобы из наборов значений каждого столбца сформировать единственное результирующее значение, над этими столбцами выполняются некоторые групповые операции (например, вычисление суммы или среднего арифметического).

Ниже перечислены основные групповые операции, используемые в Access:

- **Sum** – сумма значений группы
- **Avg** – среднее арифметическое
- **Min** – минимальное значение
- **Max** – максимальное значение
- **Count** – количество элементов в группе
- **First** – первое значение из группы
- **Last** – последнее значение из группы

Кроме того, есть две специальные групповые операции, применение которых будет показано ниже:

- **Условие**
- **Выражение**

Условия отбора результатов. Состав данных, входящих в результирующий набор, может быть ограничен путем задания дополнительных условий отбора. Условия помещаются в специально отведенные поля в бланке запроса. На рис. 3.31 показан пример запроса с условием отбора (на основе запроса, составленного ранее) и результат работы этого запроса (отображается только клиент с фамилией «Иванов»).

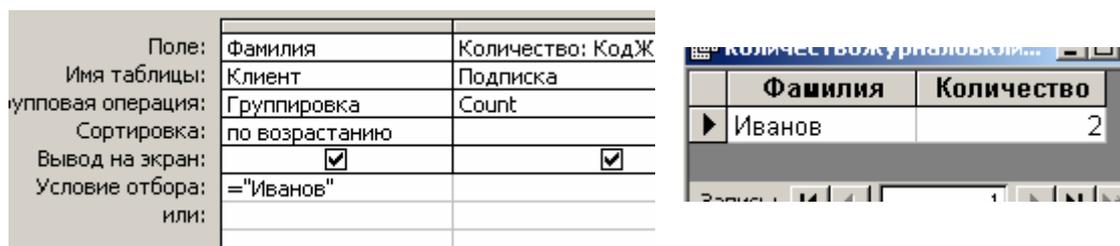


Рис. 3.31. Запрос с условием отбора

В бланке запроса можно задавать несколько условий. При этом условия, размещенные в разных столбцах одной строки объединяются операцией «И» (одновременно должны выполняться условия, заданные в каждом из столбцов). Условия, размещенные в разных строках, объединяются операцией «Или» (должны соблюдаться условия хотя бы в одной из строк). В запросах, использующих групповые операции, задача определения условий отбора становится двусмысленной: можно применить ограничения до выполнения группировки

или после нее – результат будет различным. Рассмотрим пример запроса на рис. 3.32. Словами его можно переформулировать так: сформировать список клиентов, подсчитав для каждого количество выписанных журналов. Вывести в результирующий набор строки, в которых количество равно «1».

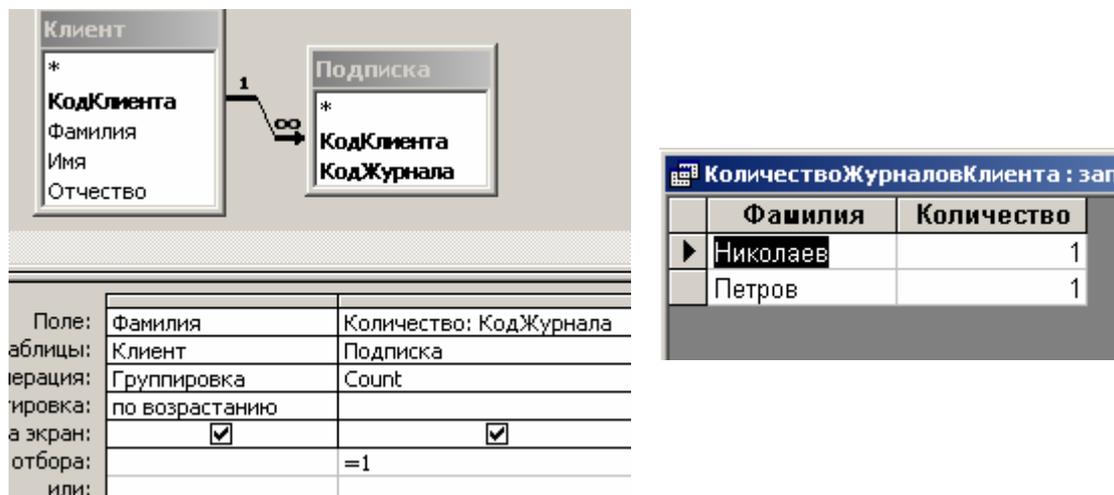


Рис. 3.32. Условие отбора, применяемое после группировки

Если требуется до выполнения группировки наложить ограничения на значение кода журнала, условие отбора нужно задать, как показано на рис. 3.33: В бланк запроса помещается еще один столбец (сформированный из того поля, по которому нужно выполнить отбор) и для этого столбца задается групповая операция «Условие». Сам этот столбец не должен отображаться в результирующем наборе (флажок «Вывод на экран» сброшен). В таком виде запрос имеет следующий смысл: сформировать список клиентов, подписанных на журнал с кодом «1» и подсчитать для них количество выписанных журналов.

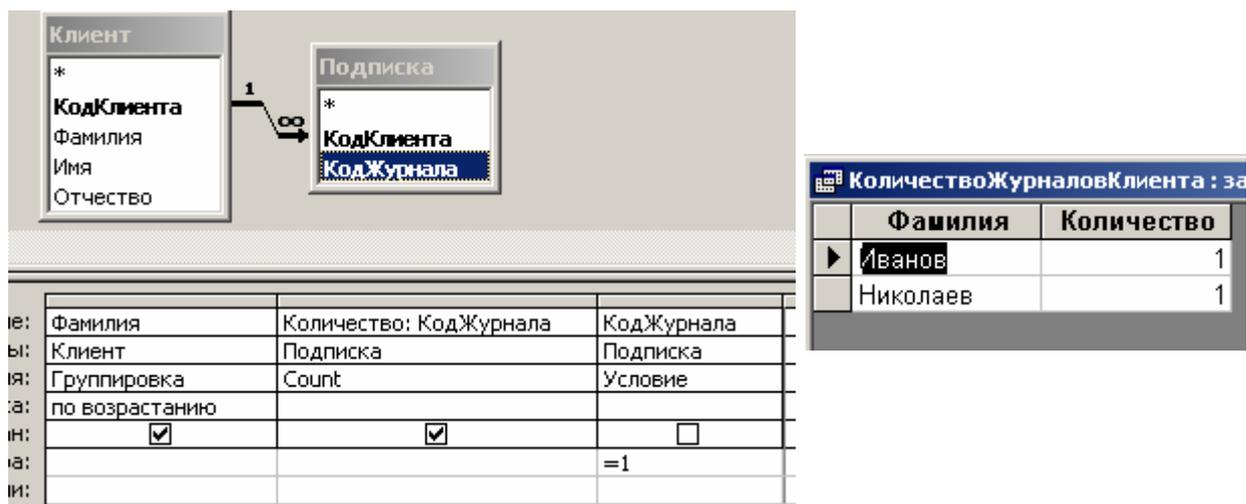


Рис. 3.33. Условие отбора, применяемое до группировки

Использование выражений в запросах. При формировании результирующих наборов могут использоваться не только поля таблиц, но и построенные на их основе выражения. Выражения следует вписывать в строку «Поле»

бланка запроса. На рис. 3.34 показан запрос, формирующий список клиентов с их инициалами и результат его работы. В запросе используется операция сложения строк и функция Left(). Для избегания неоднозначного толкования выражений, имена полей заключаются в квадратные скобки. Ниже перечислены некоторые функции, которые могут быть полезны при построении запросов. Следует иметь в виду, что при использовании функций в конструкторе запросов их аргументы отделяются не запятой, а точкой с запятой.

Функции обработки текста:

- **Left**(строка, n) – возвращает **n** левых символов строки.
- **Right**(строка, n) – возвращает **n** правых символов строки.
- **Mid**(строка, n1, n2) – возвращает **n2** символов строки, начиная с позиции **n1**.
- **InStr**(строка1, строка2) – номер позиции, с которой **строка2** входит в **строка1**.
- **Ltrim**(строка), **Rtrim**(строка) – удаляют пробелы из начала и конца строки соответственно.
- **Trim**(строка) – удаляет пробелы из начала и конца строки.

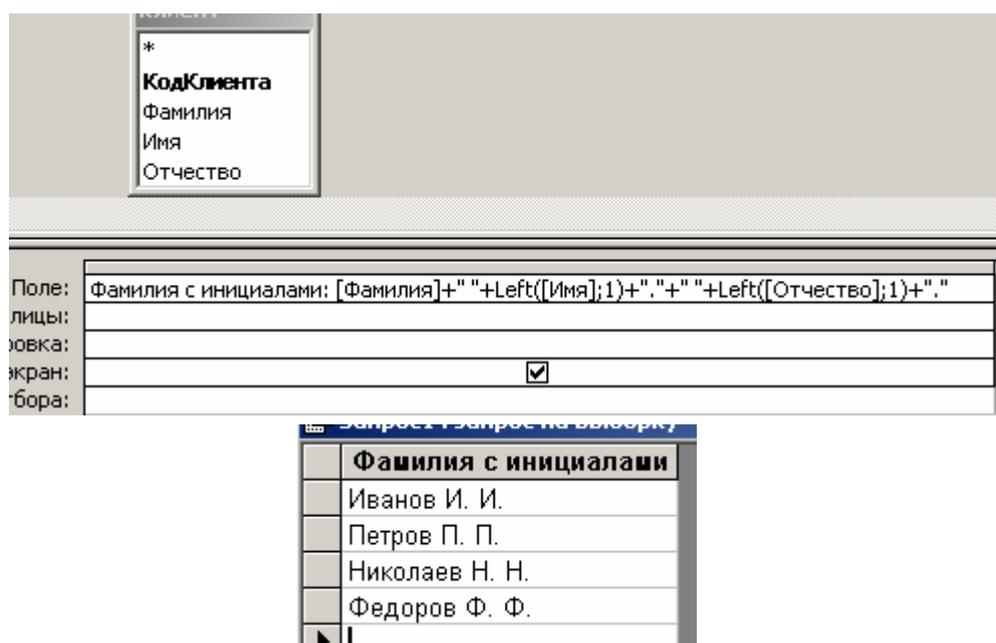


Рис. 3.34. Выражение в запросе

Функции обработки даты и времени:

- **Date**() – возвращает текущую дату.
- **Now**() – возвращает текущую дату и время.
- **DateDiff**(интервал, дата1, дата2) – определяет разницу между датами. Аргумент **интервал** задает способ представления разницы: “уууу” – год, “q” – квартал, “m” – месяц, “y” – день года, “d” – день, “w” – неделя, “h” – час, “n” – минута, “s” – секунда
- **DateAdd**(интервал, число, дата) – будущая дата, отстоящая от указанной на заданное число интервалов.

- **Year**(дата), **Month**(дата), **Day**(дата) – возвращают число - значение года, месяца и дня для указанной даты.

Функции преобразования типов:

- **Str**(аргумент) – преобразует значение аргумента в текстовую строку
- **Val**(строка) – преобразует строку в число
- **Int**(число) – возвращает целую часть числа

Функции преобразования нужны, если выполняются операции над данными разных типов. Например, если требуется объединить фамилию и дату в одну строку, дата должна быть сначала преобразована в текстовую переменную.

Условная функция: If(выражение, если истинно, если ложно) – вычисляет значение аргумента **выражение**. Если значение истинно, возвращает значение второго аргумента, если ложно – значение третьего аргумента.

Пример: If([количество]>0, “Есть в наличии”, “Отсутствует”)

Если выражение, указанное в строке «Поле» бланка запроса должно играть роль групповой (статистической) операции в запросе с группировкой, в строке «Групповая операция» бланка запроса следует вместо конкретной операции выбрать пункт «Выражение» (этот специальный тип групповой операции упоминался выше). При построении запроса выражения и функции могут быть использованы не только в строке «Поле» бланка конструктора, но и в строке «Условие отбора». Кроме того, объекты базы данных «Модули» позволяют пользователю определять собственные функции, которые могут быть использованы наравне со встроенными.

В некоторых случаях может оказаться полезным входящий в Access построитель выражений – инструмент, помогающий создавать выражения, используя объекты БД, операторы, встроенные и пользовательские функции.

Для запуска построителя выражений следует в контекстном меню соответствующей строки бланка запроса выбрать пункт «**Построить**» (рис. 3.35).

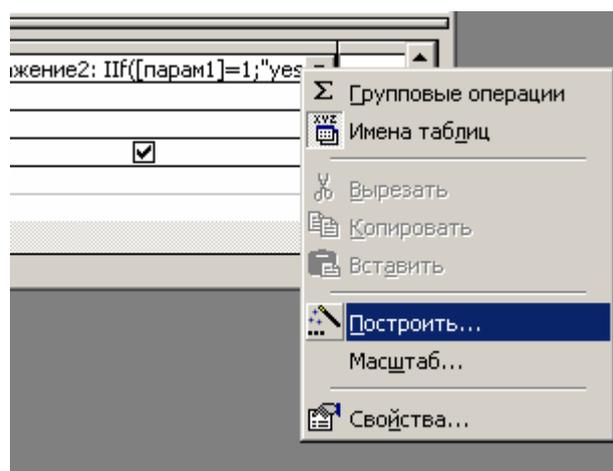


Рис. 3.35. Запуск построителя выражений

Вид окна построителя выражений показан на рис. 3.36.

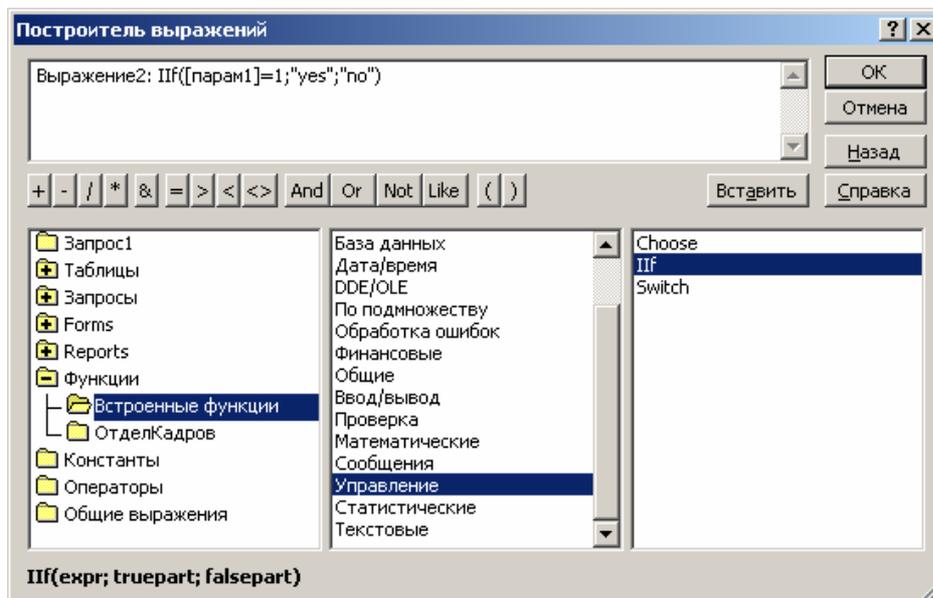


Рис. 3.36. Окно построителя выражений

Параметры запросов. В запросах могут присутствовать параметры. Они используются для передачи в запрос некоторых данных в момент его выполнения. Один и тот же запрос, выполненный с разными значениями параметров, может создавать различные результирующие наборы. Каждый параметр запроса должен иметь уникальное (в пределах запроса) имя. Имя параметра участвует в выражениях, входящих в запрос. При вызове запроса на выполнение открывается диалоговое окно с предложением указать значения его параметров. Для просмотра и изменения списка параметров следует выбрать пункт «Параметры» в контекстном меню окна конструктора (рис. 3.37). При этом будет открыто диалоговое окно со списком параметров запроса.

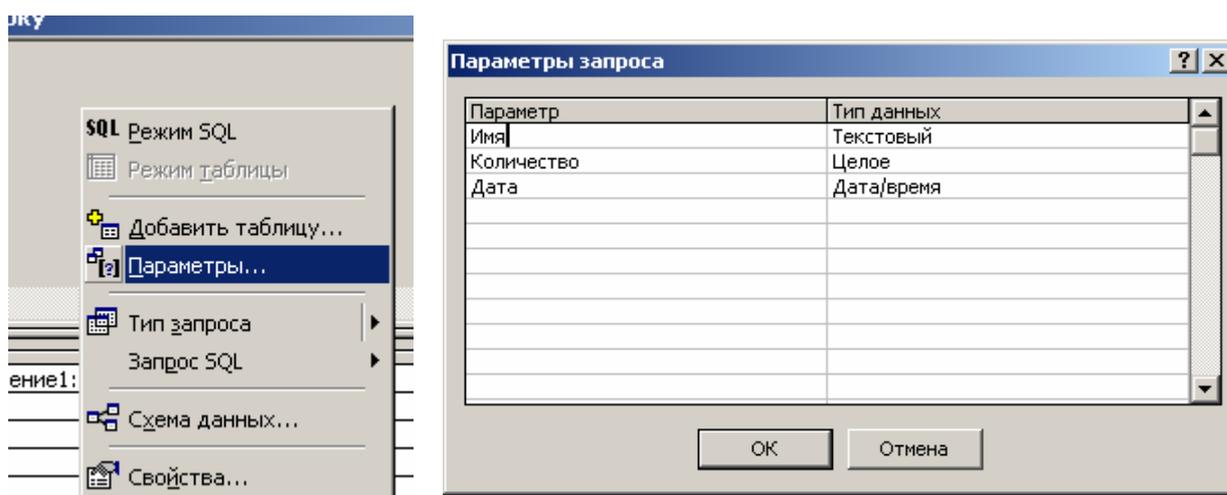


Рис. 3.37. Редактирование списка параметров запроса

В некоторых случаях нет необходимости задавать имя параметра явно. Если при запуске запроса в его теле встречается некоторое неизвестное имя, оно автоматически считается именем параметра и Access выдает окно с предложением ввести его значение. На рис. 3.38 приведен пример запроса, содержащего

параметр с именем «Введите фамилию» (имя параметра заключено в квадратные скобки). Это имя воспринимается как параметр, поскольку полей с таким именем в выбранных таблицах не существует.

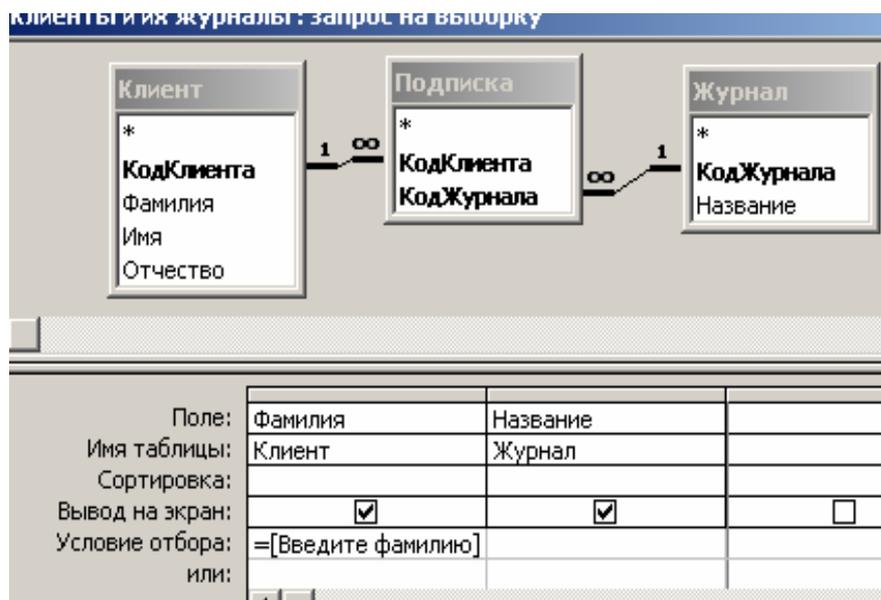


Рис. 3.38. Пример запроса с параметром

При запуске этого запроса на выполнение будет выдано окно для ввода значения этого параметра. Формируемый результирующий набор зависит от введенного значения (рис. 3.39).

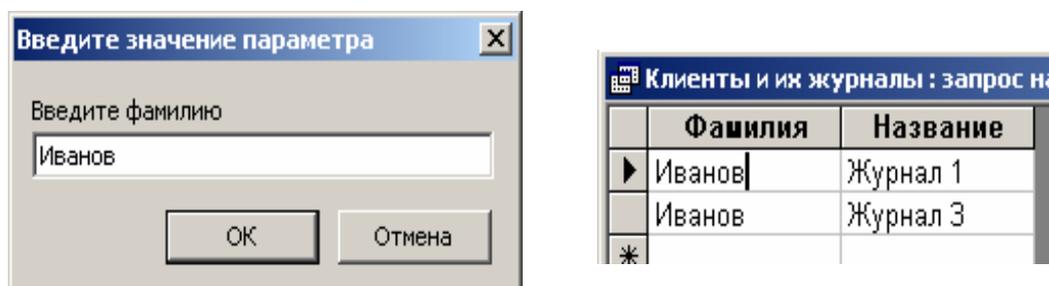


Рис. 3.39. Выполнение запроса с параметром

Перекрестные запросы. Перекрестный запрос в некоторых случаях позволяет отобразить данные в удобной для анализа форме. Для построения перекрестного запроса, данные предварительно должны быть приведены к определенному виду: следует сформировать три колонки данных. Перекрестный запрос будет использовать данные из одной колонки в качестве названий строк таблицы, другой – в качестве названий столбцов, а данные из третьей колонки будут размещены в таблице на пересечении соответствующих строк и столбцов. На рис. 3.40 показано, как выглядят запрос на выборку и перекрестный запрос, построенные на основе одних и тех же данных.

Запрос1 : запрос на выборку			
	Фамилия	Название	КодЖурнала
▶	Иванов	Журнал 1	1
	Иванов	Журнал 3	3
	Николаев	Журнал 1	1
	Петров	Журнал 2	2
*			

Запрос1 : перекрестный запрос				
	Фамилия	Журнал 1	Журнал 2	Журнал 3
▶	Иванов	1		1
	Николаев	1		
	Петров		1	

Рис. 3.40. Сравнение запроса на выборку и перекрестного

Соответствующий запрос на выборку показан на рис. 3.41.

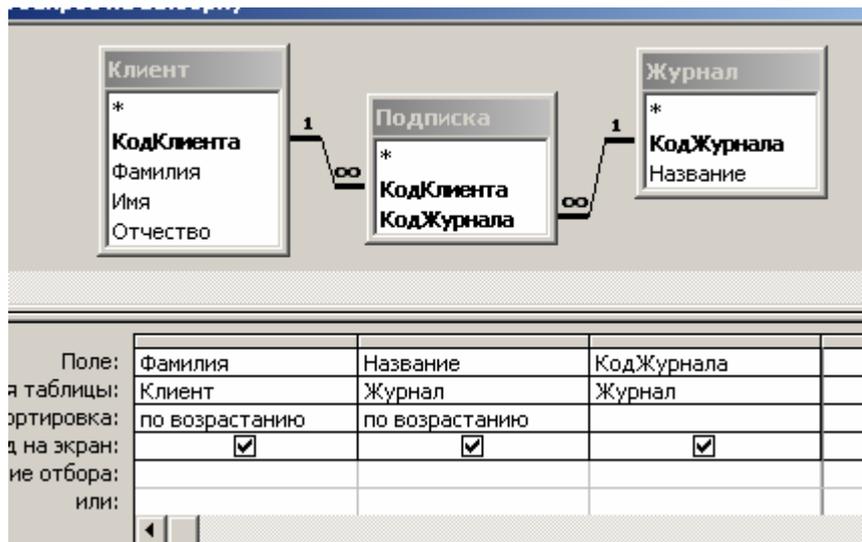


Рис. 3.41. Запрос на выборку

Рассмотрим, какие действия нужно проделать для превращения этого запроса в перекрестный. Сначала изменим тип запроса. Для этого в контекстном меню конструктора выберем пункт «Тип запроса -> Перекрестный» (рис. 3.42).

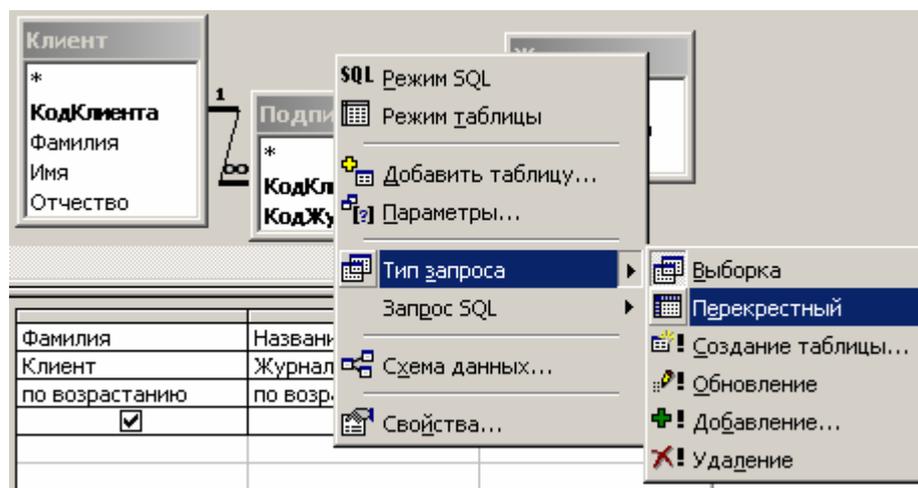


Рис. 3.42. Изменение типа запроса на перекрестный

В бланке запроса появятся новые строки «Групповая операция» и «Перекрестная таблица». Для первых двух столбцов выберем групповую операцию «группировка» (чтобы не повторялись названия строк и столбцов), для третьего – «Count» (подсчет количества). Укажем, что значения поля «Фамилия» (первый столбец) образуют заголовки строк, поля «Название» - заголовки столбцов, а поля «Код журнала» - значения внутри таблицы (рис. 3.43).

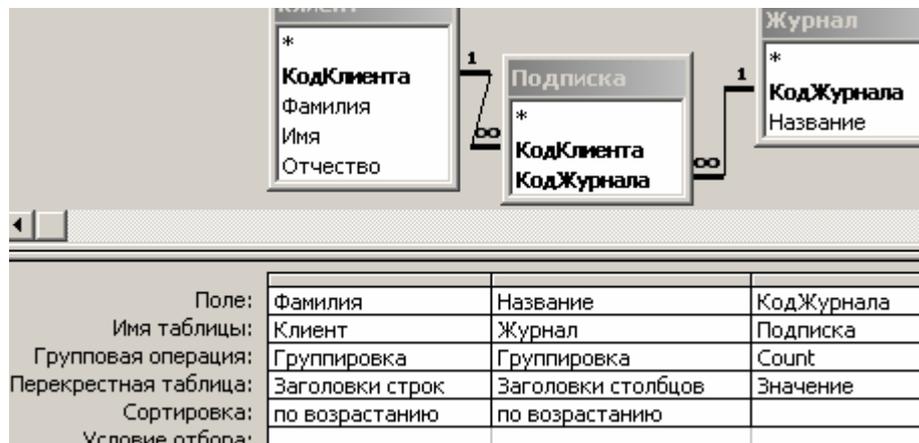


Рис. 3.43. Перекрестный запрос

Все, перекрестный запрос создан. Результат его работы был показан ранее на рис. 3.40. В ячейках таблицы отображается количество подписок, оформленных клиентом (указанным в названии строки) на журнал (указанный в названии столбца). Поскольку в таблице «Подписка» поля «КодКлиента» и «КодЖурнала» образуют первичный ключ (их комбинации не могут повторяться внутри таблицы), значения, отображаемые в созданном перекрестном запросе, должны быть равны только «1». Впрочем, это правило может быть нарушено, если у нескольких клиентов будут одинаковые фамилии, или у нескольких журналов – одинаковые названия. Внесем в созданный запрос некоторое «эстетическое» усовершенствование для демонстрации применения групповой операции «**Выражение**» и функции **IIf()**. Пусть в ячейках таблицы отображается не количество подписок клиента на журнал, а знак «+», если это количество больше нуля. Заменим в третьей колонке бланка запроса групповую операцию «Count» на «Выражение» и в строке «Поле» запишем следующее выражение:

IIf(Count(Подписка.КодЖурнала) > 0; "+" ; "")

Функция **IIf()** будет возвращать знак «+» для групп, для которых значение, возвращаемое функцией **Count()**, больше «0». Для остальных групп возвращается пустая строка. Функция **Count()**, как и раньше, присутствует в запросе, но теперь она является частью более сложного выражения. На рис. 3.44 показаны новый вид запроса и результат его работы. «ДаНет» - имя поля, содержащего выражение. Оно не играет особой роли, поскольку нигде не отображается.

Некоторые из приведенных выше примеров запросов были упрощены для большей наглядности и могут быть непригодны для использования в реальных базах данных.

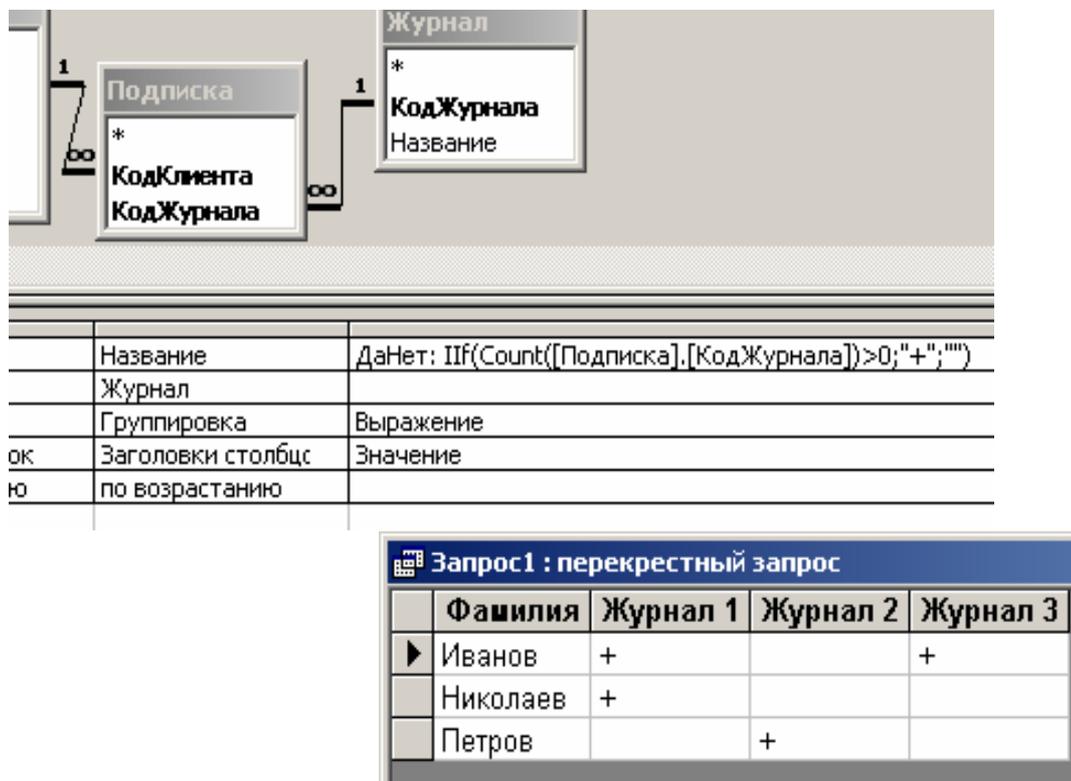


Рис. 3.44. Измененный запрос

3.3. Формы и отчеты

Инструменты создания форм. Для создания форм в Access существуют следующие инструменты:

- **Мастер форм.** Позволяет создавать несложные формы на основе одной или нескольких связанных таблиц или запросов.
- Автоформа.** Средство автоматического создания форм. Может создавать простые формы нескольких разновидностей:
 - **Форма в столбец** - одновременно на экран выводятся данные одной записи таблицы или запроса, расположенные в столбец;

- **Ленточная форма** - одновременно на экран выводятся данные нескольких записей, каждая запись занимает одну строку;

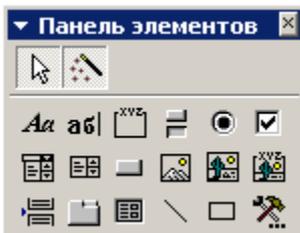
- **Табличная форма** - данные отображаются в виде таблицы.

- **Конструктор.** Позволяет вручную создавать и редактировать любые формы. Иногда удобно сперва создать форму в виде автоформы или с помощью мастера, а затем «довести до ума» конструктором.

Конструктор форм. Конструктор форм включает в себя следующие средства:

- **Бланк формы.** В нем располагаются элементы, предназначенные для отображения в форме. В основном, именно в бланке формируется внешний вид формы.

- **Панель элементов.** Содержит стандартный набор визуальных элементов, которые могут быть помещены в бланк формы.



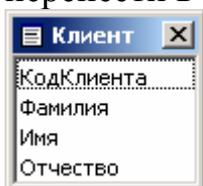
- **Редактор свойств.** Позволяет задавать свойства формы и размещенных в ней элементов (объектов). Для формы и всех элементов определены наборы свойств. Каждое свойство определяет какой-то аспект поведения или способ отображения объекта. В окне редактора свойств показываются свойства элемента, который в настоящий момент активен (выделен). В заголовке окна редактора отображается название этого элемента.



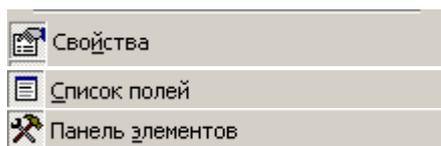
Редактор свойств содержит несколько вкладок, в которых свойства элемента группируются по их типу:

- **Макет** - свойства относящиеся к способу отображения элемента (как он будет выглядеть);
- **Данные** - из какого источника берутся данные, какие действия с ними разрешено выполнять и т.п.;
- **События** - действия, выполняемые при различных событиях (например, при щелчке мышью на элементе);
- **Другие** - свойства, не относящиеся к перечисленным группам;
- **Все** - общий список всех свойств.

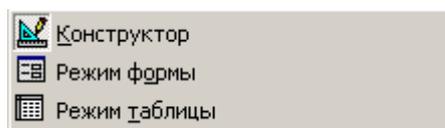
- **Список полей.** Если для формы указан источник данных (таблица или запрос), в данном списке отображаются поля источника, которые можно перенести в бланк формы.



Показать/скрыть панель элементов, редактор свойств или список полей позволяют соответствующие пункты меню «Вид»:



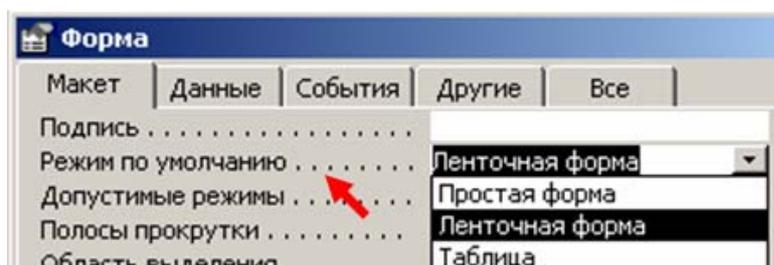
Форма может находиться в одном из состояний: «Конструктор» (режим редактирования), «Режим формы» (стандартный режим работы) и «Режим таблицы» (просмотр данных из подключенного источника в виде таблицы). Для переключения режимов используется меню «Вид»



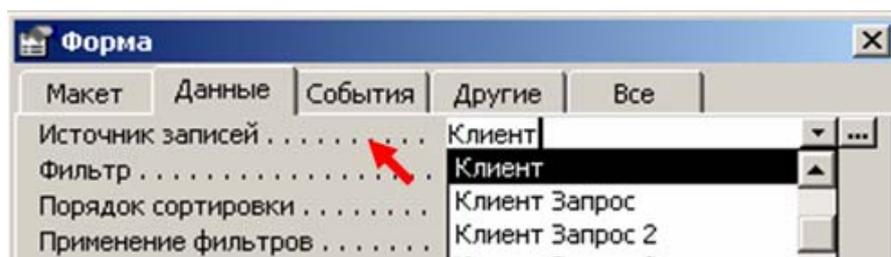
или кнопка «Вид»  на панели инструментов.

Пример создания ленточной формы. Создадим ленточную форму на основе таблицы «Клиент». Выберем пункт «Создание формы в режиме конструктора». При этом откроется пустой бланк формы. Убедимся, что открыто окно редактора свойств. После этого выделим объект «Форма», щелкнув мышью на квадрат  в левом верхнем углу бланка (также можно щелкнуть по нему правой кнопкой мыши и в контекстном меню выбрать пункт «Свойства») В окне редактора свойств отобразится список свойств объекта «Форма».

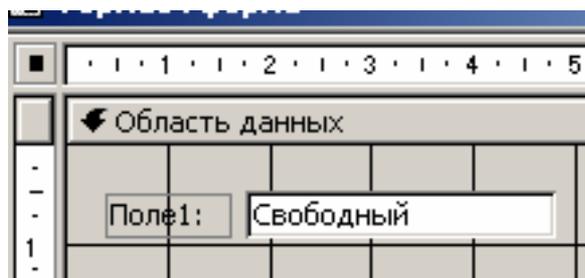
Выполним необходимую настройку свойств формы:
Зададим режим по умолчанию «Ленточная форма»;



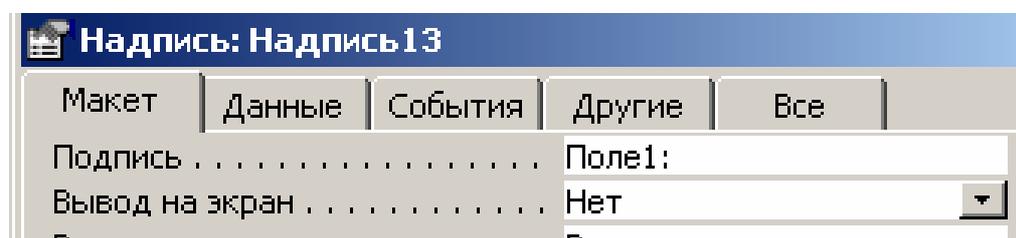
В качестве источника записей формы выберем таблицу «Клиент».



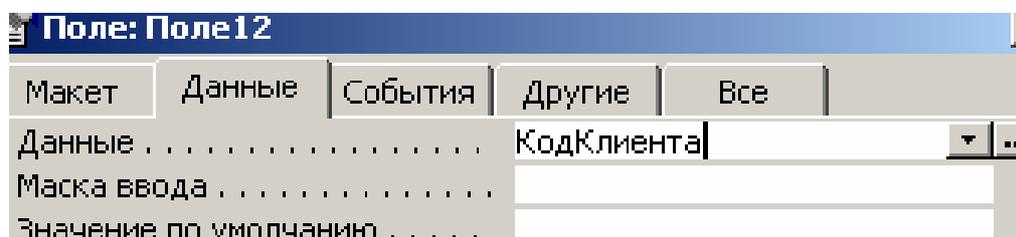
Используя кнопку «Поле»  на панели элементов, разместим в области данных бланка формы элемент «Поле».



Вместе с полем добавилась и текстовая подпись к нему (слева). Выделим подпись мышью (в редакторе свойств отобразятся свойства этого объекта) и зададим для свойства «Вывод на экран» значение «Нет».



Выделим элемент «Поле» и назначим для его свойства «Данные» значение «КодКлиента» (это имя поля из таблицы «Клиент», которое «привязывается» к данному текстовому полю в качестве источника данных).



Тем же способом создадим еще три поля и для них назначим в качестве значений свойства «Данные» соответственно имена полей «Фамилия», «Имя», «Отчество».

Используем для переключения в режим просмотра созданной формы кнопку «Вид»  на панели инструментов. На рис. 3.45 показан вид окна формы в двух режимах - редактирования и просмотра. Как видно, созданная форма работоспособна в принципе (в ней отображаются данные из таблицы «Клиент»), но требует доработки. Нужно расположить поля на форме более компактно и устранить лишнее пустое пространство. Кроме того, желательно добавить в форму заголовков с названиями полей. Для того, чтобы продолжить редактирование формы, следует переключить ее обратно в режим конструктора кнопкой «Вид»



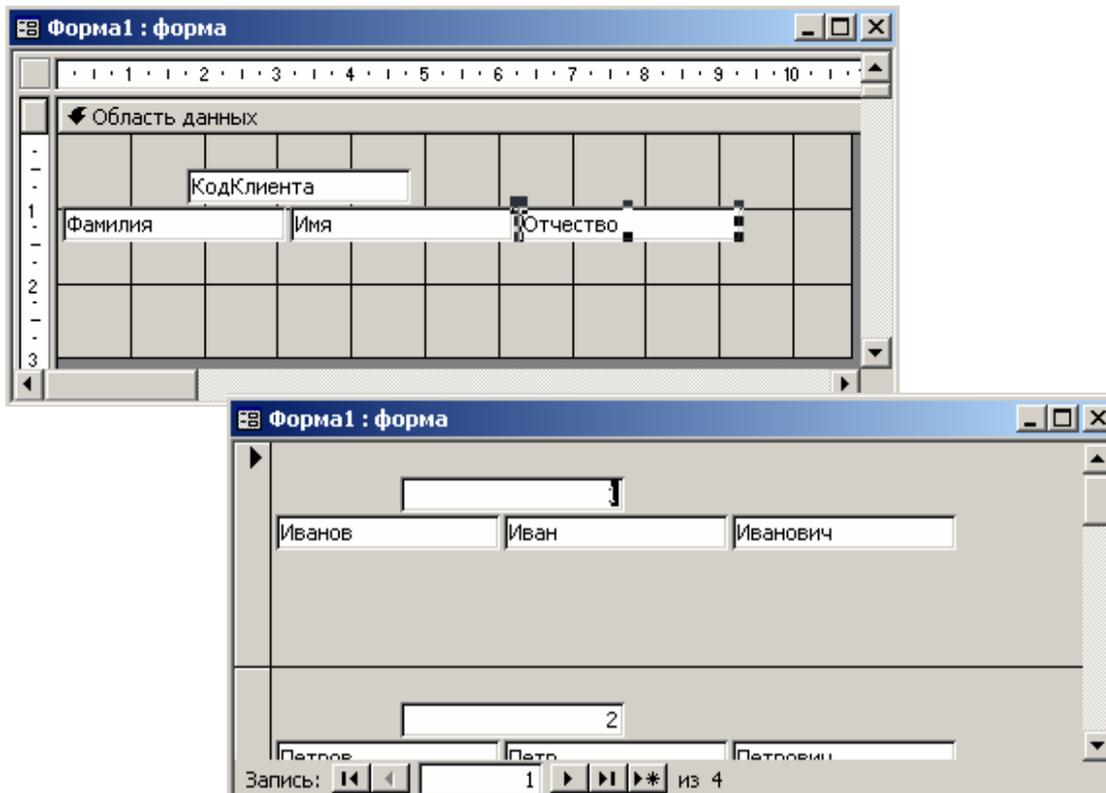
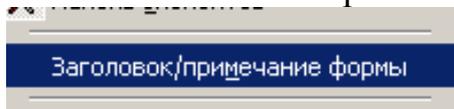
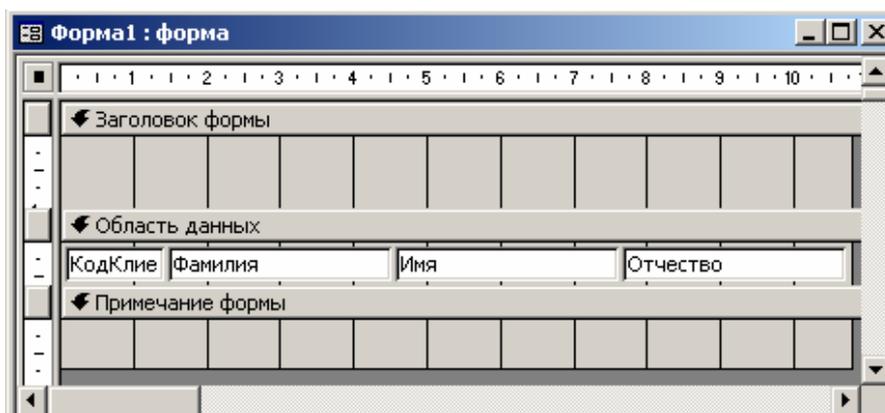


Рис. 3.45. Вид созданной формы

Уменьшим размер области данных, перемещая мышью ее границы. Для добавления заголовка выберем в меню «Вид» пункт «Заголовок/примечание формы»



В бланке формы должны появиться новые области: «Заголовок формы» и «Примечание формы».



Поскольку область примечания не нужна, уменьшим ее высоту до нуля. Затем, используя кнопку «Надпись» **Aa** на панели элементов, разместим в области заголовка формы четыре элемента «Надпись» и занесем в них названия полей. На рис. 3.46 показан новый вид формы в режиме редактирования и в режиме просмотра.

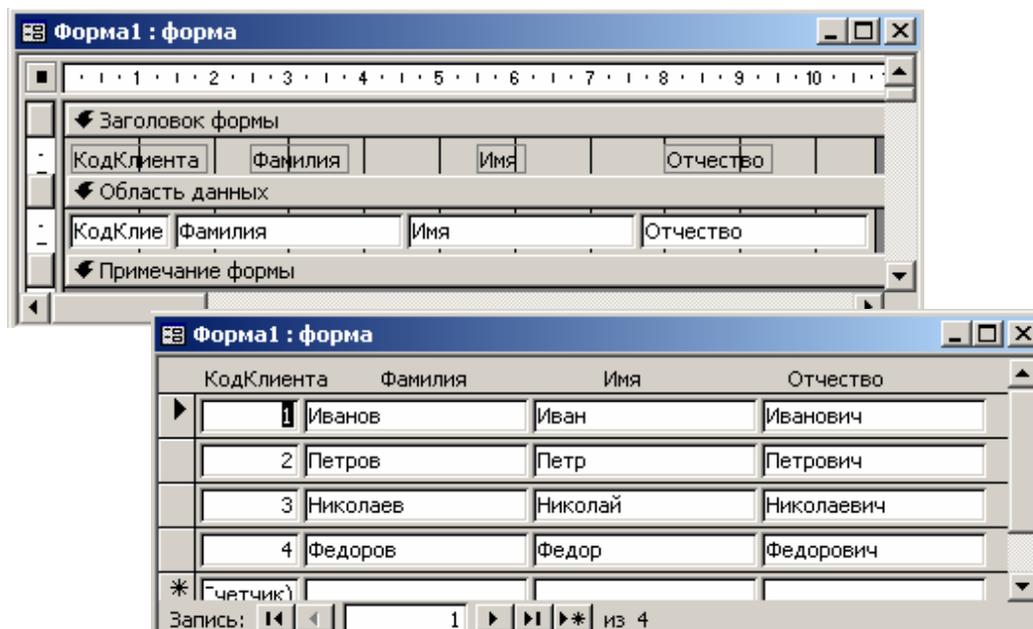


Рис. 3.46. Новый вид формы

Это ленточная форма (так было определено при ее создании) – в ней одновременно отображаются все записи из источника данных. При отображении формы в поля, размещенные в ее области данных, последовательно помещаются данные каждой из записей (область данных тиражируется по числу записей). Поля, помещенные в область заголовка или примечания формы, отображаются однократно (вверху и внизу формы соответственно).

Панель элементов. Панель элементов позволяет помещать в бланк формы различные элементы управления – надписи, переключатели, кнопки и т.п. Рассмотрим некоторые из этих элементов подробнее.

- 
 Элемент «**Надпись**». Позволяет разместить в форме некоторый текст.

Это текстовый элемент

- 
 Элемент «**Поле**». Предназначен для отображения и редактирования текстовых значений. Может быть связан с некоторым полем источника данных (таблицы или запроса) и отображать его значение для текущей записи.

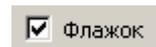
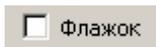
Поле:

- 
 Элемент «**Выключатель**». Кнопка с фиксацией, которая может принимать одно из двух состояний.

Выключатель

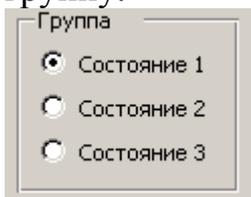
Выключатель

-  Элемент «**Флажок**». По назначению аналогичен элементу «**Выключатель**».

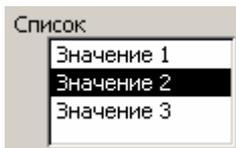


-  Элемент «**Переключатель**». Одиночный переключатель аналогичен флажку. Если переключатели объединены в группу, в каждый момент времени включен может быть только один из них.

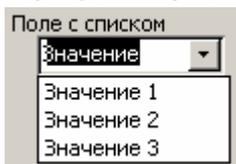
-  Элемент «**Группа переключателей**». Объединяет переключатели в группу.



-  Элемент «**Список**». Позволяет выбирать значение из списка. Источником строк может быть фиксированный набор значений, а также значения из таблицы или запроса.



-  Элемент «**Поле со списком**». Позволяет выбрать значение из выпадающего списка или вводить его вручную. Источником строк списка может быть фиксированный набор, таблица или запрос.



-  Элемент «**Подчиненная форма/отчет**». Элемент, позволяющий включать в форму другую форму.



В верхней части панели элементов расположена кнопка «**Мастера**». Если кнопка включена, при создании некоторых элементов будет вызываться соответствующий мастер, позволяющий в режиме диалога настроить свойства элемента. Мастера зачастую упрощают работу с элементами, но все необходимые свойства также могут быть заданы или изменены вручную.

Свойства элементов управления. В примере выше уже было показано, как используется редактор свойств для задания свойств формы и элементов управ-

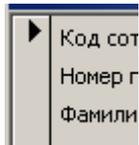
ления. Различных свойств довольно много и у разных элементов могут быть разные наборы свойств. Рассмотрим некоторые из них.

- Раздел «**Макет**»

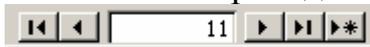
- **Подпись** - текстовая надпись, отображаемая в окне формы (или в заголовке окна для объекта-формы).

- **Режим по умолчанию** - только для объекта-формы, может принимать значения «**Простая форма**», «**Ленточная форма**» или «**Таблица**».

- **Область выделения** - только для формы, разрешает/запрещает отображение области выделения слева от записей в форме.



- **Кнопки перехода** - только для формы, разрешает/запрещает отображение кнопок перехода внизу формы.



- **Ширина, Высота** - задают размеры элемента.

- **Шрифт, Размер шрифта** - гарнитура и размер шрифта при отображении текста.

- Раздел «**Данные**»

- **Данные** - связь элемента (поля, флажка и т.п.) с определенным полем источника записей, заданного для формы. Если объект связан с некоторым полем, он называется **присоединенным**, иначе – **свободным**.

- **Источник записей** - только для объекта-формы, привязывает к форме некоторый источник данных (таблицу или запрос).

- **Тип источника строк** - для списка или поля со списком. Определяет, как формируются строки списка. Варианты: «**Таблица или запрос**», «**Список значений**» (явно заданный набор строк) или «**Список полей**» (список будет содержать перечень полей **источника строк**).

- **Источник строк** - для типов «Таблица или запрос» и «Список полей» указывается имя таблицы или запроса. Для типа «Список значений» - набор строк, разделенных «;».

- **Присоединенный столбец** - номер столбца источника строк, содержащего значения элемента (значения подставляются в поле данных, связанное с элементом).

- Раздел «**События**»

События - различные изменения, действия, которые могут происходить с объектом. Набор событий для различных объектов может сильно различаться. Для любого события может быть назначен обработчик – некоторая функция, вызываемая на выполнение при наступлении данного события.

Обычно обработчиками событий служат функции на языке Visual Basic.

- Раздел «**Другие**»

- **Имя** - уникальное в пределах формы имя элемента. По этому имени к нему можно обращаться, например, из функций на Visual Basic.

- **Текст строки состояния** - текст, отображаемый в строке состояния Ac-

cess (в нижней части окна), когда элемент формы выделен (получил фокус ввода).

Пример создания формы с подчиненной формой. Пусть имеются три связанные таблицы (рис. 3.47), содержащие сведения о назначении сотрудников на должности. Один и тот же сотрудник в разное время может назначаться на различные должности.



Рис. 3.47. Структура таблиц

Требуется построить форму «**Картотека сотрудников**». В главной форме отображаются данные сотрудника – фамилия, имя, отчество и т.д., в подчиненной – сведения о его назначениях. Запустим конструктор для создания новой формы. В панели элементов запретим использование мастеров  (для целей обучения лучше проделать всю работу вручную).

В редакторе свойств зададим следующие свойства формы:

- Макет → **Режим по умолчанию** = «Простая форма»
- Данные → **Источник записей** = «Сотрудники» (название таблицы)

Используя список полей, перенесем нужные поля в бланк формы (рис. 3.48). Разместим поля компактно.

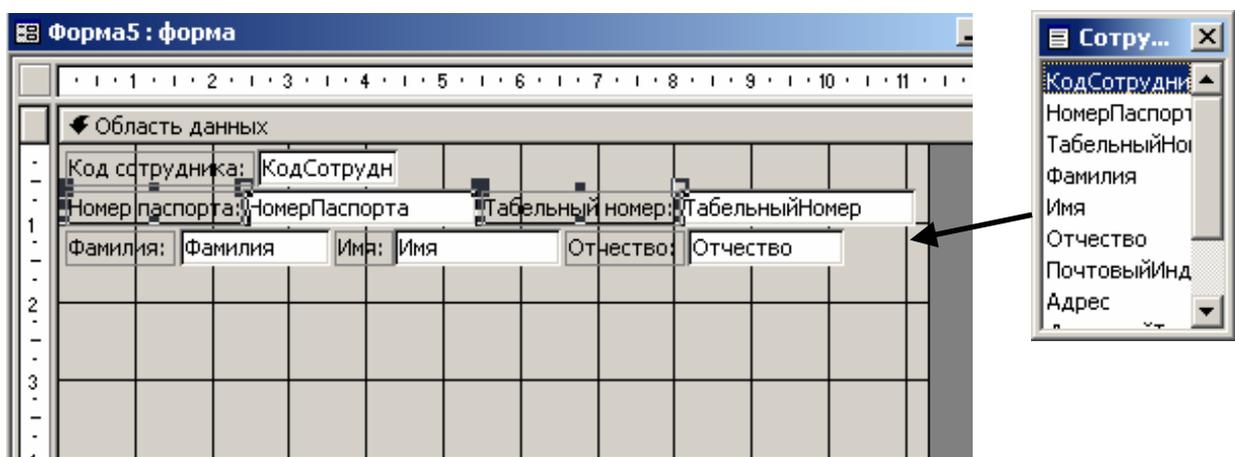


Рис. 3.48. Размещение полей в форме

Просмотрим предварительные результаты работы (рис. 3.49). Созданная форма связана с таблицей «Сотрудники» и позволяет просматривать хранящиеся в ней записи – в каждый момент времени показана одна запись.

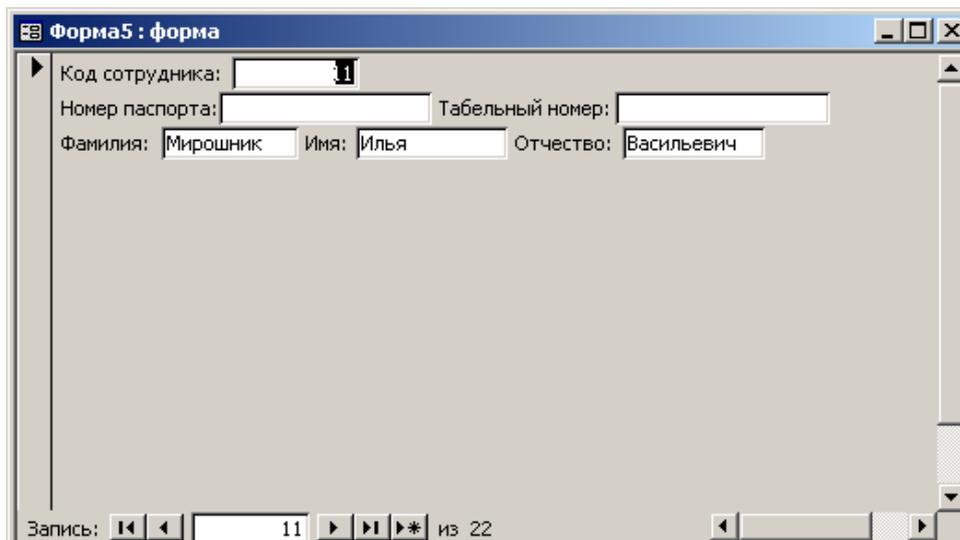


Рис. 3.49. Вид простой формы

В нижней части формы оставлено свободное место – для размещения подчиненной формы со списком назначений сотрудника. Вернемся в режим конструктора и с помощью панели инструментов добавим в бланк формы элемент «Подчиненная форма/отчет» (рис. 3.50). Появившуюся при этом над прямоугольником подчиненной формы надпись можно удалить или сделать невидимой.

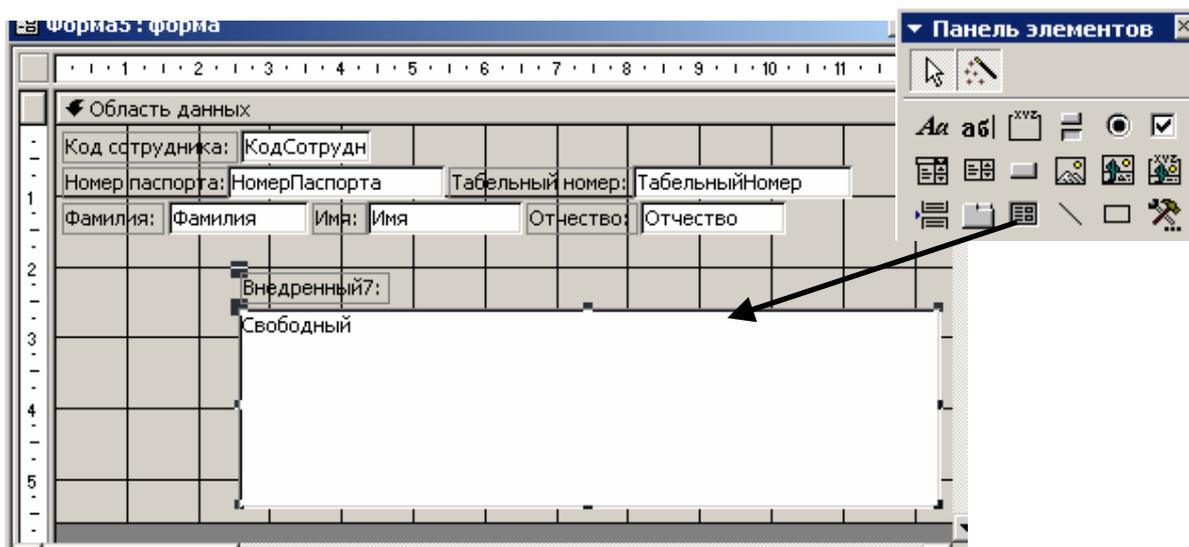


Рис. 3.50. Добавление подчиненной формы

Теперь зададим свойства подчиненной формы, которые свяжут ее с главной формой и с таблицей – источником данных:

- Данные → **Объект-источник** = «Таблица.Назначения».
Это свойство определяет объект, формирующий набор данных для под-

чиненной формы (для главной формы было задано другое свойство «Данные → Источник записей»).

- Данные → Основные поля = «КодСотрудника». Свойство задает имя поля в таблице, являющейся источником данных для главной формы. С этим полем будут связываться данные подчиненной формы.
- Данные → Подчиненные поля = «КодСотрудника». Свойство задает имя поля из источника данных подчиненной формы, которое должно быть связано с «основным» полем.

Имена основного и подчиненного полей в данном случае совпадают, но это поля разных источников. В подчиненной форме будут отображаться только те записи из таблицы «Назначения», в которых поле «КодСотрудника» (подчиненное) содержит то же значение, что и поле «КодСотрудника» (основное) записи из таблицы «Сотрудники», отображаемой в данный момент в главной форме.

На рис. 3.51 показано, как будет выглядеть созданная форма.

ДатаНазначения	Назначен	Приказ
20.09.2003	Курьер	
25.09.2003	Секретарь	
15.10.2003	Менеджер	

Рис. 3.51. Форма с подчиненной формой

Здесь в качестве объекта-источника была напрямую указана таблица (и таблица отображается в окне подчиненной формы). Источником может быть и другая форма. В этом случае ее можно создать отдельно, а затем подключить в

качестве подчиненной. Если на панели элементов включена кнопка , разрешающая использование мастера, мастер может облегчить задачу создания отдельной формы для использования в качестве подчиненной. На рис. 3.52 показана форма, которая может быть использована самостоятельно или в качестве подчиненной в созданной ранее форме. Когда форма запущена как самостоятельная, она отображает все записи таблицы «Назначения». Когда она используется в роли подчиненной, срабатывает ограничение, определенное связью между полями (основное поле/подчиненное поле) и в каждый момент времени отображаются только записи, связанные с текущей записью главной формы.

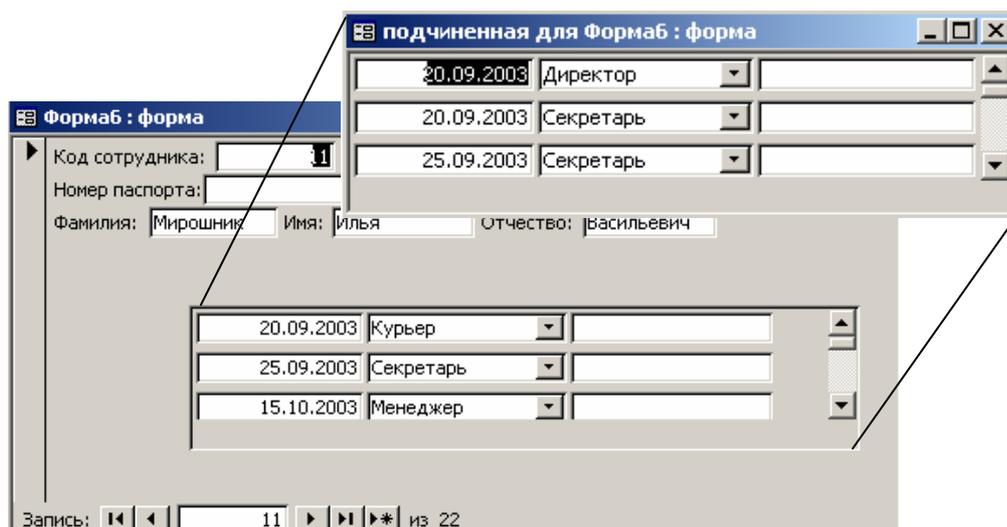


Рис. 3.52. Использование отдельной формы в качестве подчиненной

Отчеты. Способы построения отчетов и форм очень близки. Различия между ними связаны с тем, что формы предназначены для просмотра и ввода данных, а отчеты – для просмотра и вывода на печать. В отчетах присутствуют некоторые дополнительные возможности по форматированию данных в документах. Рассмотрим процесс построения отчета с помощью конструктора на примере: составим отчет, выводящий список сотрудников в алфавитном порядке. Запустим конструктор отчетов для создания нового отчета. В бланке отчета (рис. 3.53) присутствуют области, которых не было в формах: верхний и нижний **колонтитул**. Колонтитул выводится на каждой печатной странице вверху или внизу листа.

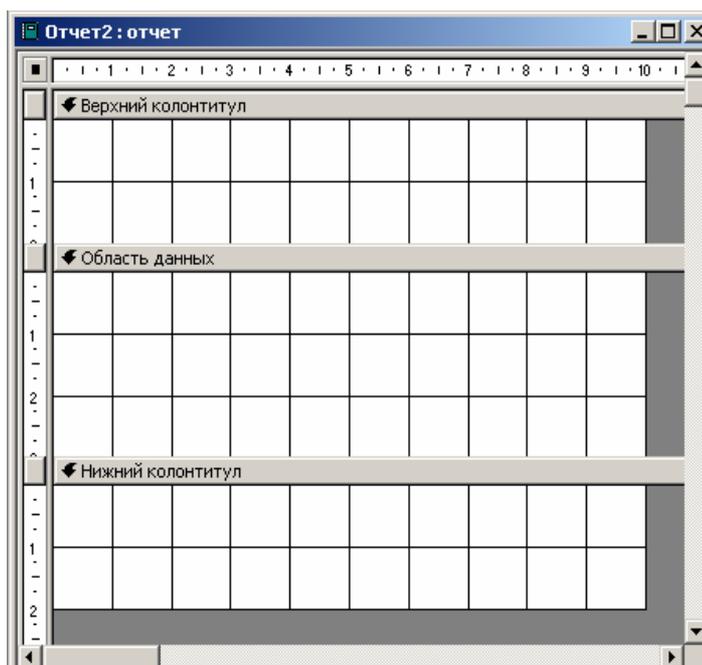
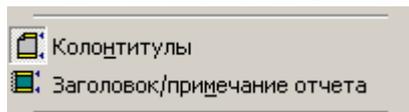


Рис. 3.53. Бланк отчета

Колонтитулы можно разрешить или запретить с помощью контекстного меню бланка отчета или в меню «Вид».



Также можно разрешить или запретить использования заголовка и примечания отчета, выводимых в начале и в конце отчета соответственно.

Зададим свойство объекта отчета:

- Данные → **Источник записей** = «Сотрудники»
(имя таблицы)

Разместим в области данных элемент «Поле» и зададим для него следующее свойство:

- Данные → **Данные** = «=[Фамилия]+" "+[Имя]+" "+[Отчество]»
(три поля из источника данных объединяются с помощью выражения, признак того, что это выражение – знак «=» в начале)

Уменьшим размеры области данных и запретим колонтитулы. На рис. 3.54 показано, как выглядит бланк отчета (в режиме конструктора) и этот же отчет в режиме просмотра (для просмотра отчета используется кнопка «Предварительный просмотр»  на панели инструментов).

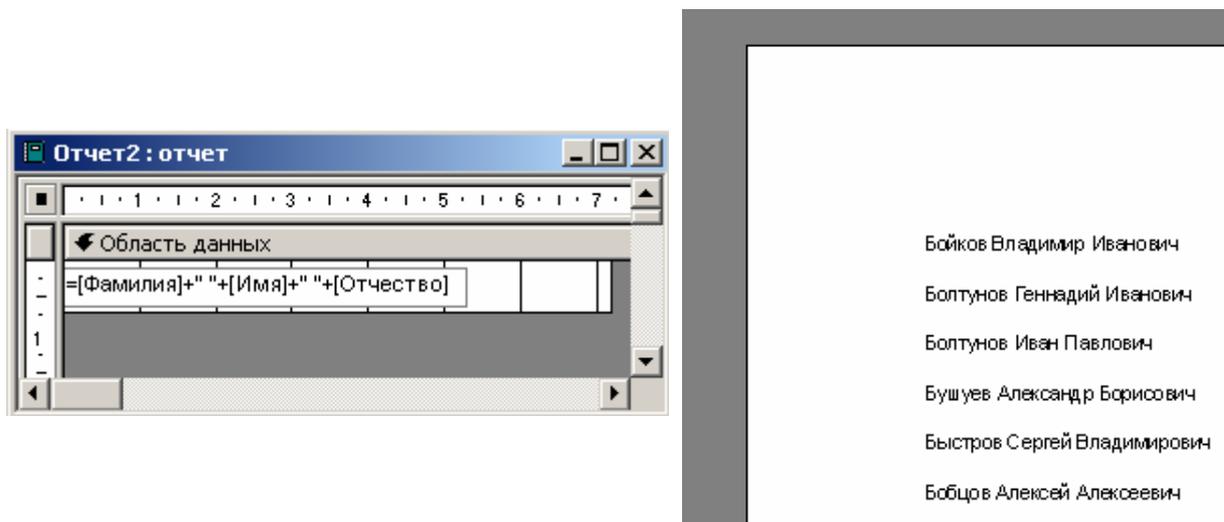
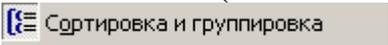
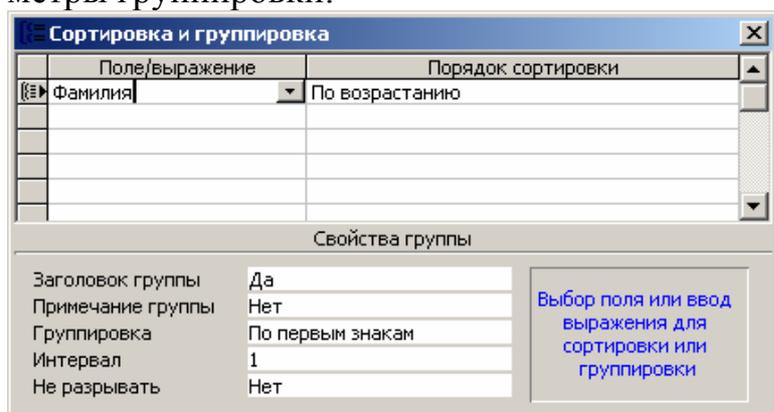


Рис. 3.54. Пример отчета

Применим теперь некоторые возможности, которых не было при работе с формами:

- Сгруппируем сотрудников по первой букве фамилии и перед каждой группой укажем эту букву.
- Выполним сортировку по фамилии.
- Пронумеруем сотрудников внутри каждой группы.

В контекстном меню бланка (или меню «Вид») выберем пункт «Сортировка и группировка» . В появившемся окне настроим параметры группировки:



(возможные варианты группировки - «По полному значению» или «По первым знакам»). Свойство «Интервал» - количество первых знаков, по которым выполняется группировка. Свойство «Заголовок группы» указывает на то, что в бланк отчета следует добавить соответствующую область (заголовок группы будет выводиться один раз для каждой сформированной группы).

Зададим свойства объекта отчета:

- Данные → Сортировка включена = «Да»
- Данные → Порядок сортировки = «Фамилия, Имя, Отчество»

В область заголовка поместим элемент «Поле» и зададим его свойства:

- Данные → Данные = «=Left(Фамилия;1)»
(выражение, отделяющее первую букву фамилии)

Для определения счетчика создадим элемент «Поле» в области данных, слева от фамилии и зададим его свойства:

- Данные → Данные = «=1»
(равняется единице)
- Данные → Сумма с накоплением = «Для группы»
(будет выводиться не единица, а значение счетчика, к которому каждый раз добавляется по единице)

На рис. 3.55 показано, как выглядит измененный отчет в режиме конструктора и в режиме просмотра.

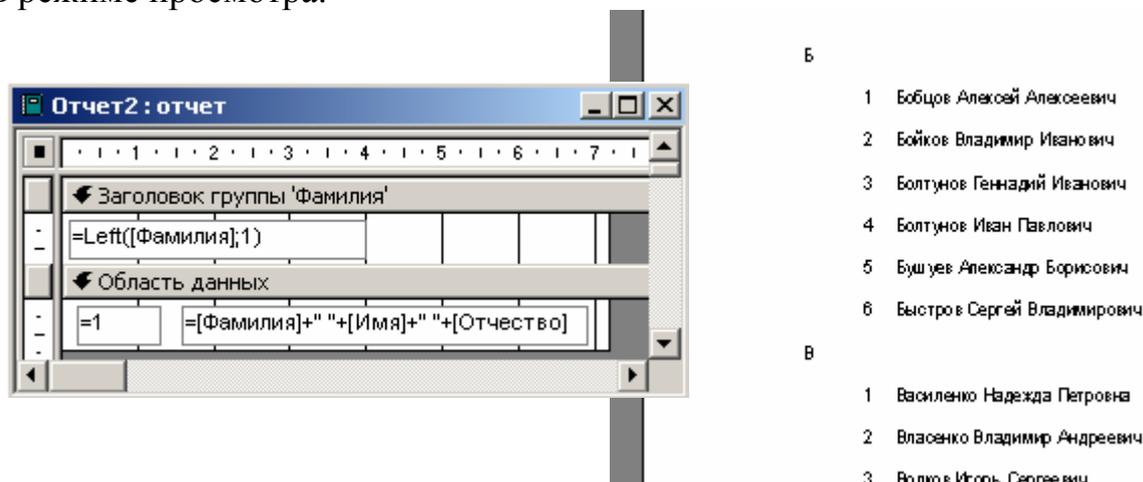


Рис. 3.55. Отчет с группировкой

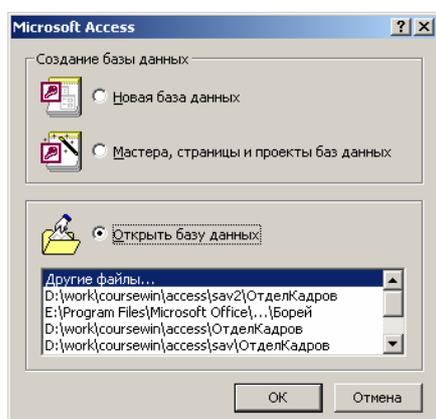
Задание 1. Работа с готовой базой данных. Приемы работы

Цель задания: Научиться запускать программу MS Access, открывать существующую базу данных и работать с ней. Освоить приемы работы с формами, таблицами, запросами и отчетами (просмотр, поиск, сортировка и фильтрация данных). Для работы используется учебная база данных «Борей», входящая в комплект поставки MS Access.

Последовательность действий:

1. Запустите Microsoft Access и откройте базу данных «Борей».

- Запустите программу MS Access, используя соответствующий ярлык в разделе «**Программы**» меню Windows.  Microsoft Access
- С помощью появившегося диалогового окна откройте базу данных «Борей» одним из двух способов:



а) Выберите пункт «**База данных Борей**» в списке известных и нажмите кнопку «**ОК**».

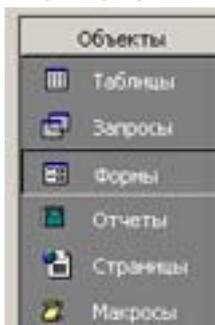
б) Выберите пункт «**Другие файлы...**», нажмите кнопку «**ОК**» и с помощью диалога открытия файлов отыщите файл «**Борей.mdb**» (при просмотре файлов их расширения могут не отображаться). Этот файл располагается в подкаталоге **Samples** каталога установки MS Access. Например «E:\Program Files\Microsoft Office\Office\Samples».

- Закройте возникшее окно заставки БД Борей.

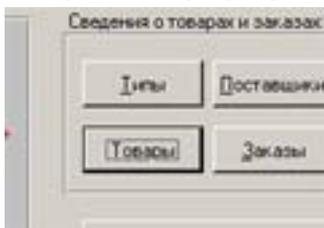


2. Откройте главную кнопочную форму и с ее помощью – форму «Товары».

- В списке типов объектов окна базы данных выберите пункт «**Формы**».



- Двойным щелчком мыши откройте объект «**Главная кнопочная форма**».
- В появившемся окне формы щелкните по кнопке «**Товары**».



3. Просмотрите список товаров, отыщите товар с самой большой и самой маленькой ценой за единицу измерения.

- Последовательно просмотрите отображаемые в форме записи, используя кнопки навигации.  Запись:   5   из 77
- Поместите курсор в поле «**Цена**». 
- Выполните сортировку записей по возрастанию цены нажатием кнопки «**Сортировка по возрастанию**» на панели инструментов окна Access.



- Перейдите к первой записи, используя кнопку навигации , и обратите внимание на марку и цену самого дешевого (за единицу измерения) товара.
- Перейдите к последней записи нажатием на кнопку  и выясните марку и цену самого дорогого товара.

4. Используйте фильтр для просмотра только тех записей, которые относятся к кондитерским изделиям.

- Нажмите кнопку «**Изменить фильтр**»  на панели инструментов Access.
- В поле «**Тип**»  редактора фильтра введите текст «**Кондитерские изделия**».
- Нажмите кнопку «**Применение фильтра**»  на панели инструментов.
- Используя кнопки навигации, просмотрите все записи, соответствующие созданному фильтру.

- По окончании просмотра отмените действие фильтра нажатием кнопки «Удалить фильтр».



5. Добавьте данные о новом товаре.

- Перейдите к редактированию новой записи нажатием кнопки.
- В поле «Марка» введите текст «Колбаса Докторская».

Марка: Колбаса Докторская

- В поле «Поставщик» выберите строку «ООО Экзотика» из выпадающего списка.
- В поле «Тип» задайте значение «Мясо/птица».
- В поле «Единица измерения» введите текст «1 Ящик».
- В поле «Цена» введите значение «1 200,00р.».

6. Просмотрите список поставщиков, используя соответствующую форму.

- Закройте окно формы «Товары».
- Нажмите кнопку «Поставщики» на главной кнопочной форме.
- Используя кнопки навигации, найдите запись о поставщике «ООО Экзотика».
- Для просмотра списка товаров, поставляемых данной фирмой, нажмите кнопку «Просмотр товаров».
- Убедитесь, что в открывшемся списке присутствует добавленный ранее товар.

7. Просмотрите список категорий товаров, найдите категорию «Мясо/птица» и убедитесь, что в ней присутствует добавленный товар.

- Закройте окно формы «Поставщики».
- Откройте форму «Типы» нажатием соответствующей кнопки на главной кнопочной форме. Форма «Типы» содержит подробную информацию о категориях товаров (название, описание, иллюстрация, список товаров, принадлежащих данной категории). Двойной щелчок по картинке позволяет вызвать графический редактор для ее редактирования. Используя контекстное меню рисунка также можно скопировать его в буфер обмена, вставить из буфера новый рисунок или загрузить произвольный объект из файла.
- Поместите курсор в поле «Категория».

Категория: Кондитерские изделия

- Для обнаружения категории «Мясо/птица» воспользуйтесь функцией

поиска. Нажмите кнопку «Найти» на панели инструментов. При этом откроется диалоговое окно «Поиск и замена»:



- В поле «Образец» введите текст «мясо».

- В поле «Совпадение» выберите вариант «С начала поля».

- Нажмите кнопку «Найти далее».

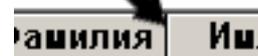
В форме «Типы» должна отобразиться искомая запись

- Закройте диалоговое окно «Поиск и замена».
- Просмотрите список товаров, относящихся к категории «Мясо/птица» и убедитесь, что в нем присутствует информация о добавленном ранее товаре.
- Закройте окно формы «Типы».

8. Просмотрите данные о сотрудниках с помощью формы и сравните их с данными в соответствующей таблице. *Все данные хранятся в таблицах, а формы – удобный пользовательский интерфейс для просмотра и редактирования данных.*

- Откройте форму «Сотрудники» (она располагается в списке форм окна базы данных – там же, где и главная кнопочная форма).
- Просмотрите служебные и личные данные о сотрудниках.
- Добавьте с помощью формы данные о новом сотруднике.
- В списке типов объектов окна базы данных выберите пункт «Таблицы».
- Откройте таблицу «Сотрудники».
- Для удобства просмотра данных настройте ширину столбцов таблицы,

перемещая мышью разделитель заголовков столбцов.



Если в процессе работы с таблицей были изменены элементы ее макета (размеры окна или столбцов), при закрытии окна будет задан вопрос о том, нужно ли сохранить эти изменения. Если изменены, добавлены или удалены данные в таблице, эти изменения сохраняются автоматически.

- Просмотрите записи таблицы и сравните их с данными, отображаемыми в форме.

Для перемещения по записям можно использовать кнопки навигации, расположенные внизу окна таблицы.

Самостоятельная работа: Изучите объекты базы данных «Борей» (таблицы, формы, отчеты и запросы), которые не были рассмотрены выше. Для каких целей предназначен каждый из них? Проанализируйте, какие варианты использования базы данных возможны (с точки зрения сотрудника отдела кадров, менеджера по продажам, вице-президента компании). Какие объекты было бы полезно добавить для использования базы данных в вашей работе?

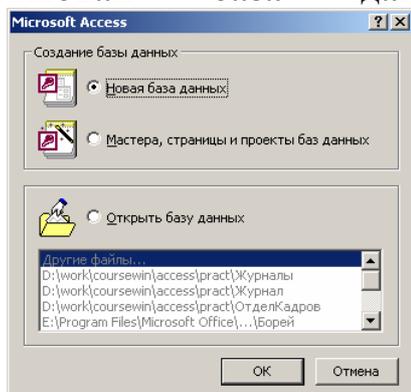
Задание 2. Создание и редактирование таблиц. Работа со схемой данных

Цель задания: Научиться создавать новую базу данных, создавать и редактировать структуру таблиц и устанавливая связи между ними, используя схему

данных, а также просматривать и редактировать данные, хранящиеся в таблицах.

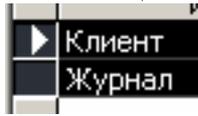
Последовательность действий:

1. Запустите программу Microsoft Access.
2. Создайте новую базу данных «Подписка на журналы».
- В появившемся при старте MS Access диалоговом окне выберите пункт **«Новая база данных»** и нажмите кнопку **«ОК»**.



- С помощью стандартного диалогового окна Windows задайте имя и местоположение файла базы данных. В качестве местоположения базы данных следует использовать ваш рабочий каталог, в качестве имени файла – **«Журналы.mdb»**.
Расширение .mdb вводить не обязательно, оно будет добавлено автоматически.
- 3. Создайте таблицы **«Журнал»**, **«Клиент»** и **«Подписка»**.
- Выбрав раздел объектов **«Таблицы»**, щелкните дважды по ярлыку **«Создание таблицы в режиме конструктора»**.
- В открывшемся окне конструктора введите следующие имена и типы данных полей: **«КодЖурнала»** - тип **«Счетчик»**, **«Название»** - тип **«Текстовый»**.
- Объявите поле **«КодЖурнала»** первичным ключом таблицы. Для этого следует поместить курсор в строку, описывающую это поле и нажать кнопку **«Ключевое поле»**  на панели инструментов Access. Другой способ - нажать правой кнопкой мыши на соответствующей строке и в появившемся контекстном меню выбрать пункт **«Ключевое поле»**.
- Для завершения создания таблицы нажмите на кнопку закрытия окна конструктора  (не окна базы данных и не окна Access). На вопрос о необходимости сохранения структуры таблицы следует ответить утвердительно и ввести название таблицы – **«Журнал»**.
- Тем же способом создайте таблицу **«Клиент»**, содержащую следующие поля: **«КодКлиента»** (счетчик), **«Фамилия»**, **«Имя»** и **«Отчество»** (текстовые поля). Сделайте поле **«КодКлиента»** первичным ключом таблицы.
- Создайте таблицу **«Подписка»** с полями **«Клиент»** и **«Журнал»**, имеющими числовой тип данных (длинное целое). Оба эти поля должны обра-

зывать составной первичный ключ таблицы. Для создания составного первичного ключа следует выделить строки, описывающие соответствующие поля, с помощью области выделения (колонка по левому краю



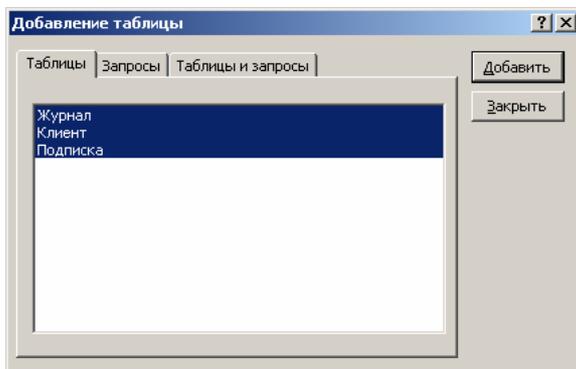
конструктора) , удерживая клавишу **Ctrl**, и нажать кнопку

«**Ключевое поле**»



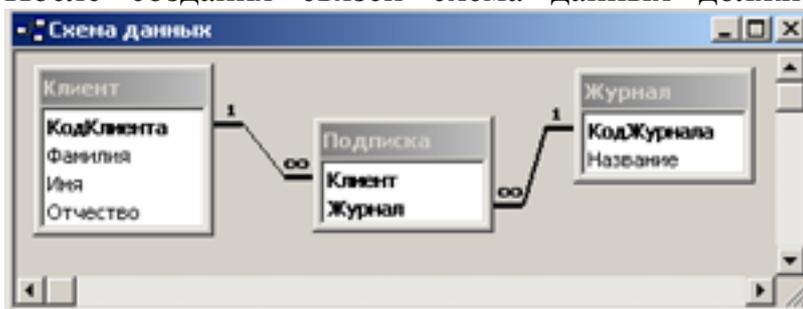
4. Определите связи между созданными таблицами.

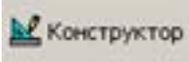
- Откройте окно схемы данных нажатием на кнопку  на панели инструментов окна Access.
- В появившемся диалоговом окне выделите все созданные таблицы и нажмите кнопку «**Добавить**».



- Закройте диалоговое окно добавления таблиц.
- Установите связь между полем «**КодКлиента**» таблицы «**Клиент**» и полем «**Клиент**» таблицы «**Подписка**». Для этого следует «перетащить» мышкой одно из связываемых полей на другое. В открывшемся диалоговом окне установите свойства связи: «**Обеспечение целостности данных**», «**каскадное обновление связанных полей**», «**каскадное удаление связанных записей**». Поле «**КодКлиента**» является первичным ключом, поле «**Клиент**» - внешним (вторичным) ключом. Свойство «обеспечение целостности» гарантирует, что внешний ключ не сможет принимать значения, которых нет среди значений первичного ключа. Свойство «каскадное обновление» обеспечивает автоматическое изменение значений внешних ключей при изменении значения первичного ключа. Свойство «каскадное удаление» обеспечивает удаление из связанной таблицы записей, которые связаны с удаляемыми записями главной таблицы.
- Установите связь между полем «**КодЖурнала**» таблицы «**Журнал**» и полем «**Журнал**» таблицы «**Подписка**».

После создания связей схема данных должна иметь следующий вид:



- Закройте окно схемы данных. На вопрос о сохранении изменений макета следует ответить утвердительно.
5. Введите данные в созданные таблицы.
- Для открытия таблицы в режиме редактирования данных следует дважды щелкнуть мышкой по ее названию.
 - Введите данные о нескольких клиентах в таблицу «Клиент».
 - Введите данные о нескольких журналах в таблицу «Журнал».
 - Для некоторых клиентов введите данные в таблицу «Подписка», устанавливающие их связи с некоторыми из журналов.
6. Измените свойства полей таблицы «Подписка» так, чтобы при ее просмотре отображались не малоинформативные коды клиента и журнала, а фамилия клиента и название журнала соответственно.
- Для редактирования структуры таблицы выделите мышью ее название и нажмите кнопку «Конструктор»  в окне базы данных.
 - Поместите курсор в строку, описывающую поле «Клиент».
 - В нижней части окна конструктора откройте вкладку «Подстановка». Поле «Клиент» таблицы «Подписка» является внешним ключом и должно принимать значения из множества значений первичного ключа таблицы «Клиент». Подстановка позволяет выбирать с помощью выпадающего списка значения, подставляемые в поле.
 - В свойстве «Тип элемента управления» замените значение «Поле» на «Список».
 - Задайте значение «Таблица или запрос» для свойства «Тип источника данных».
 - В качестве источника строк выберите таблицу «Клиент».
 - Задайте значение «1» для свойства «Присоединенный столбец». Присоединенный – это столбец из источника строк, значения которого должны подставляться в поле. В данном случае присоединенным является столбец «КодКлиента» таблицы «Клиент», его порядковый номер в этой таблице равен «1». Значения из этого столбца будут использованы для подстановки в поле «Клиент».
 - Задайте значение «2» для свойства «Число столбцов». Это означает, что при просмотре списка будут использованы первые два столбца источника строк – в данном случае поля «КодКлиента» и «Фамилия» таблицы «Клиент».

- Свойство «**Ширина столбцов**» позволяет указать, сколько места отводится для отображения каждого из отобранных столбцов. Например, можно задать «1см;1см» - точка с запятой является разделителем. В нашем случае следует задать значение «0см». Это означает, что первый столбец отображаться не должен (он используется для подстановки), а второй (столбец «**Фамилия**») будет занимать столько места, сколько ему будет отведено (его ширина жестко не задается).
- Перейдите в режим просмотра/редактирования данных таблицы нажав кнопку «**Режим таблицы**» . На запрос о сохранении таблицы ответьте утвердительно. При этом таблица должна иметь вид, как показано на рисунке.

Подписка : таблица		
	Клиент	Журнал
	Иванов	1
	Иванов	3
▶	Петров ▾	2
	Иванов	1
*	Петров	0
	Николаев	
	Федоров	

Обратите внимание: в таблице «Подписка» по-прежнему хранятся **числовые коды** клиентов и журналов, но в поле «Клиент» при просмотре и редактировании данных отображаются соответствующие текстовые элементы.

- Вернитесь к редактированию структуры таблицы нажатием кнопки «**Конструктор**» на панели инструментов Access .
- Аналогично определите подстановку для поля «**Журнал**». В качестве источника строк задается таблица «**Журнал**».
- Закройте окно конструктора, сохранив внесенные изменения.

Самостоятельная работа:

1. Изучите схему данных базы данных «Борей». Выясните, какие поля таблиц являются первичными ключами, а какие – внешними.

2. Созданная база данных имеет недостатки, которые следует устранить:

- Информация о клиенте должна содержать также его адрес
- Клиент может выписать несколько экземпляров журнала, поэтому необходимо хранить сведения о количестве выписанных экземпляров. Кроме того, подписка оформляется на определенный срок (будем считать, что всегда на 1 год). Этот факт также должен быть отражен в таблице «Подписка» (поле, хранящее номер года, должно иметь числовой тип). Измените структуру первичного ключа таблицы «Подписка». Какие из полей должны входить в него теперь?

При определении составного первичного ключа обеспечивается уникальность комбинации значений полей, входящих в него.

Введите данные в добавленные поля таблиц.

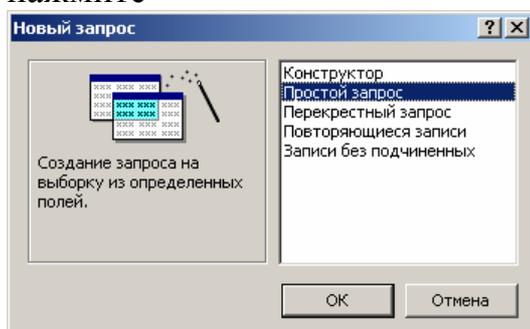
Задание 3. Создание и редактирование запросов

Цель задания: Научиться создавать и редактировать запросы к базе данных. Используется разработанная ранее база данных «Журналы».

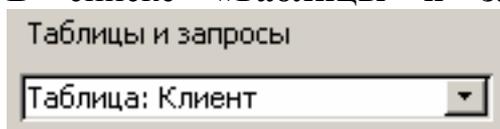
Последовательность действий:

1. Запустите Microsoft Access и откройте базу данных «Журналы».
2. Создайте запрос «Журналы клиентов», отображающий список клиентов и журналы, на которые они подписаны.

- В списке типов объектов окна базы данных выберите пункт «Запросы».
- Нажмите кнопку «Создать»  в окне базы данных.
- В появившемся диалоговом окне выберите пункт «Простой запрос» и нажмите кнопку  «ОК».

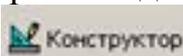


- В списке «Таблицы и запросы» выберите таблицу «Клиент».



- Перенесите названия полей «Фамилия», «Имя» и «Отчество» из списка «Доступные поля» в список «Выбранные поля», используя кнопку переноса .
- В списке «Таблицы и запросы» выберите таблицу «Подписка», перенесите поле «Журнал» в список «Выбранные поля» и нажмите кнопку «Далее».
- Еще раз нажмите кнопку «Далее». В поле «Задайте имя запроса» введите «Журналы клиентов» и нажмите кнопку «Готово».
- Просмотрите результат работы созданного запроса. В результирующей таблице в поле «Журнал» отображается не код журнала, а его название, поскольку для поля «Журнал» таблицы «Подписка» ранее было задано свойство подстановки.
- Закройте окно запроса.

3. Измените созданный запрос таким образом, чтобы в списке отображались и те клиенты, у которых нет подписки ни на один журнал (для них поле «Журнал» должно отображаться пустым).

- Для редактирования запроса выделите мышью его название и нажмите кнопку «Конструктор»  в окне базы данных.

- В окне конструктора, в области таблиц нажмите правой кнопкой мыши на линию связи между таблицами «Клиент» и «Подписка». В контекстном меню выберите пункт «**Параметры объединения**». В открывшемся диалоговом окне выберите второй вариант объединения и нажмите кнопку «**ОК**».

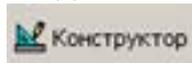
Объединение ВСЕХ записей из "Клиент" и только тех записей из "Подписка", в которых связанные поля совпадают.

Такой вариант объединения таблиц называют левым внешним объединением. Изменение параметров объединения влияет только на поведение данного запроса, но не имеет отношения к связям схемы данных.

- Нажмите на кнопку просмотра результатов работы запроса и убедитесь, что его поведение изменилось.
- Нажмите на кнопку закрытия окна запроса. Подтвердите сохранение внесенных изменений.

4. Сделайте копию созданного запроса и измените новый запрос таким образом, чтобы он выдавал список клиентов и для каждого из них подсчитывал количество журналов, на которые они подписаны. Строки должны быть отсортированы в алфавитном порядке по фамилии клиента.

- Нажмите правой кнопкой мыши на названии созданного запроса и в контекстном меню выберите пункт «**Копировать**».
- Нажмите правой кнопкой мыши в области окна базы данных и выберите пункт «**Вставить**».
- В открывшемся диалоговом окне введите название нового запроса «**Количество журналов клиента**».
- Выделите новый запрос мышью и нажмите кнопку «**Конструктор**»



в окне базы данных.

- Нажмите кнопку «**Групповые операции**» на панели инструментов Access. При этом в бланке запроса должна появиться соответствующая строка.
- В строке «**Групповая операция**» задайте для поля «**Журнал**» операцию «**Count**» (подсчет количества), для полей «**Фамилия**», «**Имя**» и «**Отчество**» оставьте операцию «**Группировка**».
- Для поля «**Фамилия**» определите тип сортировки «**по возрастанию**».
- Перейдите в режим просмотра результатов работы запроса . Просмотрите результаты.
- Нажмите на кнопку закрытия окна запроса. Подтвердите сохранение внесенных изменений.

5. Создайте перекрестный запрос, в качестве заголовков строк использующий фамилии клиентов, в качестве заголовков столбцов – названия журналов, а в качестве значений таблицы – отметки о наличии или отсутствии у клиента подписки на соответствующий журнал. От рассмотренного в разделе 3.1 данный запрос должен отличаться тем, что в результирующей таблице будут при-

существовать также клиенты, у которых нет подписки ни на один журнал и журналы, на которые никто не подписан. Для удобства следует создать вспомогательный запрос.

- Щелкните мышкой ярлык «Создание запроса в режиме конструктора».
- В диалоговом окне «Добавление таблицы» выделите таблицы «Клиент» и «Журнал» и нажмите кнопку «Добавить». Закройте диалоговое окно.

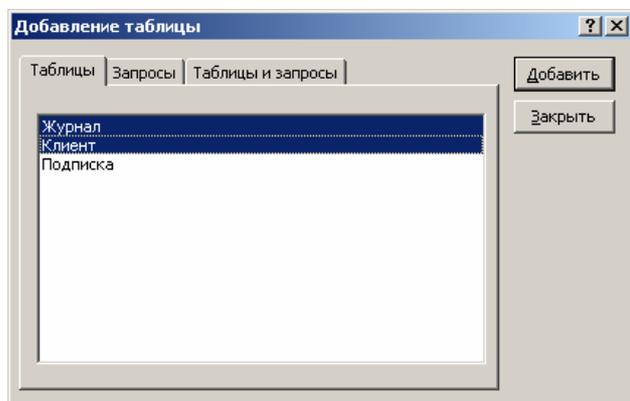


Таблица «Подписка» не должна использоваться при построении данного запроса.

- Перенесите мышкой в бланк запроса поля «КодКлиента» и «Фамилия» из таблицы «Клиент» и поля «КодЖурнала» и «Название» из таблицы «Журнал».
- Закройте конструктор, сохранив запрос под именем «Клиенты и журналы - вспомогательный».
- Выполните запрос и просмотрите результат его работы. В результирующей таблице должны присутствовать коды и фамилии всех клиентов во всех возможных сочетаниях с кодами и названиями журналов. Если таблица «Клиент» содержит информацию о 10 клиентах и таблица «Журнал» содержит информацию о 5 журналах, то результирующая таблица будет содержать 50 строк. Данный запрос образует декартово произведение таблиц.
- Запустите конструктор для создания нового запроса. В диалоговом окне откройте вкладку «Таблицы и запросы», выделите в списке запрос «Клиенты и журналы - вспомогательный» и таблицу «Подписка» и нажмите кнопку «Добавить». Закройте диалоговое окно.
- Установите связь между полями «КодКлиента» и «Клиент» (перетащите мышью поле «КодКлиента» на поле «Клиент»).
- Установите связь между полями «КодЖурнала» и «Журнал».
- Измените вид объединения для каждой из связей на внешнее левое объединение (на конце каждой из связей, направленном на таблицу «Подписка», должна появиться стрелка).
- Нажмите правой кнопкой мыши на поле таблиц конструктора и в контекстном меню выберите пункт «Тип запроса → Перекрестный».
- Перенесите мышью в бланк запроса поля «Фамилия», «Название» и «Журнал».

- Для полей «**Фамилия**» и «**Название**» задайте операцию «**Группировка**», для поля «**Журнал**» - операцию «**Count**».
- В строке «**Перекрестная таблица**» для поля «**Фамилия**» выберите вариант «**Заголовки строк**», для поля «**Название**» - «**Заголовки столбцов**», для поля «**Журнал**» - «**Значение**».
- Просмотрите результат работы запроса. В ячейках таблицы содержатся значения «1», если соответствующего клиента есть подписка на данный журнал и значения «0» в противном случае. Если в таблице «**Подписка**» присутствует поле «**Количество**», его можно использовать в запросе вместо поля «**Журнал**» с групповой операцией «**Sum**».
- Закройте окно запроса, сохранив его под именем «**Клиенты и журналы - перекрестный**».

Самостоятельная работа:

1. Ознакомьтесь с работой следующих запросов базы данных «Борей»:

- Итоги продаж по объему
- Продажи по сотрудникам и странам
- Продажи товаров в 1997
- Счета

Используя конструктор запросов, изучите их структуру. Сравните условия отбора данных в запросах «Продажи по сотрудникам и странам» и «Продажи товаров в 1997».

2. Следует внести некоторые изменения в созданный запрос «Количество журналов клиента».

- Если правильно была проделана самостоятельная работа в соответствии с заданием 2, таблица «Подписка» должна содержать поле «Количество», определяющее количество экземпляров данного журнала, выписанных клиентом. Следует добавить это поле в бланк запроса и выполнить над ним групповую операцию «Sum» (суммирование). Если использованная ранее над полем «Журнал» операция «Count» дает в результате количество уникальных журналов, на которые подписан клиент, то добавленное выражение будет подсчитывать общее количество журналов.
- Поскольку гипотетически возможна ситуация, когда существуют клиенты – полные тезки, следует различать клиентов по их уникальному коду. Для этого нужно добавить в бланк запроса поле «КодКлиента» из таблицы «Клиент» и определить для него операцию «Группировка».
- В таблице «Подписка» должна присутствовать информация о годе, на который оформлена подписка. Измените запрос таким образом, чтобы при его запуске запрашивался параметр «Год» и производился подсчет только тех записей, которые соответствуют введенному значению.

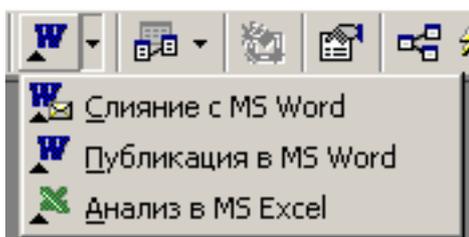
3. Измените запрос «Клиенты и журналы - перекрестный» таким образом, чтобы значения, равные нулю, не отображались в результирующей таблице. Для этого можно воспользоваться функцией «If».

Задание 4. Интеграция Access с приложениями MS Office. Экспорт и импорт данных

Цель задания: Освоить методы организации взаимодействия между MS Access и приложениями MS Office (Word, Excel). Научиться экспортировать и импортировать данные в Access. Используются учебная база данных «Борей», входящая в комплект поставки MS Access, и разработанная ранее база данных «Журналы».

Последовательность действий:

1. Запустите Microsoft Access и откройте базу данных «Борей».
2. Опубликуйте в MS Word данные из таблицы «Товары».
 - Выделите мышью имя таблицы «**Товары**» и запустите процедуру публикации, выбрав соответствующую кнопку:



Также можно использовать пункт меню «Сервис → Связи с Office → Публикация в MS Word». При этом будет создан файл с именем «Товары.rtf». Если такой файл уже существует, будет предложено заменить его или создать другой файл. Файл будет создан в рабочем каталоге Access (имя рабочего каталога можно изменить с помощью вкладки «Общие» диалога «Параметры» - пункт меню «Сервис → Параметры...»). Созданный файл будет открыт программой MS Word.

- Используя средства Word, отредактируйте сформированную таблицу. Удалите столбцы, которые не нужны в создаваемом документе. Настройте ширину столбцов. При необходимости задайте альбомное расположение страниц.
 - Поместите в начале документа заголовок: «Товары и цены».
 - Используя пункт меню Word «Файл → Сохранить как...» сохраните созданный документ в формате Word в свой рабочий каталог.
 - Закройте MS Word и вернитесь к работе с Access.
3. Выполните анализ в MS Excel данных из запроса «Сведения о заказах».
 - Выделите мышью имя запроса «Сведения о заказах».
 - Нажмите кнопку «Анализ в MS Excel» или выберите соответствующий пункт меню.
 - При этом будет создан и открыт в Excel файл «Сведения о заказах.xls», содержащий данные из выбранного запроса.
 - Введите в ячейку «I2» формулу «=D2*E2*(1-F2)».
 - Выделите мышью ячейку «I2» и выполните команду копирования в буфер обмена.

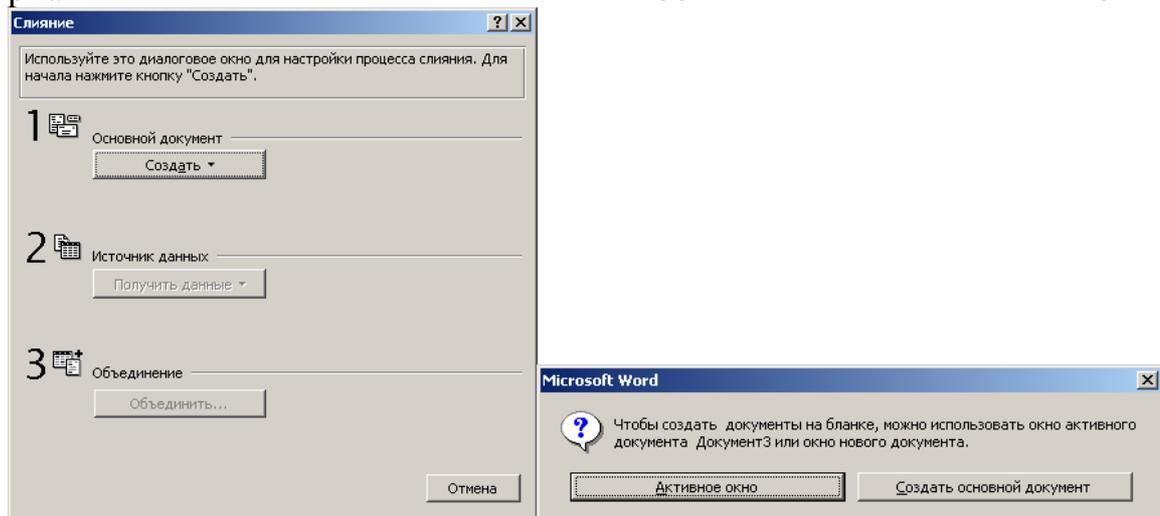
- Выделите в столбце «I» ячейки начиная с «I3» до нижней границы данных.
 - Выполните команду вставки из буфера обмена. При этом значения, которые отображаются в ячейках столбца «I», должны стать равными соответствующим значениям столбца «G».
 - Выделите в столбце «I» ячейки начиная с «I2» до нижней границы данных и нажмите кнопку «Автосумма»  на панели инструментов Excel. В столбце «I» снизу должно отобразиться значение, равное сумме всех заключенных договоров.
 - Сохраните составленный документ, закройте Excel и вернитесь к работе с Access.
4. Выполните экспорт данных из таблицы «Сотрудники» в текстовый файл, записи должны разделяться символом табуляции.
- Нажмите правой кнопкой мыши на имени таблицы «Сотрудники», в контекстном меню выберите пункт «Экспорт...».
 - В диалоговом окне выберите тип сохраняемого файла «Текстовые файлы», задайте местоположение и имя сохраняемого файла («Сотрудники.txt») и нажмите кнопку «Сохранить».
 - Выберите вариант формата экспорта «с разделителями» и нажмите кнопку «Далее».
 - Выберите в качестве разделителя «Tab», поместите флажок в поле «Включить имена полей в первой строке» и нажмите кнопку «Готово».
 - Используя доступное средство работы с текстом (например, «Блокнот») просмотрите содержимое созданного файла.
5. Создайте новую базу данных «Мои сотрудники». Создайте в ней таблицу «Список моих сотрудников», импортировав в нее данные из текстового файла.
- В своем рабочем каталоге создайте базу данных «Мои сотрудники».
 - В окне базы данных откройте раздел «Таблицы».
 - Нажмите правой кнопкой мыши в окне базы данных и в контекстном меню выберите пункт «Импорт...».
 - Также можно использовать пункт меню «Файл → Внешние данные → Импорт...».
 - Выберите тип файлов «Текстовые файлы» и откройте файл «Сотрудники.txt».
 - Выберите вариант формата «с разделителями» и нажмите кнопку «Далее».
 - Выберите в качестве разделителя «Tab», поместите флажок в поле «Первая строка содержит имена полей» и нажмите кнопку «Далее».
 - Выберите вариант сохранения «В новой таблице». Нажмите кнопку «Дополнительно».
 - В таблице описания полей для поля «КодСотрудника» определите тип данных «Длинное целое», в графе «Индекс» выберите «Да (Совпадения не допускаются)».

Обратите внимание, что невозможно для импортируемых данных определить тип данных «Счетчик». Другие способы – исключить поле «КодСотрудника» из списка импорта и затем создать его самостоятельно в режиме конструктора или создать готовую таблицу нужной структуры и импортировать данные в нее.

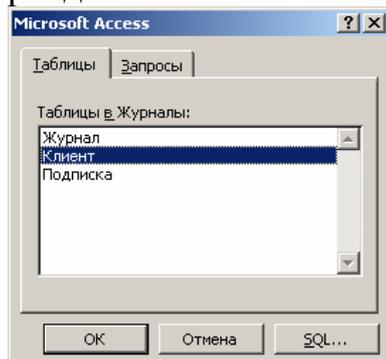
- Для поля «Фотография» поставьте флажок в графе «Пропустить». Нажмите кнопку «ОК». Дважды нажмите кнопку «Далее».
- Определите в качестве ключевого поля «КодСотрудника» и нажмите кнопку «Далее».
- Задайте в качестве имени таблицы «Список моих сотрудников» и нажмите кнопку «Готово».

6. Создайте в MS Word документ «Справка» (вид документа приведен ниже), выполните слияние, инициировав его из MS Word, подставьте в документ поля из таблицы «Клиент» базы данных «Журналы».

- Завершите программу MS Access. Запустите MS Word и создайте в нем бланк документа в соответствии с приведенным ниже примером.
- Выберите в меню Word пункт «Сервис → Слияние...».
- В открывшемся диалоговом окне выберите вид основного документа «Документы на бланке...», на вопрос о создании документа укажите вариант «Активное окно».



- Для определения источника данных выберите пункт «Открыть источник данных...», укажите тип файлов «Базы данных MS Access» и откройте файл базы данных «Журналы».
- В появившемся диалоговом окне выберите таблицу «Клиент» и затем перейдите к правке основного документа.



- Используйте инструмент **«Добавить поле слияния»**  для размещения полей слияния в бланке документа.
- Нажмите кнопку **«Поля/данные»**  на панели инструментов Word для отображения в бланке значений полей вместо их названий.
- Используйте кнопки навигации  для перемещения по записям источника данных. Убедитесь, что при этом в бланке отображаются фамилия, имя и отчество из соответствующей записи.
- Сохраните созданный документ Word под именем **«Справка»**. В дальнейшем этот документ может быть использован для печати справок о выбранных клиентах.

Вид бланка справки:

Справка

Выдана в том, что «Фамилия» «Имя» «Отчество» является постоянным и уважаемым подписчиком нашего издательства.

Генеральный директор издательства «Йцукен»
Фролов Виктор Николаевич

Самостоятельная работа:

1. Добавьте в таблицу **«Клиент»** базы данных **«Журналы»** поле **«Пол клиента»**, имеющее логический тип. Одно из его возможных значений будет обозначать мужской пол, а другое – женский.

2. Создайте запрос **«Клиенты с обращением»**, выводящий список клиентов и способ обращения к ним, зависящий от их пола (при составлении запроса имеет смысл использовать функцию ПФ). Для мужчин будет обращение **«Уважаемый»**, для женщин **«Уважаемая»**.

3. Создайте бланк поздравительного письма клиенту следующего вида:

«Обращение» «Имя» «Отчество»! Поздравляю Вас с наступающим Новым Годом. Желаю успехов и ...

Генеральный директор издательства «Йцукен»
Фролов Виктор Николаевич

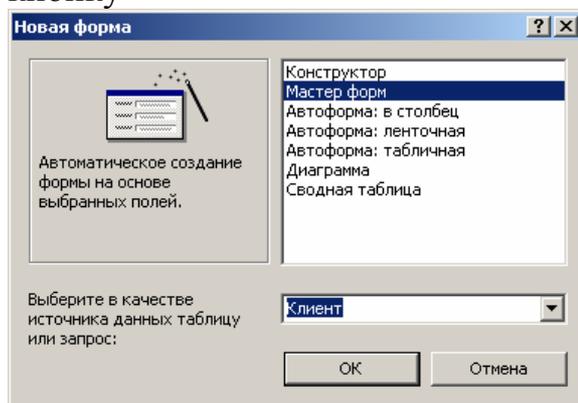
Задание 5. Создание и редактирование форм и отчетов

Цель задания: Научиться создавать и редактировать формы и отчеты. Используется разработанная ранее база данных «Журналы».

Последовательность действий:

1. Запустите Microsoft Access и откройте базу данных «Журналы».
2. Создайте форму для просмотра и редактирования списка клиентов, используя мастер форм.

- В списке типов объектов окна базы данных выберите пункт «**Формы**».
- Нажмите кнопку «**Создать**»  в окне базы данных.
- В появившемся диалоговом окне выберите пункт «**Мастер форм**».
- В качестве источника данных выберите таблицу «**Клиент**» и нажмите кнопку «**ОК**».



- В открывшемся окне скопируйте имена всех полей в список выбранных полей и нажмите кнопку «**Далее**».
- Выберите тип формы «**Выровненный**» и нажмите кнопку «**Далее**».
- Выберите фоновый рисунок из предложенного списка и нажмите кнопку «**Далее**».
- Задайте имя формы «**Список клиентов**» и нажмите кнопку «**Готово**».
- Просмотрите данные, отображаемые в форме.
- Следует скрыть отображаемую в левой части окна область выделения.

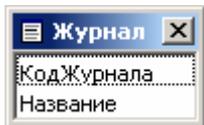
Для этого перейдите в режим конструктора ().

- Нажмите правой кнопкой мыши на квадрат в левом верхнем углу окна конструктора () в контекстном меню выберите пункт «**Свойства**». Откроется окно редактирования свойств формы.
- Выберите вкладку «**Макет**» и замените значение свойства «**Область выделения**» на «**Нет**».
- Закройте окно свойств формы и перейдите в режим просмотра формы ().
- Убедитесь, что область выделения не отображается.
- Закройте окно формы, подтвердив сохранение изменений.

3. Создайте форму для просмотра и изменения списка журналов, используя конструктор.

- Щелкните мышкой по ярлыку «**Создание формы в режиме конструктора**». Откроется окно конструктора форм.
- Нажмите правой кнопкой мыши в окне конструктора и в контекстном меню выберите пункт «**Свойства**». Откроется окно редактора свойств.

- Нажмите мышью на квадрат в левом верхнем углу окна . При этом в окне редактора свойств отобразятся свойства создаваемой формы.
- Выберите вкладку «Данные» и в качестве значения свойства «Источник записей» выберите имя таблицы «Журнал». При этом должно открыться окно «Список полей», содержащее имена полей выбранной таблицы.



При необходимости окно «Список полей» может быть показано или скрыто с помощью пункта меню «Вид → Список полей».

- С помощью мыши поместите поля «КодЖурнала» и «Название» в бланк формы. Выровняйте положение и размер полей и надписей. Для более точного позиционирования элемента можно, нажав на него мышью, в окне редактора свойств задать значения свойств «От левого края», «От верхнего края», «Ширина» и «Высота».

- Перейдите в режим просмотра формы ().

- Закройте окно формы, сохранив ее под именем «Список журналов».

4. Добавьте в форму «Список журналов» подчиненную форму, отображающую список подписчиков выбранного журнала. Для этого следует создать запрос, объединяющий имена клиентов и коды журналов, на которые они подписаны, создать форму на основе построенного запроса и подключить ее в качестве подчиненной к форме «Список журналов».

- Создайте запрос с именем «Клиенты с кодами журналов», на основе таблиц «Клиент» и «Подписка» формирующий таблицу из двух полей. Первое поле – «Журнал», из таблицы «Подписка». Второе поле – «ФИО», образуется следующим выражением: «ФИО: [Фамилия] & " " & [Имя] & " " & [Отчество]» (фамилия, имя и отчество клиента объединяются в одну строку).
- Запустите конструктор форм для создания новой формы.
- Задайте в качестве значения свойства «Источник записей» формы имя запроса «Клиенты с кодами журналов».
- Во вкладке «Макет» для свойства формы «Режим по умолчанию» задайте значение «Ленточная форма».
- Скройте область выделения формы.
- Скройте панель навигации формы (свойство «Кнопки перехода»).
- Во вкладке «Данные» назначьте значение «Нет» для свойств «Разрешить изменение», «Разрешить удаление» и «Разрешить добавление».
- Добавьте в область данных формы поле «ФИО». Удалите его текстовую надпись. Настройте его ширину и положение. Уменьшите размеры области данных, передвинув мышью ее границы.
- Перейдите в режим просмотра формы ().
- Закройте окно формы, сохранив ее под именем «Подписчики – подчиненная».

- Откройте форму «**Список журналов**» для редактирования в конструкторе.
- Убедитесь, что доступна панель элементов формы (для ее открытия используется пункт меню «**Вид → Панель элементов**»).
- Установите кнопку «**Мастера**»  на панели элементов в верхнее положение.
- Поместите элемент «**Подчиненная форма/отчет**» в область данных формы.
- Измените текст надписи над размещенным элементом на «**Подписчики**».
- Выделите мышью объект подчиненной формы. В окне редактора свойств определите значения следующих свойств подчиненной формы: «**Источник записей**» - имя формы «**Подписчики - подчиненная**», «**Подчиненные поля**» - имя поля «**Журнал**», «**Основные поля**» - имя поля «**Код Журнала**».

Подчиненное – поле из источника записей подчиненной формы, основное – поле из источника записей главной формы. С помощью этих полей устанавливается связь между данными в главной и подчиненной формах.

- Настройте расположение и размер объекта подчиненной формы.
- Проверьте работу формы «**Список журналов**». Для каждого поля с данными о журнале подчиненная форма должна отображать список клиентов, подписанных на этот журнал.
- Закройте окно формы, сохранив внесенные изменения.

5. Создайте форму для просмотра, добавления, изменения и удаления записей о подписке, используя конструктор.

- В качестве источника записей формы используйте таблицу «**Подписка**».
- Убедитесь, что при работе в конструкторе панель элементов доступна и

кнопка «**Мастера**»  находится в верхнем положении.

- Разместите в области данных формы два элемента «**Поле со списком**»



- Задайте значения свойства «**Данные**» для каждого из списков. Для первого списка – имя поля «**Клиент**», для второго – «**Журнал**».
- Задайте значения свойства «**Источник строк**» для каждого из списков. Для первого списка – имя таблицы «**Клиент**», для второго – «**Журнал**».
- Посмотрите, как работает созданная форма. В каждом из списков отображается содержимое первого столбца таблицы, использованной в качестве источника строк («**КодКлиента**» и «**КодЖурнала**» соответственно). Требуется настроить свойства подстановки таким образом, чтобы отображались не числовые поля, а текстовые («**Фамилия**» и «**Название**» соответственно).
- Для первого списка во вкладке «**Данные**» свойство «**Присоединенный столбец**» имеет значение «**1**». Этот номер соответствует столбцу «**КодКлиента**», значения из которого должны подставляться в поле «**Клиент**» таблицы «**Подписка**». Это значение назначено верно.

- Во вкладке «**Макет**» присвойте свойству «**Число столбцов**» значение «**2**», свойству «**Ширина столбцов**» - значение «**0см**». При подстановке используются первые 2 столбца таблицы, причем первый из них («**КодКлиента**») не виден, поскольку для него задана ширина «**0**».
 - Посмотрите, как работает форма. Первый список отображает фамилии клиентов, при этом в таблицу «**Подписка**» подставляются их коды.
 - По аналогии задайте свойства для второго списка. Вместо того, чтобы в качестве источника строк использовать таблицы напрямую, можно сформировать для этой цели специальные запросы. Например, запрос, в первом столбце выводящий код клиента, а во втором – ФИО.
 - Переведите кнопку «**Мастера**» на панели элементов в «утопленное» положение .
 - Разместите в области данных формы элемент «**Кнопка**» .
 - В открывшемся окне мастера выберите категорию «**Обработка записей**» и действие «**Добавить запись**». Нажмите кнопку «**Далее**». Выберите рисунок для кнопки из предложенного списка или определите текстовую надпись. Нажмите кнопку «**Готово**». При нажатии на данную кнопку форма будет ожидать ввода данных для новой записи.
 - Аналогично, добавьте кнопку для удаления текущей записи.
 - Сохраните созданную форму под именем «**Подписка – форма**» и закройте окно формы.
6. Создайте отчет, выводящий список журналов и для каждого из журналов – список его подписчиков.
- Создайте запрос на основе таблиц «**Клиент**», «**Подписка**» и «**Журнал**», выводящий название журнала, фамилию, имя и отчество клиента, отсортированные по названию и фамилии.
 - В разделе «**Отчеты**» окна базы данных щелкните по ярлыку «**Создание отчета в режиме конструктора**».
 - В качестве источника записей для отчета укажите созданный ранее запрос.
 - Уменьшите высоту нижнего колонтитула до нуля, в область верхнего колонтитула поместите надпись с текстом «**Журналы с подписчиками**».
 - Используя пункт меню «**Вид → Сортировка и группировка**», откройте окно свойств сортировки и группировки.
 - В графе «**Поле/выражение**» выберите имя поля «**Название**». В графе «**Порядок сортировки**» установите значение «**По возрастанию**». Для свойства «**Заголовок группы**» задайте значение «**Да**».
 - В бланке формы, в раздел «**Заголовок группы 'Название'**» поместите элемент, связанный с полем «**Название**». В область данных поместите поля «**Фамилия**», «**Имя**» и «**Отчество**».

- Определите размеры шрифтов для каждого из элементов: для надписи колонтитула – 18, для поля «**Название**» - 14, для полей «**Фамилия**», «**Имя**» и «**Отчество**» - 12.
- Отрегулируйте размеры и расположение элементов.
- Просмотрите результат работы отчета, нажав на кнопку «**Просмотр**» .
- Закройте окно отчета, сохранив его под именем «**Журналы с подписчиками - отчет**».

Самостоятельная работа:

1. Ознакомьтесь с работой следующих форм базы данных «Борей»:

- Сотрудники
- Сотрудники (с переводом страницы)
- Главная кнопочная форма

Исследуйте их, используя конструктор форм. Выясните, какие значения присвоены свойствам форм и их элементов. В частности, свойствам «Источник записей», «Источник строк», свойствам обработки событий (для кнопок).

2. Ознакомьтесь с работой отчетов «Наклейки для клиентов», «Продажи по сотрудникам и странам», «Суммы продаж по годам» базы данных «Борей». Исследуйте их используя конструктор отчетов.

Обратите внимание, что на внешний вид отчета (в частности, на количество столбцов) влияют параметры страницы (пункт меню «**Файл → Параметры страницы...**»).

3. Примените обнаруженные приемы работы для создания новых форм и отчетов в базе данных «Журналы». Создайте форму – аналог главной кнопочной формы.

Литература

1. Карпов Б. Microsoft Access 2000: Справочник. // СПб: Питер, 2001.
2. Кузнецов С.Д. Основы современных баз данных. // в электронной форме: <http://citforum.ru/database/osbd/contents.shtml>.
3. Ролланд Ф.Д. Основные концепции баз данных // М.- СПб - Киев: «Вильямс», 2002.
4. Справка по Microsoft Access (входит в состав пакета Access).
5. Форт С., Хоуи Т., Релстон Дж. Программирование в среде Access 2000. Энциклопедия пользователя: Киев: Издательство «ДиаСофт», 2000.
6. Microsoft Access 2000. Шаг за шагом: Практ. пособ. // М.: Издательство ЭКОМ, 2002.

История кафедры КОТ неразрывно связана с развитием университета в последнее десятилетие, с превращением ИТМО в ведущий компьютерный вуз страны. Научную и научно-методическую основу создания и развития кафедры определили два события в жизни университета:

1999 год: создание центра дистанционного обучения (ЦДО) университета.

2000 год: открытие Санкт-Петербургского регионального центра «Федерации Интернет Образования» (СПб РЦ ФИО).

ЦДО ИТМО был создан как полигон для разработки дистанционных курсов университета. Сегодня ЦДО стал не только научно-методическим центром ИТМО, но и реальным участником учебного процесса вуза. Аттестации студентов, виртуальные лаборатории в среде системы ДО ИТМО стали неотъемлемой частью учебного процесса различных кафедр университета, разумеется, и кафедры КОТ.

СПб РЦ ФИО создан в университете в рамках негосударственного образовательного проекта «Поколение.ги» для массовой переподготовки работников среднего образования в области интернет-технологий. Успешная работа в этом направлении позволила обучить к октябрю 2004 года более 5000 учителей. Методики и программы обучения интернет-технологиям на курсах СПб РЦ ФИО стали основой многих студенческих учебных программ.

2001 год: открытие на факультете ИТП кафедры КОТ.

Вначале кафедра КОТ обеспечивала учебный процесс по специальности 071900 – «Информационные системы и технологии». Дисциплины кафедры: «Информатика», «Информационные технологии», «Web-программирование».

2003 год:

- открытие в ИТМО новой инженерной специальности 073700 – «Информационные технологии в образовании»;
- переход кафедры КОТ в разряд выпускающих кафедр;
- разработан и утвержден учебный план по специализации «Аппаратно-программные комплексы образовательных систем»;
- произведен первый набор студентов специальности 073700.

С этого времени развитие кафедры подчинено главной цели: качественной подготовке специалистов высшей квалификации в области информационных технологий для сферы образования.

Алексей Алексеевич Бобцов
Виталий Валерьевич Шиегин

Банки и базы данных. Основы работы с MS Access.
Часть 1 (для пользователей)

Учебное пособие

В авторской редакции
компьютерный набор и верстка А.А. Бобцов
Редакционно-издательский отдел СПбГУ ИТМО
Зав. РИО Н.Ф. Гусарова
Лицензия ИД № 00408 от 05.11.99
Подписано к печати
Тираж 100 экз. Заказ № 851
Отпечатано на ризографе.