

# БАЗА ДАННЫХ

## Общие положения

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

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если данные структурированы.

# Не Структурированные данные

- **Структурирование** — это введение соглашений о способах представления данных.
- Неструктурированными называют данные, записанные, например, в текстовом файле.
- 
- **Пример** . На рис. 1 пример неструктурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя, отчество и год рождения).
- Легко убедиться, что сложно организовать поиск необходимых данных, хранящихся в неструктурированном виде, а упорядочить подобную информацию практически не представляется реальным.
- Личное дело № 16493, Сергеев Петр Михайлович, дата рождения 1 января 1976 г.; Л/д № 16593, Петрова Анна Владимировна, дата рожд. 15 марта 1975 г.; № личн. дела 16693, д.р. 14.04.76, Анохин Андрей Борисович.
- 
- **Рис. 15.1.** Пример неструктурированных данных

# Структурированные данные

- Чтобы автоматизировать поиск и систематизировать эти данные, необходимо выработать определенные соглашения о способах представления данных, т.е. дату рождения нужно записывать одинаково для каждого студента, она должна иметь одинаковую длину и определенное место среди остальной информации. Эти же замечания справедливы и для остальных данных (номер личного дела, фамилия, имя, отчество).
- **Пример 15.2.** После проведения несложной структуризации с информацией, указанной в примере
- (рис. 1), она будет выглядеть так, как это показано на рис. 2.
- 

№ личного дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75

- **Рис. 15.2.** Пример структурированных данных

# Определение Базы данных и СУБД

- Пользователями базы данных могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые *конечными пользователями*.
- В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария — *системы управления базами данных*.
- **База данных (БД)** — это поименованная совокупность структурированных данных, относящихся к определенной предметной области.
- **Система управления базами данных (СУБД)** — это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.
- 
- Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данными, хранимыми в базе.

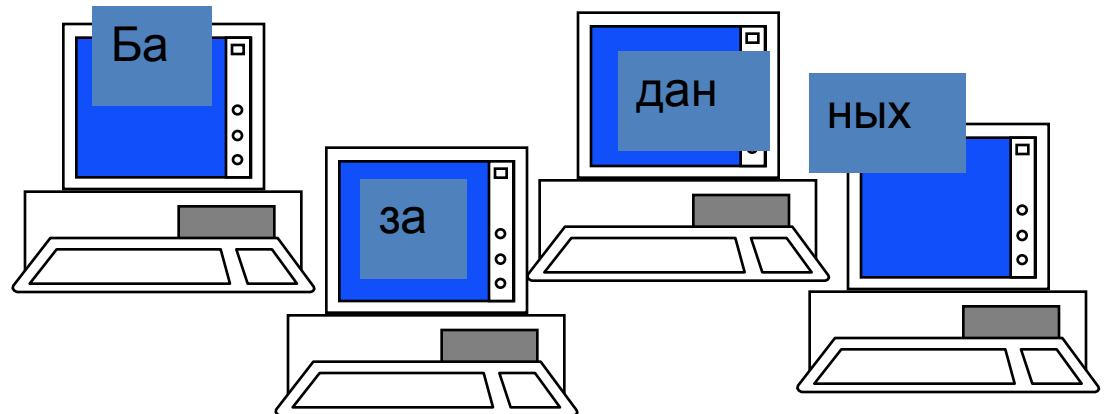
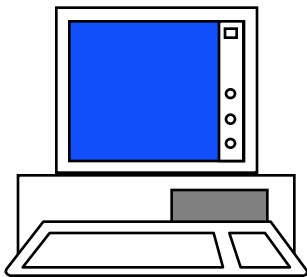
# Классификация баз данных по *технологии обработки*

- По *технологии обработки* данных базы данных подразделяются на централизованные и распределенные.
- *Централизованная база* данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.
- *Распределенная база* данных состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

# Классификация баз данных

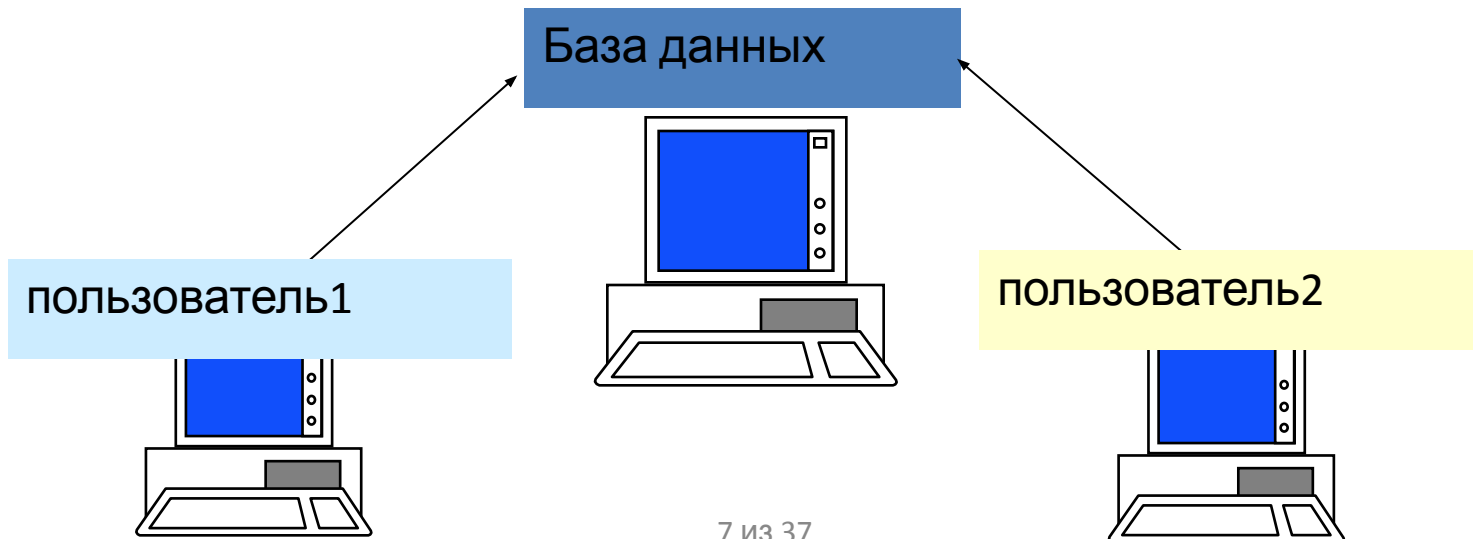
- По технологии обработки данных БД подразделяют на **централизованные** и **распределенные**.

База данных



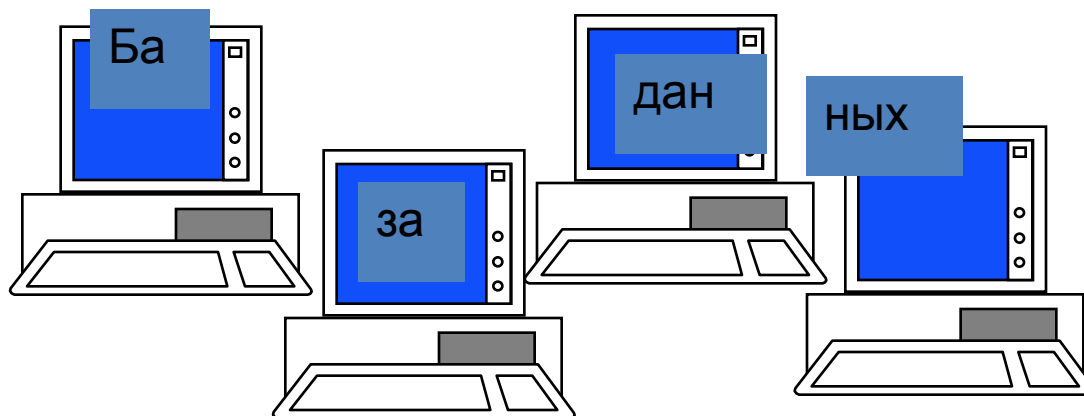
# Централизованная БД

- Хранится в памяти одной вычислительной системы.
- Если эта вычислительная система является компонентом сети ЭВМ, возможен **распределенный доступ** к такой базе.



# Распределенная БД

- Состоит из нескольких, возможно пересекающихся или дублирующих друг друга частей, хранимых на **разных** ЭВМ. Работа с такой БД осуществляется с помощью **системы управления распределенной БД (СУРБД)**.



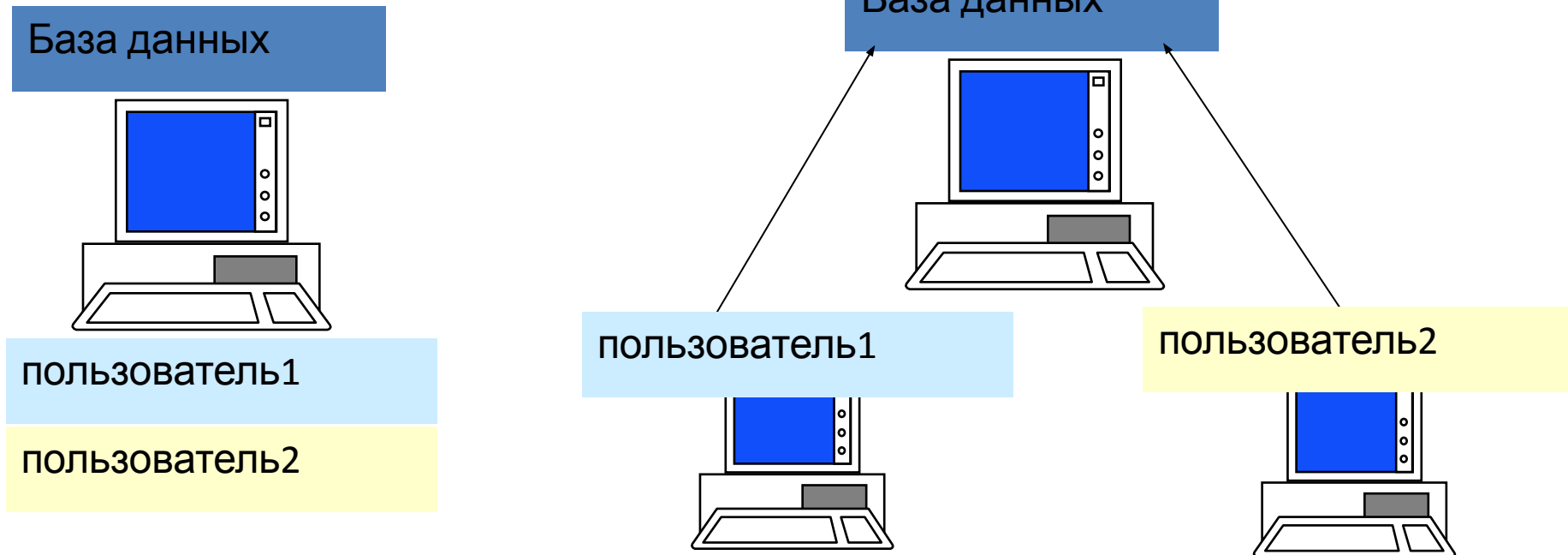


# Классификация БД по способу доступа к данным

- По *способу доступа* к данным базы данных разделяются на базы данных с *локальным доступом* и базы данных с *удаленным (сетевым) доступом*.
- Системы централизованных баз данных с сетевым доступом предполагают различные *архитектуры* подобных систем:
  - файл-сервер;
  - клиент-сервер.

# Классификация баз данных

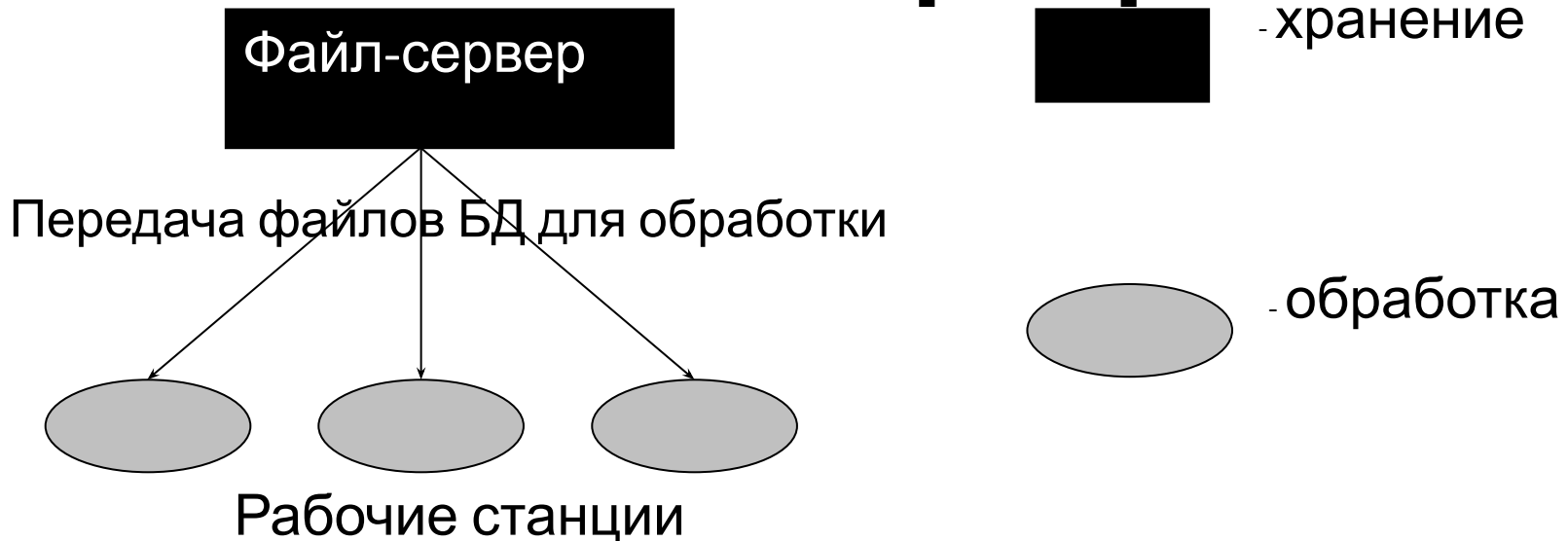
- По способу доступа к данным БД разделяются на БД с **локальным доступом** и БД с **удаленным (сетевым) доступом**.



# Файл-сервер

- **Файл-сервер.** Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монопольно. Концепция файл-сервер условно отображена на рис. 3.

# Файл-сервер

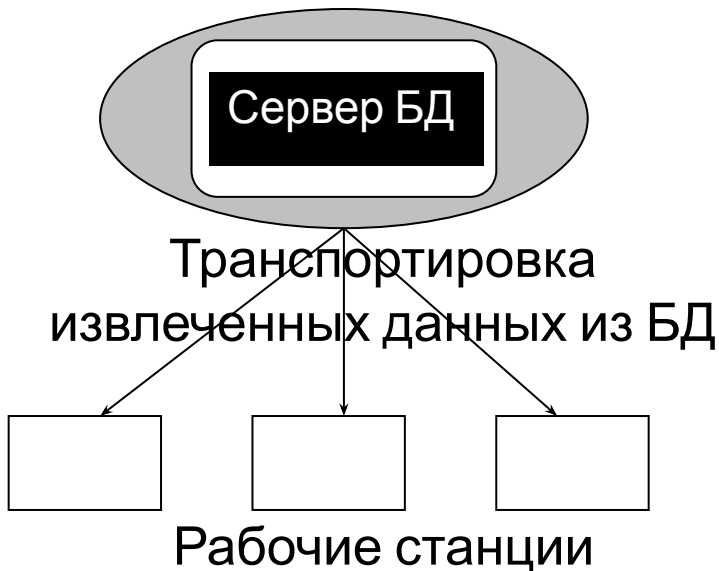


- Одна машина в сети выделяется как центральная (сервер файлов), где **хранится, используемая всеми БД**
- **Файлы БД** согласно запросам **передаются на рабочие станции**, где производится обработка
- При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает.

# Клиент-сервер

- **Клиент-сервер.** В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом
- (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL. Концепция клиент-сервер условно изображена на рис. 4.

# Клиент-сервер



- Сервер БД обеспечивает хранение и обработку данных
- Запрос на данные, выдаваемый клиентом рабочей станции, порождает поиск и извлечение данных на сервере.

**Извлеченные данные** (но не файлы БД) **поступают по сети к клиенту.**

# Структурные элементы базы данных

- Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл .
- **Поле** — элементарная единица логической организации данных, которая соответствует неделимой единице информации — реквизиту. Для описания поля используются следующие *характеристики*:
  - *имя*, например, Фамилия, Имя, Отчество, Дата рождения;
  - *тип*, например, символьный, числовой, календарный;
  - *длина*, например, 15 байт, причем будет определяться максимально возможным количеством символов;
  - *точность* для числовых данных, например два десятичных знака для отображения дробной части числа.
- **Запись** — совокупность логически связанных полей. Экземпляр записи — отдельная реализация записи, содержащая конкретные значения ее полей.
- **Файл (таблица)** — совокупность экземпляров записей одной структуры.

# ВИДЫ МОДЕЛЕЙ ДАННЫХ

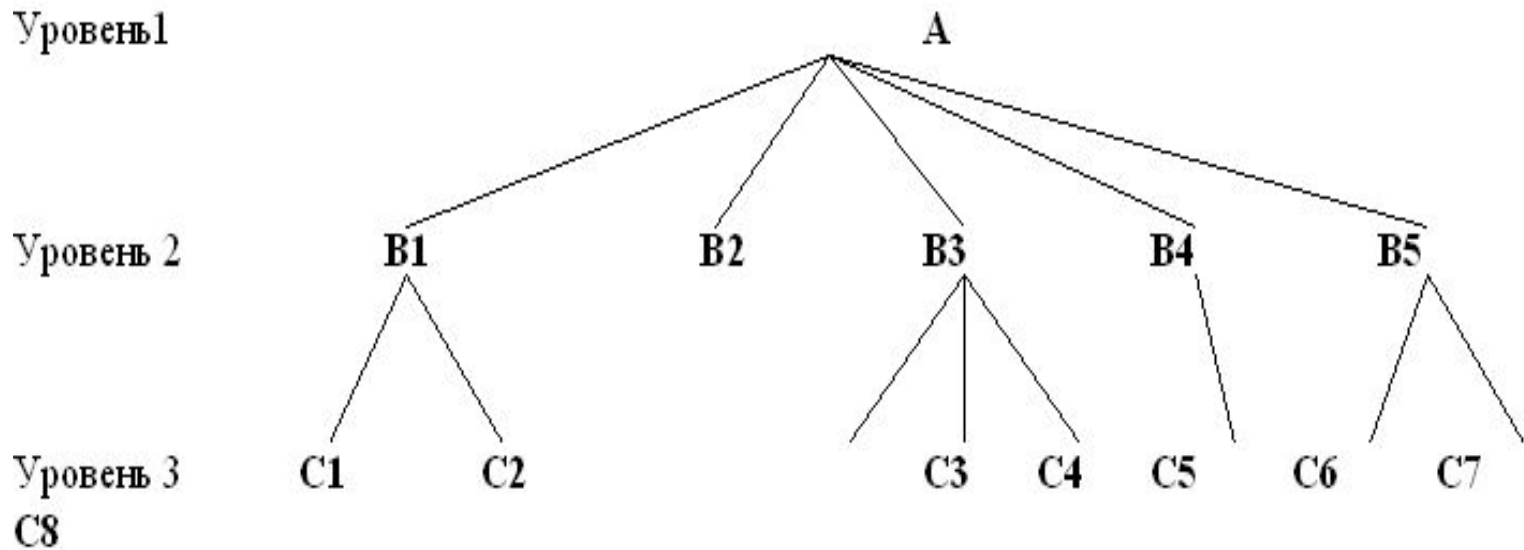
- Ядром любой базы данных является модель данных. Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.
- **Модель данных** — совокупность структур данных и операций их обработки.
- СУБД основывается на использовании иерархической, сетевой или, реляционной модели, на комбинации этих моделей или на некотором их подмножестве [.
- Существует три основных типа модели данных:
- **иерархическая,**
- **сетевая;**
- **реляционная.**



# Иерархическая модель данных

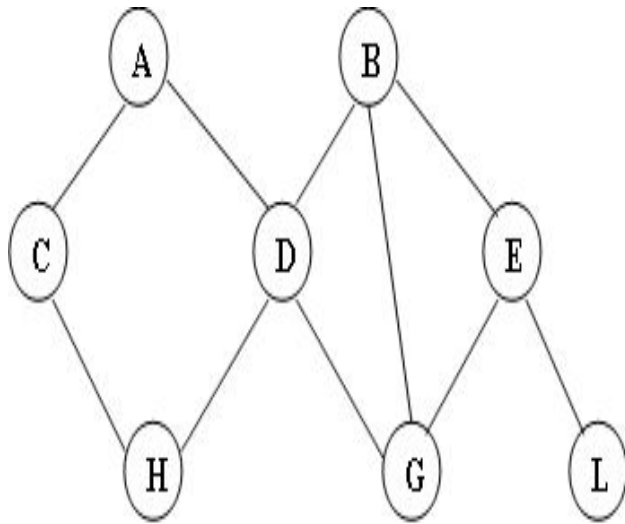
- Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево), вид которого представлен на рис.
- К основным понятиям иерархической структуры относятся: **уровень, элемент (узел), связь.**
- *Узел* — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа.
- Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне.
- Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне.
- Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

# Графическое изображение иерархической структуры БД



К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис., для записи C4 путь проходит через записи A и B3.

# Сетевая модель данных



сетевая структура базы данных в виде графа

- В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.
- На рис. изображена сетевая структура базы данных в виде графа.
- **Пример** . Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о студентах, участвующих в научно-исследовательских работах (НИРС). Возможно участие одного студента в нескольких НИРС, а также участие нескольких студентов в разработке одной НИРС.

•

# Пример сетевой структуры БД

- **Сетевая БД** отличается большей гибкостью, т.к. в ней существует возможность устанавливать **дополнительно к вертикальным иерархическим связям горизонтальные связи..**
- **Сетевая структура данных представляет собой произвольный граф.**
- **Пример 15.5.** Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о студентах, участвующих в научно-исследовательских работах (НИРС). Возможно участие одного студента в нескольких НИРС, а также участие нескольких студентов в разработке одной НИРС.
- Графическое изображение описанной в примере сетевой структуры, состоящей только из двух типов записей. Единственное отношение представляет собой сложную связь между записями в обоих направлениях.

# Реляционная модель данных

- *Понятие реляционный* (англ. *relation* — отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.
- Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

# Свойства реляционных таблиц

- Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая *реляционная таблица* представляет собой двумерный массив и обладает следующими свойствами:
  - • каждый элемент таблицы — один элемент данных;
  - • все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
  - • каждый столбец имеет уникальное имя;
  - • одинаковые строки в таблице отсутствуют;
  - • порядок следования строк и столбцов может быть произвольным

# Пример реляционной БД

- 

№ личного дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75

- Отношения в БД представлены в виде *таблиц*, строки которых соответствуют кортежам или *записям*, а столбцы — атрибутам отношений, доменам, *полям*.

-

# Виды ключей

- Поле, каждое значение которого однозначно определяет соответствующую запись, называется *простым ключом* (ключевым полем).
- Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет *составной ключ*. В данном случае ключевым полем таблицы является "№ личного дела".
- В структуре записи файла указываются поля, значения которых являются *ключами: первичными* (ПК), которые идентифицируют экземпляр записи, и *вторичными* (ВК), которые выполняют роль поисковых или группировочных признаков (по значению вторичного ключа можно найти несколько записей).
-



# Необходимость ключей

- Каждая таблица должна иметь **первичный ключ**, т.е. одно или несколько полей, которые однозначно определяют каждую запись таблицы.
- **Внешний ключ** – поле (или поля), которое является первичным ключом в другой таблице. Процесс установления первичных и внешних ключей, а также создания связей называют **нормализацией**. Это приводит к достижению одной из важнейших целей – **уменьшению избыточности** информации в БД.
- Связь между таблицами устанавливается через поля, совпадающие по значению. Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы **внешний ключ** - ключ второй таблицы.

# Информационный объект

- *Информационный объект* — это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных *реквизитов* (информационных элементов). Такими сущностями для информационных объектов могут служить: цех, склад, материал, вуз, студент, сдача экзаменов и т.д.
- Информационный объект определенного реквизитного состава и структуры образует класс (тип), которому присваивается уникальное имя (символьное обозначение), например Студент, Сессия, Стипендия.
- Информационный объект имеет множество реализаций — экземпляров, каждый из которых представлен совокупностью конкретных значений реквизитов и идентифицируется значением ключа (простого — один реквизит или составного — несколько реквизитов). Остальные реквизиты информационного объекта являются описательными. При этом одни и те же реквизиты в одних информационных объектах могут быть ключевыми, а в других — описательными. Информационный объект может иметь несколько ключей

# Понятие нормализации отношений

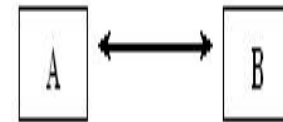
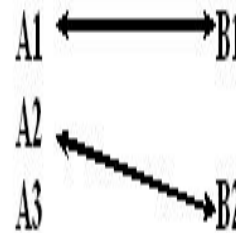
- Одни и те же данные могут группироваться в таблицы (отношения) различными способами, т.е. возможна организация различных наборов отношений взаимосвязанных информационных объектов. Группировка атрибутов в отношениях должна быть рациональной, т.е. минимизирующей дублирование данных и упрощающей процедуры их обработки и обновления.
- Определенный набор отношений обладает лучшими свойствами при включении, модификации, удалении данных, чем все остальные возможные наборы отношений, если он отвечает требованиям нормализации отношений .
- **Нормализация отношений** — формальный аппарат ограничений на формирование отношений (таблиц), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение (ввод, корректировку) базы данных.

# ТИПЫ СВЯЗЕЙ

- Все информационные объекты предметной области связаны между собой. Различаются *связи* нескольких типов, для которых введены следующие обозначения:
  - • один к одному (1:1);
  - • один ко многим (1:M);
  - • многие ко многим (M:M).

# Связь *один к одному*

- Связь *один к одному* (1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот.

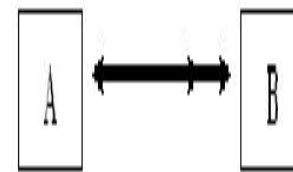
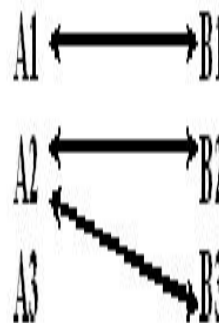


Примером связи 1:1 может служить связь между информационными объектами СТУДЕНТ и СЕССИЯ:

СТУДЕНТ	СЕССИЯ
---------	--------

# СВЯЗЬ *один ко многим*

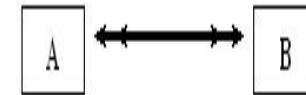
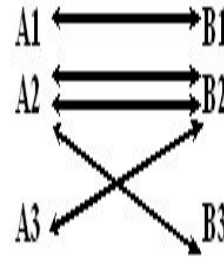
- При связи *один ко многим* (1:M) одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А.



Примером связи 1:M служит связь между информационными объектами СТИПЕНДИЯ и СЕССИЯ:  
СТИПЕНДИЯ                      СЕССИЯ  
Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

# Связь *многие ко многим*

- Связь *многие ко многим* (M:M) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В и наоборот.



Примером данного отношения Служит связь между информационными объектами СТУДЕНТ и ПРЕПОДАВАТЕЛЬ:  
СТУДЕНТ « »ПРЕПОДАВАТЕЛЬ  
Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

# СУБД

## *Системой управления базами данных (СУБД)*

называется совокупность программных средств, с помощью которых создаются и обрабатываются данные.

### **Основные функции СУБД:**

- 1. Определение данных** - позволяет определить, какая именно информация будет храниться в БД, задать свойства данных, их тип, а также указать, как эти данные связаны между собой.
- 2. Обработка данных** - данные могут обрабатываться самыми различными способами. Можно выбирать любые поля, фильтровать и сортировать данные. Можно объединять данные с другой, связанной с ними информацией, и вычислять итоговые значения.
- 3. Управление данными** - можно указать, кому разрешен доступ к данным, корректировать их или добавлять новую информацию. Можно также определять правила коллективного доступа.



# Объекты MS Access

- Основными объектами MS Access являются:
  - таблица;
  - форма;
  - запрос;
  - отчет;
  - страницы;
  - макросы и модули.

# Таблицы

- **Таблицы** – это основные объекты любой базы данных. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства).

# Запросы

- **Запросы.** Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.
- Из соображений безопасности, чем меньше доступа к базовым таблицам имеют конечные пользователи, тем лучше. Во-первых, снижается риск того, что неумелыми действиями они повредят данные в таблицах. Во-вторых, предоставив разным пользователям разные запросы, можно эффективно разграничить их доступ к данным в строгом соответствии с кругом персональных обязанностей. В банках, например, одни сотрудники имеют доступ к таблицам данных о клиентах, другие — к их расчетным счетам, третьи — к таблицам активов банка. Если и есть специальные службы, имеющие доступ ко всем информационным ресурсам банка (с целью контроля и анализа), то они лишены средств для внесения изменений — все сделано так, чтобы один человек не мог совершить фиктивную операцию, независимо от того, какую должность он занимает. В базе данных, имеющей правильно организованную структуру, для совершения противоправных действий необходим сговор нескольких участников, а такие действия пресекаются не программными, а традиционными средствами обеспечения безопасности.

# Особенность запросов

- Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную *результатирующую таблицу*. Если хотят подчеркнуть факт «временности» этой таблицы, то ее еще называют *моментальным снимком*. Когда мы работаем с основными таблицами базы, мы физически имеем дело с жестким диском, то есть с очень медленным устройством (напомним, что это связано с особенностью сохранения данных, описанной выше). Когда же на основании запроса мы получаем результирующую таблицу, то имеем дело с электронной таблицей, не имеющей аналога на жестком диске, — это только образ отобранных полей и записей. Разумеется, работа с «образом» происходит гораздо быстрее и эффективнее — это еще одно основание для того, чтобы широко использовать запросы.
- Недостатком упорядоченных табличных структур является сложность их обновления, поскольку при внесении новых записей нарушается упорядоченность — приходится переделывать всю таблицу. В системах управления базами данных и эта проблема решается благодаря запросам.
- Основным принцип состоит в том, что от базовых таблиц никакой упорядоченности не требуется. Все записи в основные таблицы вносятся только в *естественном* порядке по мере их поступления, то есть в неупорядоченном виде. Если же пользователю надо видеть данные, отсортированные или отфильтрованные по тому или иному принципу, он просто использует соответствующий запрос.. Если нужного запроса нет, он обращается к проектировщику и просит его такой запрос сделать и предоставить.

# Формы

- Если запросы — это специальные средства для отбора и анализа данных, то формы — это средства для ввода данных. Смысл их тот же — предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочие) для автоматизации ввода. Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графическими средствами так, чтобы она повторяла оформление бланка — это заметно упрощает работу наборщика, снижает его утомление и предотвращает появление печатных ошибок. На сопроводительном рисунке приведен пример простейшей формы для ввода данных.
- С помощью форм данные можно не только вводить, но и отображать. Запросы тоже отображают данные, но делают это в виде результирующей таблицы, не имеющей почти никаких средств оформления. При выводе данных с помощью форм можно применять специальные средства их оформления.

# Отчеты

- **Отчеты.** По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на печатающее устройство (принтер). В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, служебная информация о времени создания отчета и т. п.)

# Страницы

- **Страницы.** Это специальные объекты баз данных, реализованные в последней версии СУБД Microsoft Access (Access 2000). Правда, более корректно их называть *страниками доступа к данным*. Физически это особый объект, выполненный в коде *HTML*, размещаемый на Web-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является базой данных, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере. Пользуясь этими компонентами, посетитель Web-узла может просматривать записи базы в полях страницы доступа. Таким образом, страницы доступа к данным осуществляют интерфейс между клиентом, сервером и базой данных, размещенной на сервере. Эта база данных не обязательно должна быть базой данных Microsoft Access. Страницы доступа, созданные средствами Microsoft Access, позволяют работать также с базами данных Microsoft SQL Server

# Макросы и модули

- **Макросы и модули.** Эти категории объектов предназначены как для автоматизации повторяющихся операций при работе с системой управления базами данных, так и для создания новых функций путем программирования. В СУБД Microsoft Access *макросы* состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой. *Модули* создаются средствами внешнего языка программирования, в данном случае языка Visual Basic for Applications. Это одно из средств, с помощью которых разработчик базы может заложить в нее нестандартные функциональные возможности, удовлетворить специфические требования заказчика, повысить быстродействие системы управления, а также уровень ее защищенности



# Формирование таблицы

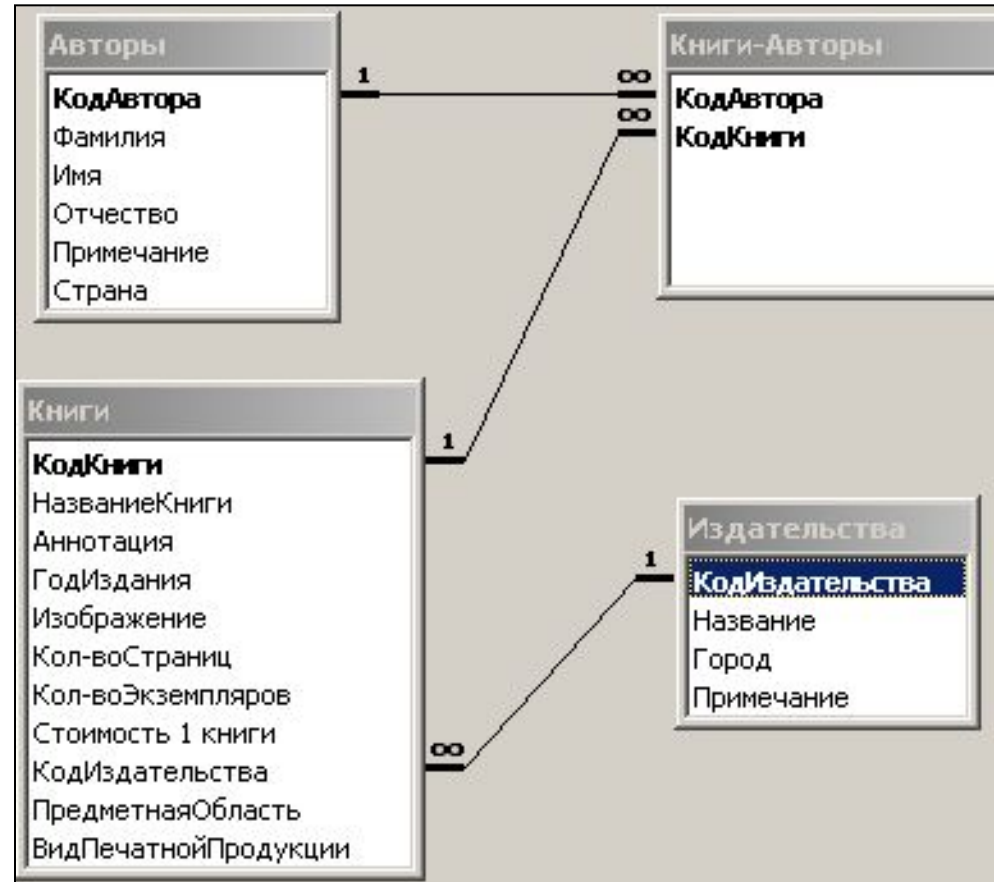
- 
- БД в Access – единый большой объект, который объединяет такие основные составляющие как таблицы, отчеты, запросы, формы и позволяет их хранить в едином дисковом формате.
- Основным структурным компонентом БД является таблица, содержащая записи определенного вида и формы.
- **Таблицы**
- MS Access поддерживает реляционную модель данных, т.е. база данных состоит из множества взаимосвязанных отношений, каждому отношению соответствует отдельная таблица.
- Таблицы состоят из строк и столбцов. Столбцы таблицы отражают общие свойства или характеристики объектов реального мира, которые моделируются с использованием данной таблицы. Для каждого свойства пользователь выделяет один столбец. Строки таблицы соответствуют различным экземплярам реальных объектов и называются записями.
- Желательно для каждой таблицы определить первичный ключ – столбец или набор столбцов однозначно определяющих строку таблицы.
- Первым этапом при создании таблицы является определение перечня полей, из которых она должна состоять, их типов и размеров. При этом каждому полю присваивается уникальное имя. Указываемый тип данных показывает СУБД, каким образом нужно обрабатывать поле.
-

# Основные типы данных в СУБД Access

Тип данных	Содержимое поля
Текстовый	(Значение по умолчанию). Текст или числа, не требующие проведения расчетов. До 255 символов
Поле МЕМО	Длинный текст или комбинация текста и чисел. До 65535 символов
Числовой	Числовые данные, используемые для проведения расчетов.
Дата/время	Даты и время, относящиеся к годам с 100 по 9999, включительно.
Денежный	Денежные значения и числовые данные, используемые в математических расчетах.
Счетчик	Уникальные последовательно возрастающие (на 1), автоматически вводящиеся при добавлении каждой новой записи в таблицу.
Логический	Логические значения, а также поля, которые могут содержать одно из двух возможных значений (True/False, Да/Нет).
Поле объекта OLE	Объект (например, электронная таблица Microsoft Excel, документ Microsoft Word, рисунок, звукозапись или другие данные в двоичном формате)

# Связи между таблицами

Для создания связей между таблицами СУБД Access имеет специальное диалоговое окно, которое называется **Схема данных**.



Связь между полями устанавливают путем **перетаскивания** имени поля из одной таблицы в другую на соответствующее ему поле - внешний ключ.

# Связи между таблицами

После перетаскивания открывается диалоговое окно Связи, в котором можно задавать свойства образующейся связи, такие как целостность и каскадное обновление полей.

Связи

Таблица/запрос:	Связанная таблица/запрос:
книги	книги-авторы
код книги	код книги

Обеспечение целостности данных

каскадное обновление связанных полей

каскадное удаление связанных записей

Тип отношения: один-ко-многим

OK

Отмена

Объединение...

# **Целостность БД** означает:

- что таблица должна иметь первичный ключ (уникальный идентификатор);
- целостность ссылок между таблицами: если установлена связь с первичным ключом другой таблицы, то в ней должно существовать соответствующее значение.

В Access для поддержки целостности можно использовать ***каскадное обновление и каскадное удаление***.

# Целостность данных

## Издательства

*Главная таблица*

КодИздательства	Название
1	Бином
2	ВНУ

*Подчиненная таблица*

## Книги

КодКниги	КодИздательства	НазваниеКниги
1	1	Азбука
2	3	Словарь

Если установить обеспечение целостности данных, получим сообщение об ошибке

# Типы запросов

- Access включает множество типов запросов, которые можно разбить на пять основных категорий

1. *Запрос на выборку.*
2. *Запрос с параметрами.*
3. *Перекрестный запрос.*
4. *Запрос на изменения.*

Существует четыре типа запросов на изменение:

- на создание таблицы,
- на удаление записей,
- на добавление
- на обновление записей.

5. *SQL.* Это запрос, который создается с помощью инструкций языка SQL.

# Построение запросов в СУБД Access

1. Объект «Запрос».
2. Режимы работы с объектом «Запрос»
3. Типология объекта «Запрос»
  - a) *Запрос на выборку (условия отбора, вычисления в запросе)*
  - b) *Запрос с параметрами*
  - c) *Перекрестный запрос*
  - d) *Запросы на изменения*



# Выражения в запросах

N	Операторы	Примеры	Пояснение к примеру
1	Операторы сравнения (<, <=, >, >=, =, <>)	>8 <b>And</b> <20	выбор значений, больших 8, но меньших 20
2	Логические операторы <b>And, Or</b>	Иванов <b>Or</b> Петров	выбор записей, содержащих в полях заданные фамилии
3	Специальные операторы:		
4	<b>Between</b> (диапазон значений)	<b>Between</b> 40 <b>And</b> 100	выбор записей, содержащих в полях значения между >=40 и <=100
5	<b>In</b> (список значений)	<b>In</b> ("Физика", "Математика", "Информатика")	поиск записей по какому-либо критерию осуществляется только в записях, содержащих в поле Предмет указанные дисциплины
6	<b>Like</b> (образец поиска в полях)	<b>Like</b> "Иванов"	поиск по шаблону записей, содержащих в поле Фамилия "Иванов"

# Символы шаблона оператора Like

Символ	Назначение	Примеры	Пояснение к примеру
*	заменяет любую группу любых символов	<b>Like"Вас*" в поле Имя</b>	вывод всех записей, имя которых начинается с букв Вас, например, Василиса, Василий и т.д.
		<b>Like"*ова" в поле Фамилия</b>	вывод всех записей, фамилия которых заканчивается на буквы ова, например, Петрова, Иванова и т.д.
?	заменяет любой один символ	<b>Like"М?ша" в поле Имя</b>	вывод всех записей, имя которых начинается с буквы М, оканчивается буквами ша, например, Миша, Маша
#	заменяет одну любую цифру	<b>Like"1#5"</b>	Вывод всех записей, содержащих в указанном поле числа 105, 115, 125 и т.д.
[ ]	указывает допустимый диапазон символов	<b>Like "[П, Т, К]оля" в поле Имя</b>	Вывод записей, имеющих имя Поля, Толя, Коля

- После создания и запуска запроса Access может вернуть и отобразить в виде строк и столбцов множество записей. Эти записи называются *результатирующим набором записей*.

Запрос1 : запрос на выборку				
	Фамилия	Имя	Должность	Дата рождения
▶	Белова	Мария	Представитель	08/12/1968
	Новиков	Павел	Вице-президент	19/02/1952
	Бабкина	Ольга	Представитель	
	Воронова	Дарья	Представитель	
	Кротов	Андрей	Менеджер по г	

-  Фильтр по выделенному
- Исключить выделенное
- Фильтр для:
-  Удалить фильтр
-  Сортировка по возрастанию
-  Сортировка по убыванию

- С результирующим набором записей можно проводить операции сортировки и фильтрации

# Режимы работы с объектом «Запрос»

- Режим конструктора
- Режим таблицы
- Режим SQL

Продажи по сотрудникам и странам : запрос на выборку

Сотрудники

- \* КодСотрудн
- Фамилия
- Имя
- Должность

Заказы

- \* КодЗаказа
- КодКлиента
- КодСотрудн
- ДатаРазмеще

Промежуточная сумма заказа

- \* КодЗаказа
- ПромежуточнаяСумма

Поле:	Страна	Фамилия	Имя	ДатаИсполнения	КодЗаказа	СуммаПродаж: ПромежуточнаяСум
Имя таблицы:	Сотрудники	Сотрудни	Сотрудни	Заказы	Заказы	Промежуточная сумма заказа
Сортировка:		по возраст				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				Between [Начальная дата]		
или:						

# Конструирование запроса

БЛАНК ЗАПРОСА

Поле:	Страна	Фамилия
Имя таблицы:	Сотрудники	Сотрудни
Сортировка:		по возраст ▾
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
или:		

<i>Поле</i>	Имя поля
<i>Имя таблицы</i>	Имя таблицы
<i>Сортировка</i>	Место ввода инструкций сортировки
<i>Вывод на экран</i>	Определяет, будет ли присутствовать поле в результирующем наборе данных
<i>Условие отбора</i>	Содержит первое условие, ограничивающее набор записей
<i>Или</i>	Другие условия ограничения набора записей

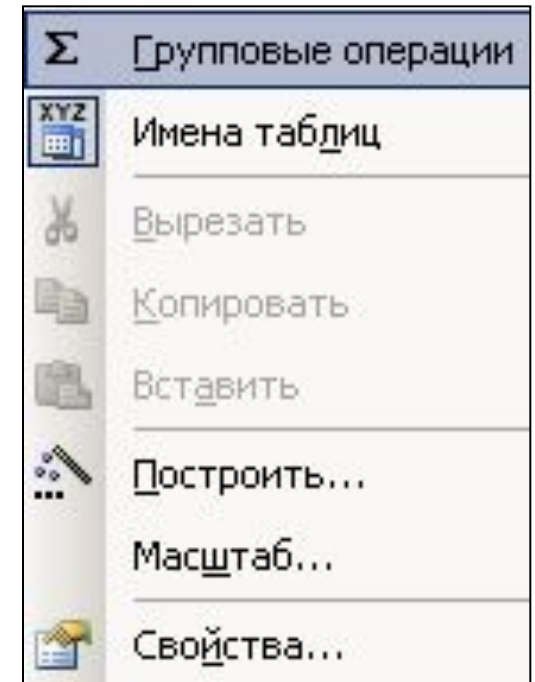
# Групповые операции в

сах

Контекстное меню

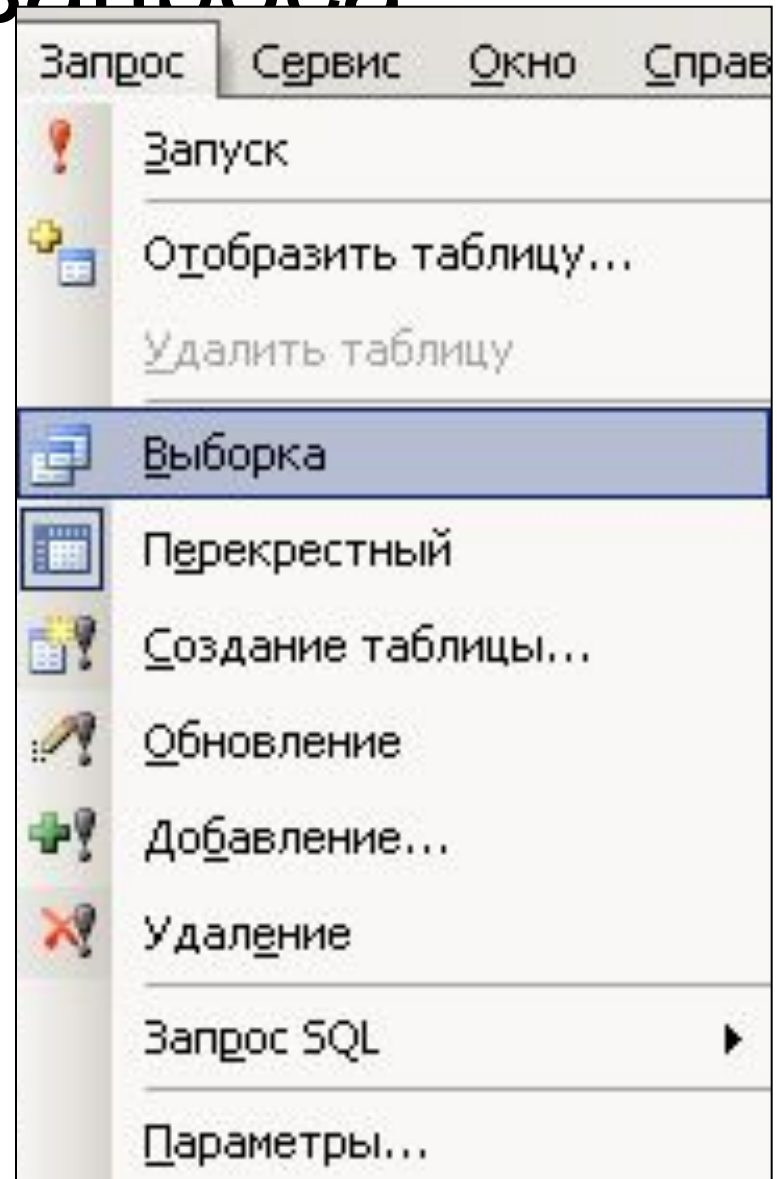
Поле:	Страна	Фамилия
Имя таблицы:	Сотрудники	Сотрудни
Сортировка:		по возраст
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
или:		

Поле:	Страна	Фамилия
Имя таблицы:	Сотрудники	Сотрудни
Групповая операция:	Группировк	Группирои
Сортировка:	Группировк	по возраст
Вывод на экран:	Sum	<input checked="" type="checkbox"/>
Условие отбора:	Avg	
или:	Min	
	Max	
	Count	
	StDev	
	Var	



# Выбор типа запроса

- При открытии запроса в режиме конструктора появляется пункт меню **Запрос**, который позволяет выбрать тип запроса



# Запрос на выборку



Извлекает данные из одной или нескольких таблиц и результаты отображает в в режиме таблицы.

- Используются для группировки записей, а также для вычисления сумм, средних значений, подсчета количества записей и расчета итоговых значений других типов



# Ввод условий отбора

- При вводе критериев отбора для создания запроса на выборку, используются операторы, которые

математические	отношения	логические	строковые	вспомогательные
* умножение / деление + сложение – вычитание	= равно ≠ не равно < меньше > больше	And Or Not	Like	between  is null

# Примеры условий отбора

Поле:	Фамилия	Имя	ДатаРождения
Имя таблицы:	Сотрудники	Сотрудники	Сотрудники
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			>01/01/1960
или:			

Поле:	Фамилия	Имя
Имя таблицы:	Сотрудники	Сотрудники
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Like "[А-К]*"
или:		

# Условия отбора

- **Оператор *Like***

Позволяет задать образец поиска, используя символы шаблона

Символ шаблона	Описание	Пример
?	Заменяет один произвольный символ	Like"?.2/98" Like"?аша"
*	Заменяет любое число произвольных символов	<b>Like"*.*.98" Like" Иван*"</b>
#	Заменяет одну любую цифру	Like"31.3.9#" Like"Мира 1##"
[ ]	Позволяет указать допустимый диапазон символов	<b>Like"[А-Л]*"</b>

# Условия отбора

- Оператор **Between**

Позволяет задать границы поиска значений

**Between** Значение1 **And** Значение2

Поле:	Фамилия	ДатаРождения
Имя таблицы:	Сотрудны	Сотрудники
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Between #04/03/1955# And #04/03/1960#

# Вычисления в запросе

- **Встроенные** («итоговые») для расчета значений по группам записей: сумма, среднее, число значений.

Запрос1 : запрос на выборку

Count-КодСотруд	Имя
1	Андрей
3	Анна
1	Дарья
2	Иван

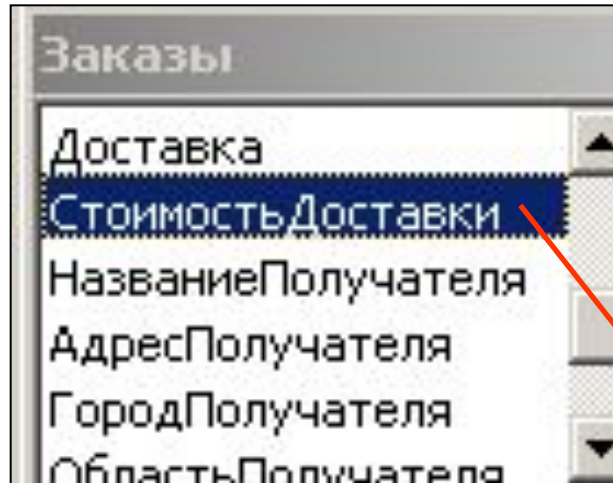
Поле:	КодСотрудника	Имя
Имя таблицы:	Сотрудники	Сотрудники
Групповая операция:	Count	Группировка
Сортировка:		по возрастанию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
или:		

# Вычисления в запросе

- **Пользовательские** вычисления для выполнения расчетов с числовыми, строковыми или значениями дат для каждой записи с использованием данных из одного или нескольких полей.
- Для ввода таких выражений необходимо создать новое **вычисляемое поле** в бланке запроса:

**Имя\_вычисляемого\_поля: Выражение**

# Вычисляемые поля в запросе



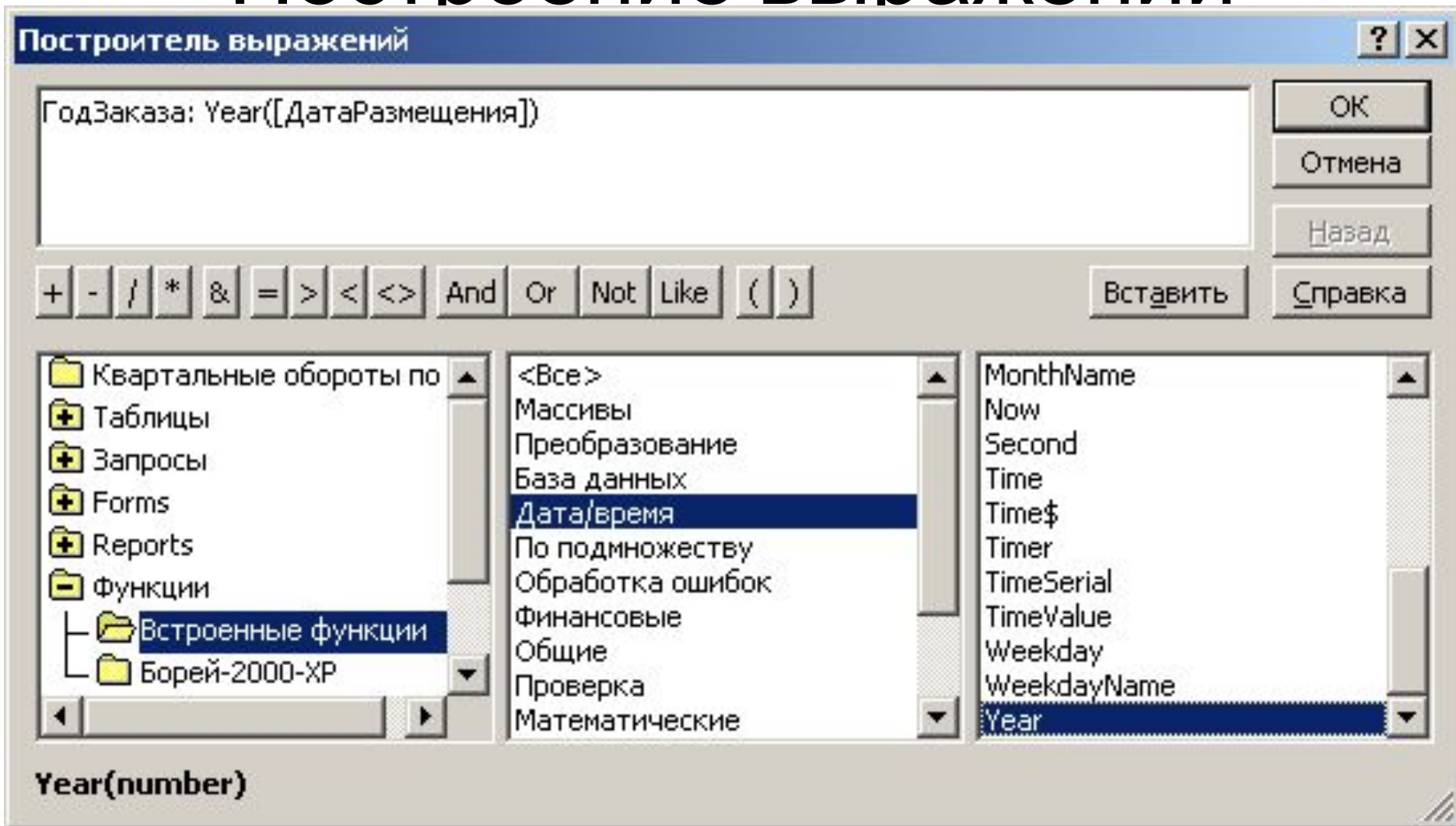
Поле:	КодЗаказа	СтоимостьДоставкиСкидка: [СтоимостьДоставки]-10
Имя таблицы:	Заказы	Заказы
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		

# Вычисляемые поля в запросе

- Для построения вычисляемого поля можно использовать **построитель выражений**
- **Построитель выражений** позволяет формировать выражение с использованием встроенных функций, имен полей таблиц, знаков математических операций.



# Построение выражений



# Примеры выражений

- $AVG([Цена])$  - среднее от значений поля «Цена»
- $Sum([Стоимость]*[Количество])$  – сумма произведений значений полей «Стоимость» и «Количество»

# Запрос с параметром

- Создается как запрос на выборку, далее в бланке запроса в строке «Условие отбора» указывается запрашиваемый параметр в [ ]

Поле:	Фамилия	Имя	Должность
Имя таблицы:	Сотрудники	Сотрудники	Сотрудники
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:			[Укажите должность]

Введите значение параметра

Укажите должность

Представитель

OK Отмена

# Перекрестный запрос

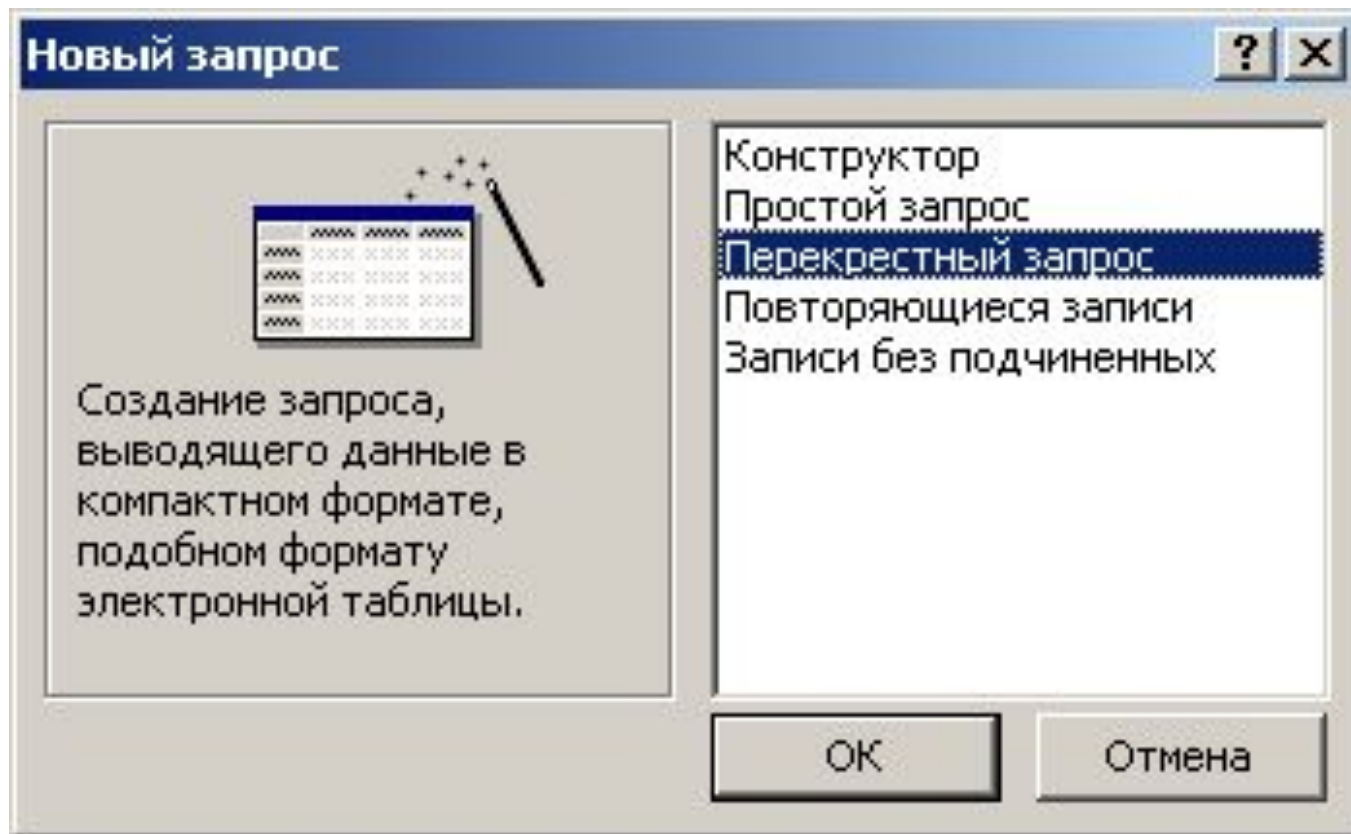


- Отображает результаты статистических расчетов, выполненных по данным одного поля. Например вычисление количества значений поля КодЗаказа по каждой фирме-клиенту по сотруднику

	Сотрудник	Итоговое значение	ALFKI	ANATR	ANTON	AROUT
▶	Белова, Мария	122	1			3
	Новиков, Павел	96				
	Бабкина, Анна	126	1	2	3	2
	Воронова, Дарья	156	1	1	1	5
	Кротов, Андрей	42				
	Акбаев, Иван	67				1
	Кралев, Иван	73		1	3	
	Крылова, Анна	105				
	Ясенева, Анна	43				2

# Как создать перекрестный запрос?

- Новый перекрестный запрос создается с помощью Мастера



# Создание перекрестного запроса с

## помощью Мастера

### Создание перекрестных таблиц

Какие вычисления необходимо провести для каждой ячейки на пересечении строк и столбцов?

Например, можно вычислить сумму заказов для каждого сотрудника (столбец) по странам и регионам (строка).

Вычислить итоговое значение для каждой строки?

Да.

Поля:

КодЗаказа  
ДатаРазмещения  
ДатаНазначения  
ДатаИсполнения  
Доставка  
СтоимостьДоставки  
НазваниеПолучателя  
АдресПолучателя  
ГородПолучателя  
ОбластьПолучателя  
ИндексПолучателя

Функции:

Дисперсия  
Максимум  
Минимум  
Отклонение  
Первый  
Последний  
Среднее  
Сумма  
Число

Образец:

КодСотрудн	КодКлиента	КодКлиента	КодКлиента
КодСотрудник	Число(КодЗаказа)		
КодСотрудник			
КодСотрудник			
КодСотрудник			

Отмена

< Назад

Далее >

Готово

# Запросы на изменения

- Запрос на удаление данных



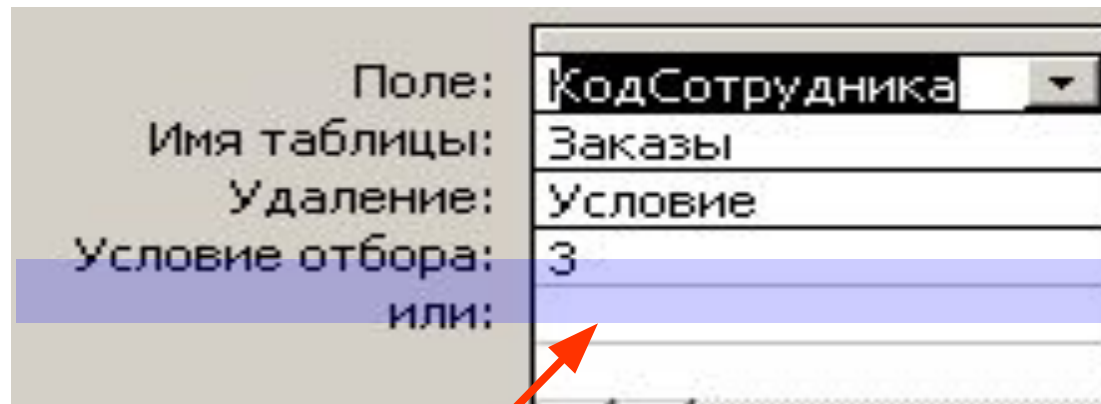
Удаление заказов

- Запрос на обновление данных



- **Запрос на удаление** позволяет удалить записи удовлетворяющие условию.
- Выбор типа запроса - меню Запрос – Удаление.

Поле:	КодСотрудника
Имя таблицы:	Заказы
Удаление:	Условие
Условие отбора:	3
или:	



- В строке «Условие отбора» указывается условие



- **Запрос на обновление** позволяет изменить значение определенного поля в записи удовлетворяющей условию.
- Выбор типа запроса - меню Запрос – Удаление

Поле:	КодКлиента	
Имя таблицы:	Клиенты	
Обновление:	"Q-1"	
Условие отбора:	"QUEEN"	
или:		

- В строке «Обновление» указывается новое значение поля

# Обновление данных с использованием вычисляемых полей

Для обновления данных в таблице с использованием запроса используются вычисляемые поля.

Например, выражение:

**[Цена]\*1,05**

в строке «Обновление» позволяет увеличить все значения в поле «Цена» на 5 проце

Поле:	Цена
Имя таблицы:	
Обновление:	[Цена]*1,05

# Виды запросов

- На выборку
- На изменение
- Простой анализ данных

# Запросы на выборку

Наиболее часто используется запрос на выборку.

При его выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или нескольких таблиц и выводятся в определенном порядке.

The screenshot displays a database query builder interface. At the top, three tables are shown: 'Сотрудники' (Employees), 'Заказы' (Orders), and 'Промежуточн' (Intermediate). 'Сотрудники' has fields: КодСотрудни, Фамилия, Имя, Должность. 'Заказы' has fields: КодЗаказа, КодКлиента, КодСотрудник, ДатаРазмеще. 'Промежуточн' has fields: КодЗаказа, Промежуточн. Relationships are shown between 'Сотрудники' and 'Заказы' (1 to ∞) and between 'Заказы' and 'Промежуточн'. Below the tables is a query configuration table with the following content:

Поле:	Фамилия	ДатаРазмещения	Промежуточная
Имя таблицы:	Сотрудники	Заказы	Промежуточная
Групповая операция:	Группировка	Условие	Sum
Сортировка:			по убыванию
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Between #6/1 /96# And #6/1 5/96#		

Annotations on the right side of the interface include:

- Добавление таблиц и запросов (Add tables and queries)
- Добавление и удаление полей (Add and remove fields)
- Выбор способа вычисления (Choose calculation method)
- Задание порядка сортировки (Specify sorting order)
- Задание условия отбора (Specify selection condition)

# Постоянные условия отбора

Поле:	Общие данные	диск
Имя таблицы:	клиенты	прокат
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Like "слово"	>"1"
или:		

# Создание запроса с параметрами с приглашением на ввод условий отбора

Запрос с параметрами отображает одно или несколько определенных диалоговых окон, выводящих приглашение пользователю ввести условия отбора. Можно создать специальное диалоговое окно для ввода параметров запроса.

# Порядок действий

- 1 Создайте запрос на выборку или перекрестный запрос.
- 2 В режиме конструктора запроса переместите с помощью мыши поля из списка полей в бланк запроса.
- 3 Для каждого поля, которое предполагается использовать как параметр, введите в ячейку строки **Условие отбора** текст приглашения, заклученный в квадратные скобки (будет выводиться при запуске запроса). Текст подсказки должен отличаться от имени поля, но может включать его.



Для поля, в котором отображаются даты, можно вывести приглашения "Введите начальную дату:" и "Введите конечную дату:" для определения диапазона отбираемых значений. Для этого введите в ячейку строки Условие отбора выражение **Between** [Введите начальную дату:] **And** [Введите конечную дату:].

- 4 Для просмотра результатов нажмите кнопку **Вид** на панели инструментов и введите значение параметра.

Поле: **Общие данные**

Имя таблицы: клиенты

Групповая операция: Группировка

Сортировка:

Вывод на экран:

Условие отбора: [текст запроса]

или:

DB1 : база данных

Таблицы    Запросы    Формы

диски

**запрос\_3**

прокат дисков

### Введите значение параметра

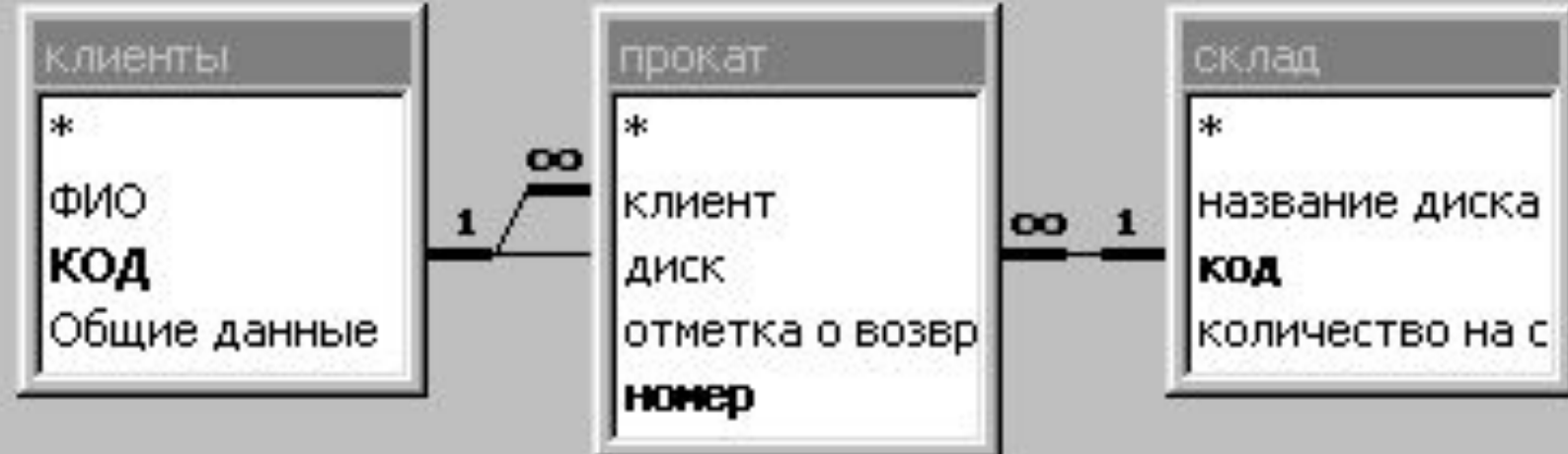
текст запроса

ОК    Отмена

# Вычисления в запросе

- Существует ряд вычислений, которые можно выполнить в запросе, например, найти сумму или среднее по значениям одного поля, перемножить значения двух полей или вычислить дату, отстоящую на три месяца от текущей даты.
- Результаты вычислений, выводящиеся в поле, не запоминаются в базовой таблице. Вычисления производятся всякий раз, когда выполняется запрос, поэтому результаты всегда представляют текущее содержимое базы данных.

- Встроенные функции Access позволяют найти следующие значения для групп записей или для всех записей: сумму, среднее, число значений в поле, минимальное значение, максимальное значение, среднеквадратичное отклонение или дисперсию.
- Для каждого вычисляемого поля следует выбрать одно выражение.
- Выражения, определяющие вычисляемые поля, создаются с помощью мастера простых запросов или вводятся пользователем в строку **Групповая операция** бланка запроса, в которой допускается выбор статистических функций для выполняемых над полем вычислений.



Поле:  
Имя таблицы:  
Групповая операция:  
Сортировка:  
Вывод на экран:  
Условие отбора:  
или:

название диска	количество на скл	
склад	склад	
Группировка	Sum	
Sum		
Avg	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Min		
Max		
Count		
StDev		
Var		
First		



# Свойства запроса



## Общие

Описание . . . . .	
Вывод всех полей . . . . .	Нет
Набор значений . . . . .	Все
Уникальные значения . . . . .	Да
Уникальные записи . . . . .	Нет
При запуске предоставляются права .	Пользова
База данных-источник . . . . .	(текущая
Строка подключения-источник . . . . .	
Блокировка записей . . . . .	Отсутсте
Тип набора записей . . . . .	Динамиче
Время ожидания ODBC . . . . .	60
Фильтр . . . . .	
Порядок сортировки . . . . .	



озвр

кол  
скл  
Sup

# Выражения, определяемые пользователем

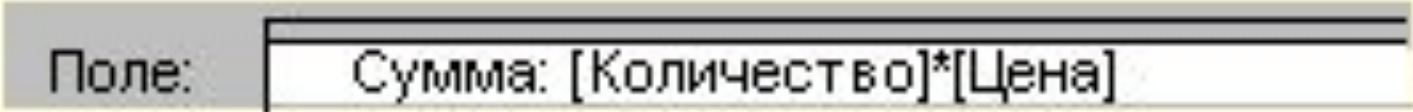
- дают возможность выполнять действия с числами, датами и текстовыми значениями в каждой записи с использованием данных из одного или нескольких полей.

**Пример:** выражение позволяет умножить значения одного поля на итоговое значение, найти разность значений двух полей типа даты, соединять несколько строковых значений в текстовом поле или создавать подчиненные запросы.

Заполняя ячейки в строке Групповая операция в бланке запроса, можно выполнить расчеты для групп записей и вычислить сумму, среднее, количество или другой тип итогового значения для вычисляемого поля.

- Для расчетов с использованием формул, определяемых пользователем, требуется создать новое вычисляемое поле прямо в бланке запроса. Вычисляемое поле создается с помощью выражения, которое вводится в пустую ячейку Поле в бланке запроса. Подробно изложите материал
- Приведите примеры и иллюстрации
- Почему это должно взволновать аудиторию

Для расчетов с использованием формул, определяемых пользователем, требуется создать новое вычисляемое поле прямо в бланке запроса. Вычисляемое поле создается с помощью выражения, которое вводится в пустую ячейку **Поле** в бланке



Поле:

При выполнении расчетов допускаются сложные выражения, например, **Sum([НаСкладе]+[Заказано])**. Имеется также возможность определить для вычисляемого поля условие отбора, позволяющее отбирать используемые в расчетах данные.



# Примеры управления текстом в ВЫЧИСЛЯЕМЫХ ПОЛЯХ

Выражение	Описание
ПолноеИмя: [Имя] & " " & [Фамилия]	Отображает значения полей «Имя» и «Фамилия», разделенные пробелом в поле «Полное имя».
Адрес2: [Город] & " " & [Район] & " " & [Индекс]	Отображает значения полей «Город», «Район» и «Индекс», разделенные пробелами в поле «Адрес2».
ПерваяБукваТовара: Left([НазваниеТовара], 1)	Использует функцию <b>Left</b> для отображения в поле «ПерваяБукваТовара» первого символа значения поля «НазваниеТовара».
КодТипа: Right([КодПродукта], 2)	Использует функцию <b>Right</b> для отображения в поле «КодТипа» последних двух символов значения поля «КодПродукта».
КодРайона: Mid([Телефон], 2, 3)	Использует функцию <b>Mid</b> для отображения в поле «КодРайона» трех символов, начиная со второго, значения поля «Телефон».

# Примеры выполнения арифметических операций в вычисляемых полях

Выражение	Описание
ПервоначальныйФрахт: [Фрахт] * 1.1	Отображает в поле «ПервоначальныйФрахт» стоимость фрахта плюс 10 процентов.
ОбъемЗаказа: [Количество] * [Цена]	Отображает в поле «ОбъемЗаказа» произведение значений полей «Количество» и «Цена».
ВремяДоставки: [ДатаЗаказа] - [ДатаОтгрузки]	Отображает в поле «ВремяДоставки» разницу между значениями полей «ДатаЗаказа» и «ДатаОтгрузки».
ВсегоВНаличии: [ТоварыНаСкладе]+[ЗаказанныеТовары]	Отображает в поле «ВсегоВНаличии» сумму значений полей «ТоварыНаСкладе» и «ЗаказанныеТовары».
ДоляФрахта: Sum([Фрахт])/Sum([ОбщийИтог]) *100	Отображает в поле «ДоляФрахта» процент на оплату фрахта для каждого итога. Для этого сумма всех значений поля «Фрахт» делится на сумму всех значений поля «ОбщийИтог». (В данном примере использована функция <b>Sum</b> .)

# Примеры вычисления и управления датами в вычисляемых полях

## Выражение

## Описание

ВремяДоставки: DateDiff("d",  
[ДатаЗаказа], [ДатаОтгрузки])

Использует функцию **DateDiff** для отображения в поле «ВремяДоставки» числа дней между датой заказа и датой отгрузки.

ГодНайма:  
DatePart("yyyy",[ДатаНайма])

Использует функцию **DatePart** для отображения в поле «ГодНайма» года найма сотрудника.

Date()- 30

Использует функцию **Date** для отображения даты, предшествующей текущей на 30 дней.

# Запросы на изменение

- **Запрос на добавление** - добавление выбранных записей из одной таблицы в другую
- **Запрос на обновление** - для быстрого изменения группы выбранных записей
- **Создание таблицы** - для сохранения выбранных записей в новой таблице
- **Запрос на удаление** - для удаления выбранных по условию записей

# Запрос на добавление

Открыть уже готовый запрос на выборку в режиме конструктора, далее открыть меню «Запрос» - создание таблицы

Создание таблицы

Создание таблицы:

имя таблицы:

в текущей базе данных

в другой базе данных

имя файла:

Общие данные | диск | Выражение1: [прои]

Общие данные	диск	Выражение1: [прои]
клиенты	прокат	
Группировка	Группировка	Выражение

# Запрос на обновление

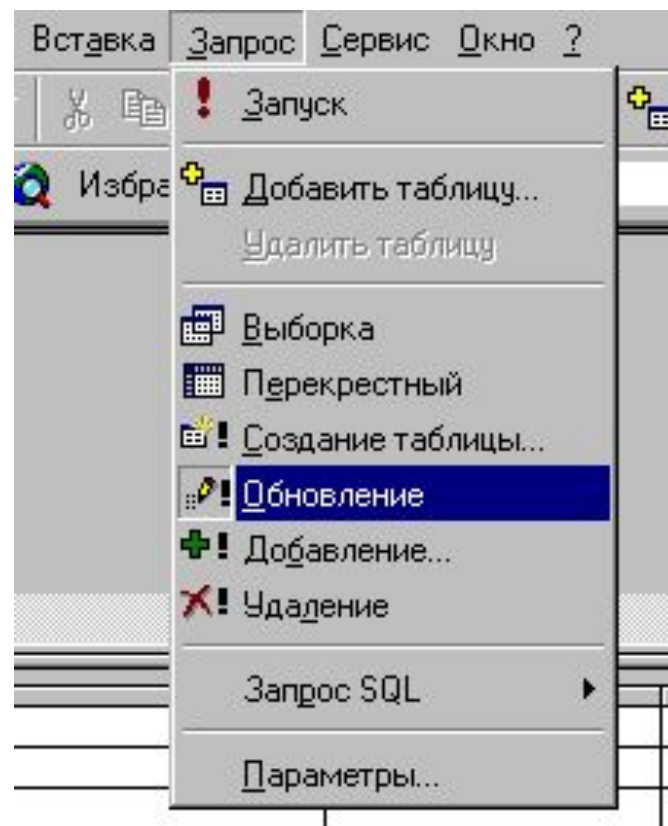
Открыть уже готовый запрос на выборку в режиме конструктора, далее открыть меню «Запрос» - обновление. Появится строка «обновление», в которое нужно ввести выражение или значения, которыми нужно заменить значения в выбранных полях



Поле:	Общие данные	ДИСК
Имя таблицы:	клиенты	прокат
Обновление:		
Условие отбора:	Like "слово"	>"1"
или:		

Вычисляемые поля используются также для обновления данных в запросе на обновление. Например, следующее выражение в ячейке **Обновление** позволяет увеличить все значения в поле «Цена» на 5 процентов.

Поле:	Цена
Имя таблицы:	Товары
Обновление:	[Цена]*1.05





# Запрос на удаление

Открыть уже готовый запрос на выборку в режиме конструктора, далее открыть меню «Запрос» - удаление. Появится строка «удаление», ниже которой вводится условие на удаление данных из этого поля

# запрос\_3 : запрос на удаление

клиенты\_1

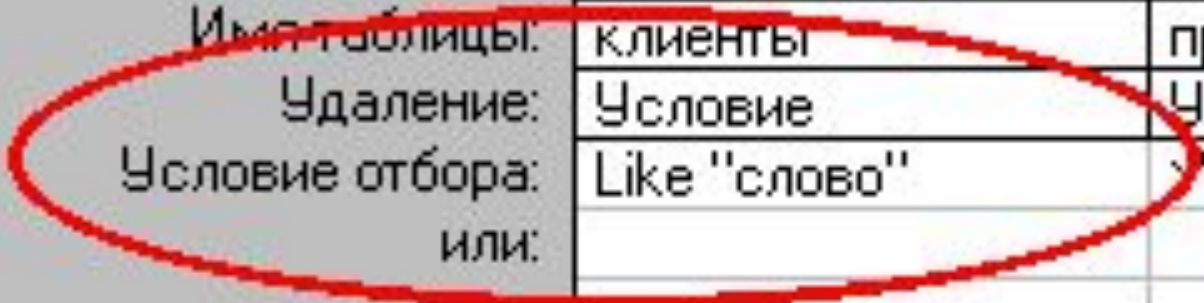
\*

ФИО

**КОД**


Общие данные

Поле:	Общие данные	диск
Имя таблицы:	клиенты	прокат
Удаление:	Условие	Условие
Условие отбора:	Like "слово"	'1'
или:		



# Создание новой таблицы

## Создание новой таблицы в запросе на создание таблицы

1. Создайте запрос, выбрав таблицы или запросы, содержащие записи, которые следует поместить в новую таблицу.  
☞ Инструкции
2. В **режиме конструктора** запроса нажмите стрелку рядом с **кнопкой Тип запроса** на панели инструментов и выберите команду **Создание таблицы**. На экране появится диалоговое окно **Создание таблицы**.
3. В поле **Имя таблицы** введите имя таблицы, которую необходимо создать или заменить.
4. Выберите параметр **В текущей базе данных**, чтобы поместить таблицу в открытую в данный момент базу данных или выберите параметр **В другой базе данных** и введите имя базы данных, в которую требуется поместить новую таблицу. В случае необходимости укажите путь.
5. Нажмите кнопку **ОК**.
6. Переместите поля, которые должна содержать новая таблица, из **списка полей** в **бланк запроса**.
7. Для полей, перемещенных в бланк запроса, в ячейке **Условие отбора** введите условие.  
Для получения дополнительных сведений об отделеении условий нажмите кнопку ☞.
8. Для просмотра новой таблицы до ее создания на панели инструментов нажмите **кнопку Вид**. Чтобы вернуться в режим конструктора запроса и внести необходимые изменения или запустить запрос, на панели инструментов нажмите **кнопку Вид**.
9. Для создания новой таблицы нажмите кнопку **Запуск**  на панели инструментов.

## Примечания

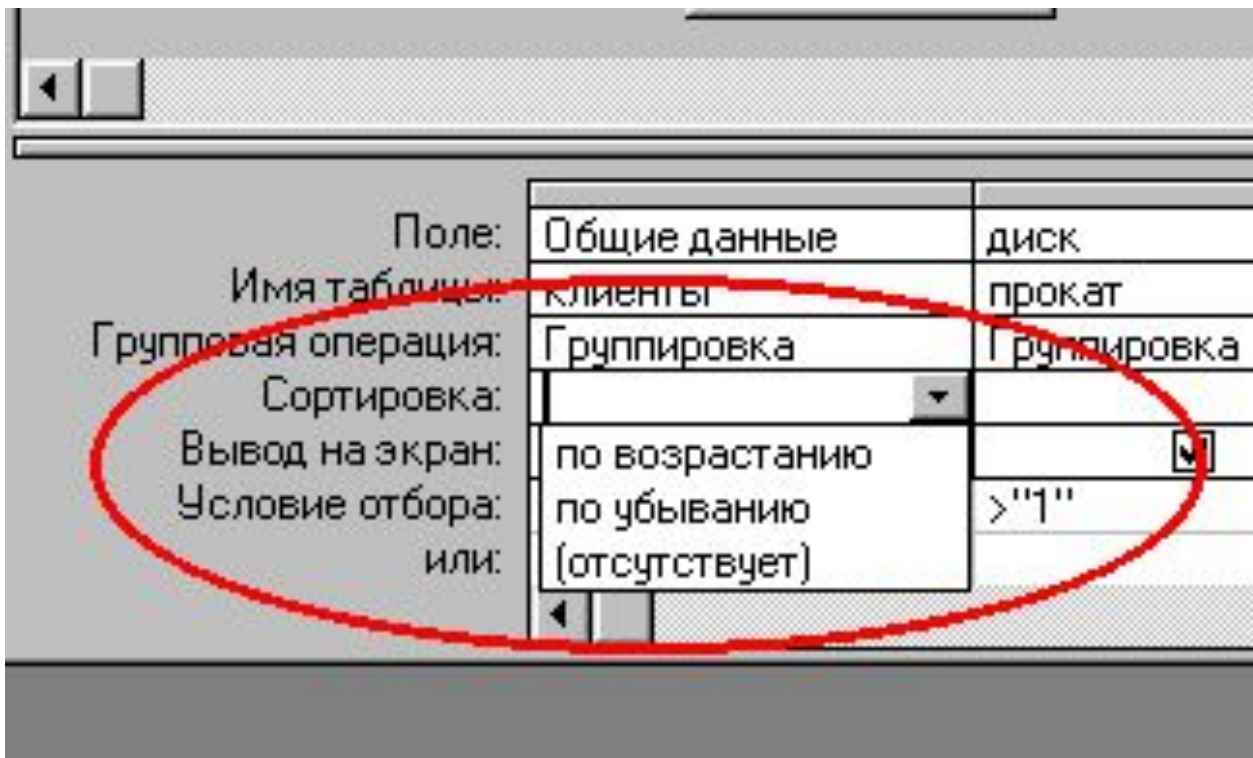
- Чтобы остановить выполнение запроса, нажмите клавиши CTRL+BREAK.
- Установленные в исходной таблице свойства полей и **ключевые поля** не переходят к данным в новой таблице.

# Простой анализ данных

- Простая сортировка
- Поиск повторяющихся записей - позволяет расположить рядом записи, заданное поле которых имеет одинаковое значение
- Поиск записей без подчиненных - позволяет найти записи в материнской таблице, для которых нет подчиненных записей в дочерней таблице

# Простая сортировка

Открыть запрос в режиме конструктора, заполнить атрибут поля «сортировка»



# Поиск повторяющихся записей

Создать запрос в режиме «Повторяющиеся записи»

**Поиск повторяющихся записей**

Выберите таблицу или запрос, в которых требуется найти записи с повторяющимися значениями.

Например, чтобы найти город, в котором имеется более одного клиента, выберите таблицу "Клиенты".

1	xxx	xxx	xxx
2	xxx	xxx	xxx
3	xxx	xxx	xxx
2	xxx	xxx	xxx
4	xxx	xxx	xxx

2	xxx	xxx	xxx
2	xxx	xxx	xxx

клиенты  
прокат  
склад

Показать:  
 таблицы  запросы  таблицы и запросы

Отмена < Назад Далее > Готово

**Новый запрос**

Самостоятельное создание нового запроса.

Конструктор  
Простой запрос  
Перекрестный запрос  
**Повторяющиеся записи**  
Записи без подчиненных

OK Отмена

# Поиск записей без

Создать запрос в режиме «Поиск записей без подчиненных»

**Поиск записей, не имеющих подчиненных**

Запрос выбирает записи из указываемой ниже таблицы, для которых не существует связанных записей в таблице, выбираемой на следующем экране. Например, такой запрос позволяет найти клиентов, не имеющих заказов.

Задайте таблицу или запрос, содержащий требуемые записи:

клиенты	
прокат	
склад	

Показать:  
 таблицы  запросы  таблицы и запросы

Отмена < Назад > Далее > Готово

**Новый запрос**

Создание запроса на поиск записей, которым не соответствует ни одна запись в подчиненной таблице.

Конструктор  
Простой запрос  
Перекрестный запрос  
Повторяющиеся записи  
**Записи без подчиненных**

OK Отмена

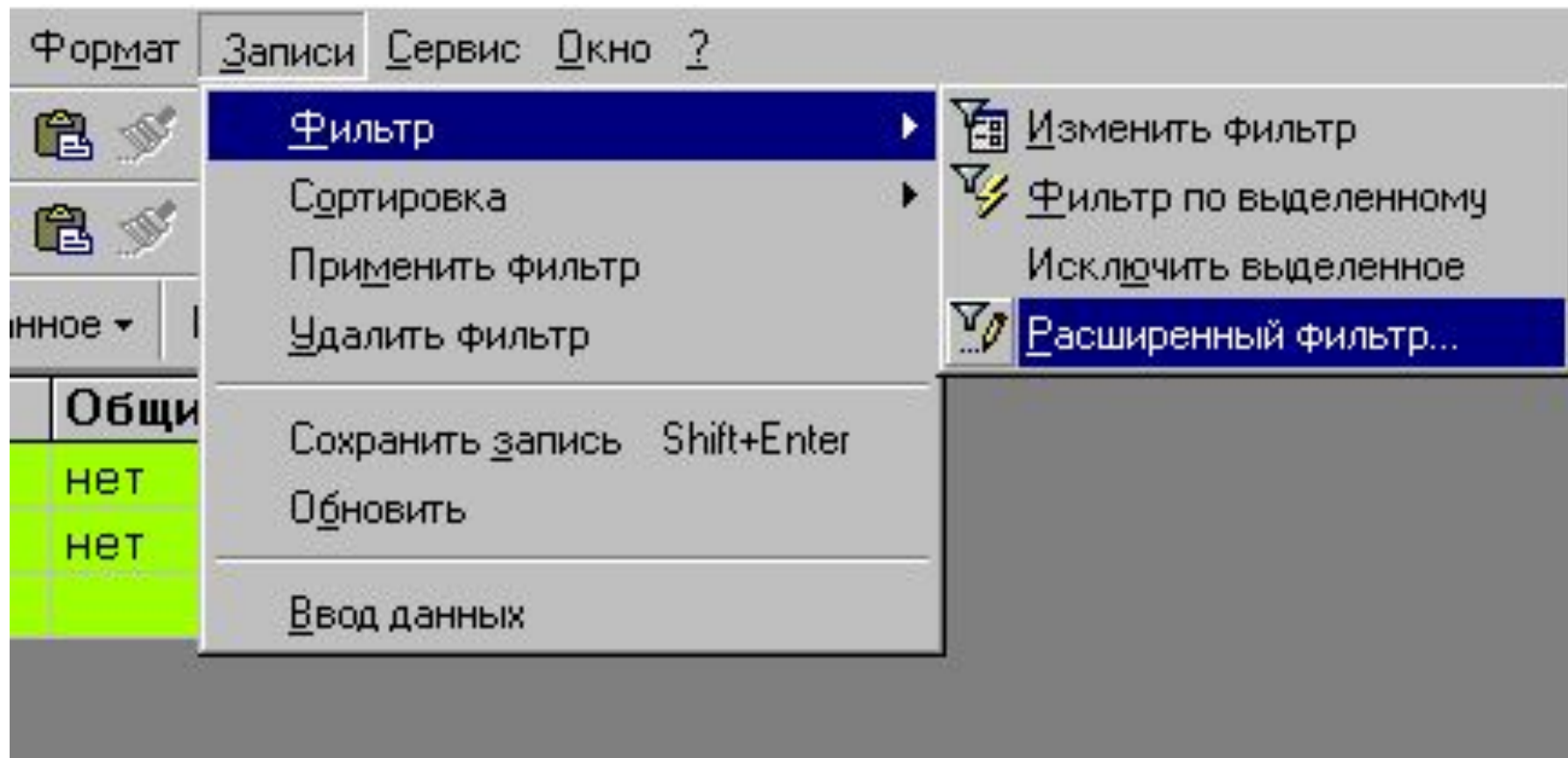
# Фильтры



# Назначение

Фильтр используют при работе в режиме формы или в режиме таблицы для просмотра или изменения подмножества записей. Запрос следует использовать для выполнения следующих действий:

- Просмотр подмножества записей без предварительного открытия конкретной таблицы или формы.
- Выбор таблиц, содержащих записи, с возможностью последующего добавления других таблиц.
- Отбор полей, выводящихся на экран при отображении результирующего набора записей.
- Выполнение вычислений над значениями полей.



# Защита БД

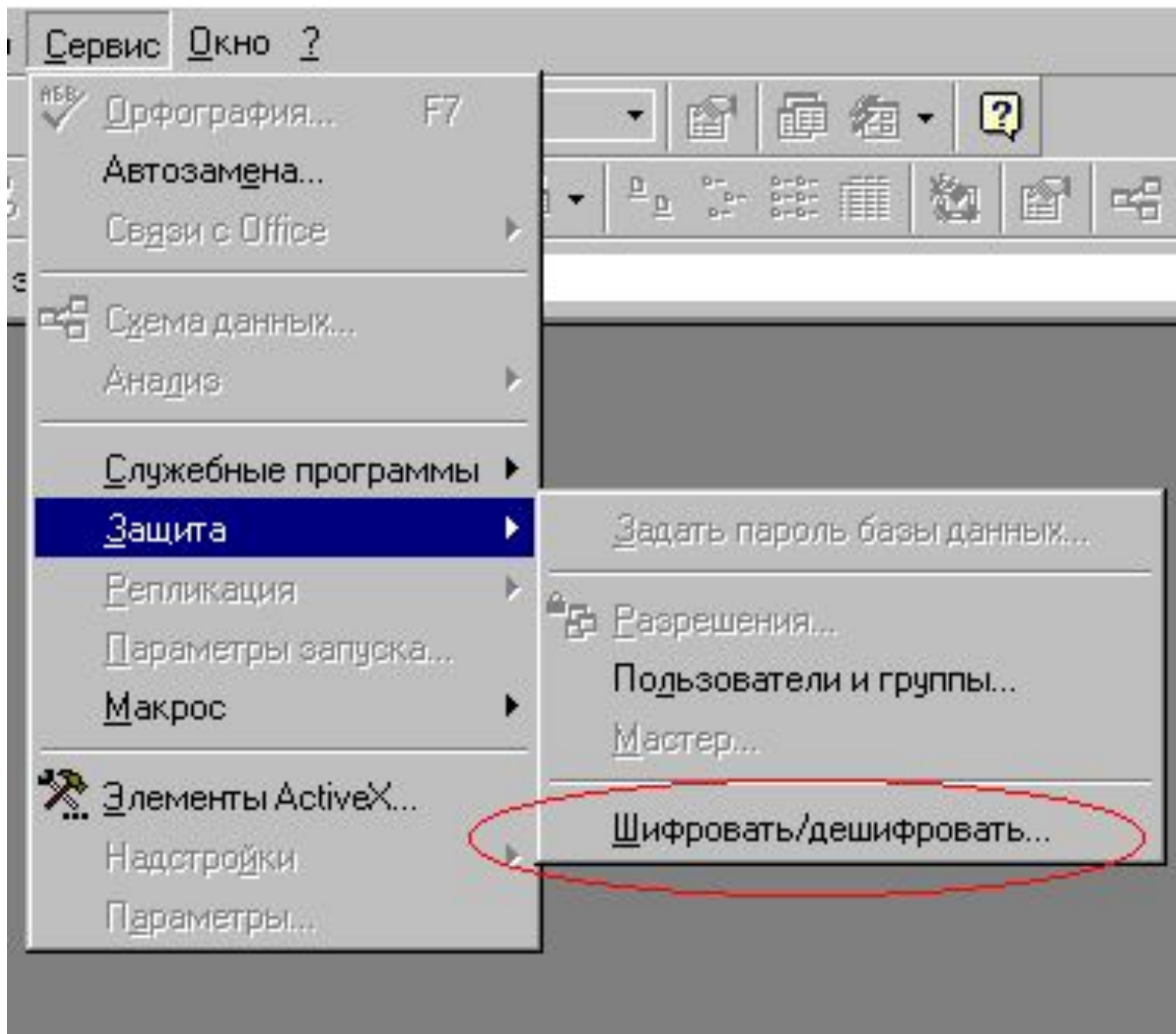
# Шифрование и дешифрование базы данных

- При шифровании базы данных ее файл сжимается и делается недоступным для чтения с помощью служебных программ или текстовых редакторов.
  - Дешифрование базы данных отменяет результаты операции шифрования.
- 1 Запустите Microsoft Access без открытия базы данных.

**Внимание!** Невозможно зашифровать или дешифровать открытую базу данных.

- 2 В меню **Сервис** выберите команду **Защита** и подкоманду **Шифровать/дешифровать**.
- 3 Укажите имя БД, которую требуется зашифровать или дешифровать, и нажмите кнопку ОК.
- 4 Укажите имя, диск и папку для конечной БД и нажмите кнопку ОК.

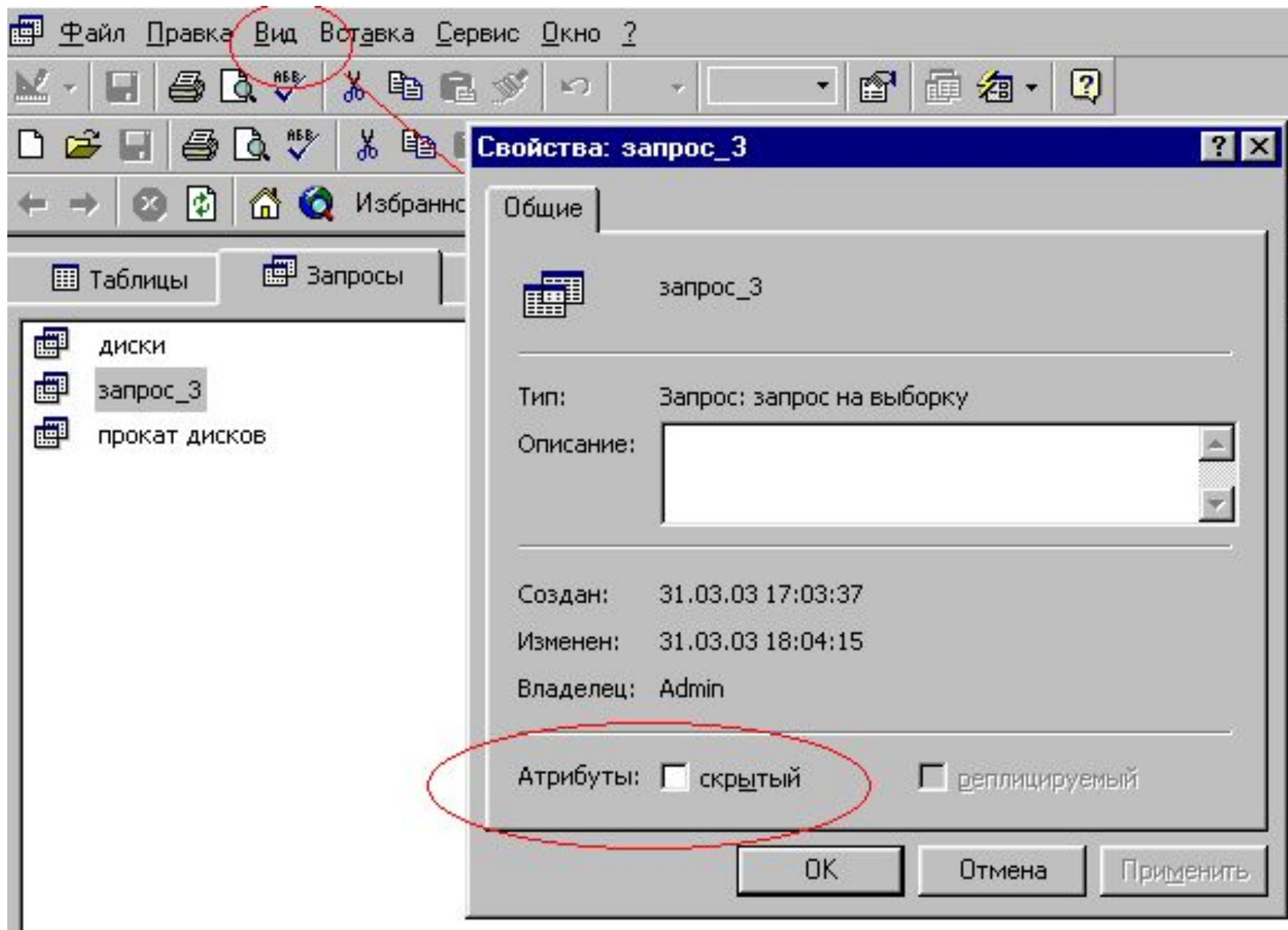
Можно указать новое имя или имя, совпадающее с исходным. Если указаны те же имя, диск и папка, а шифрование или дешифрование БД выполнено успешно, то исходный файл будет заменен на зашифрованный или дешифрованный. Если же операция приводит к ошибке, то исходный файл уладен не будет



# Скрытие объектов в окне базы данных

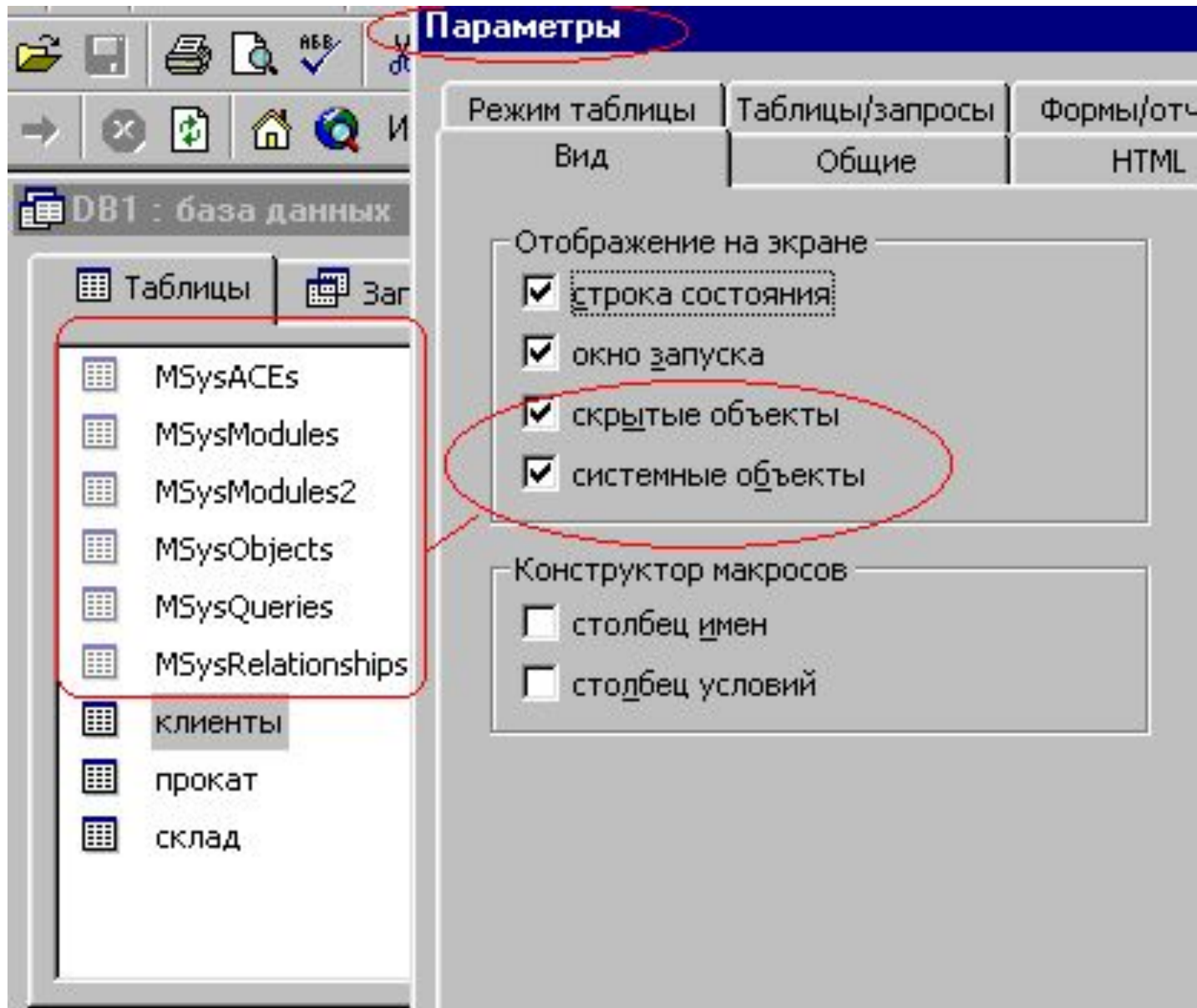
- 1 В окне базы данных выберите объект, который требуется сделать скрытым.
- 2 Нажмите кнопку **Свойства** на панели инструментов.
- 3 Установите флажок **Скрытый** и нажмите кнопку ОК.

**Примечание.** Для того чтобы показать все скрытые объекты, выберите в меню **Сервис** команду **Параметры**, выберите вкладку **Вид**, а затем установите флажок **Скрытые объекты**. Значки скрытых объектов будут отображены как неконтрастные.





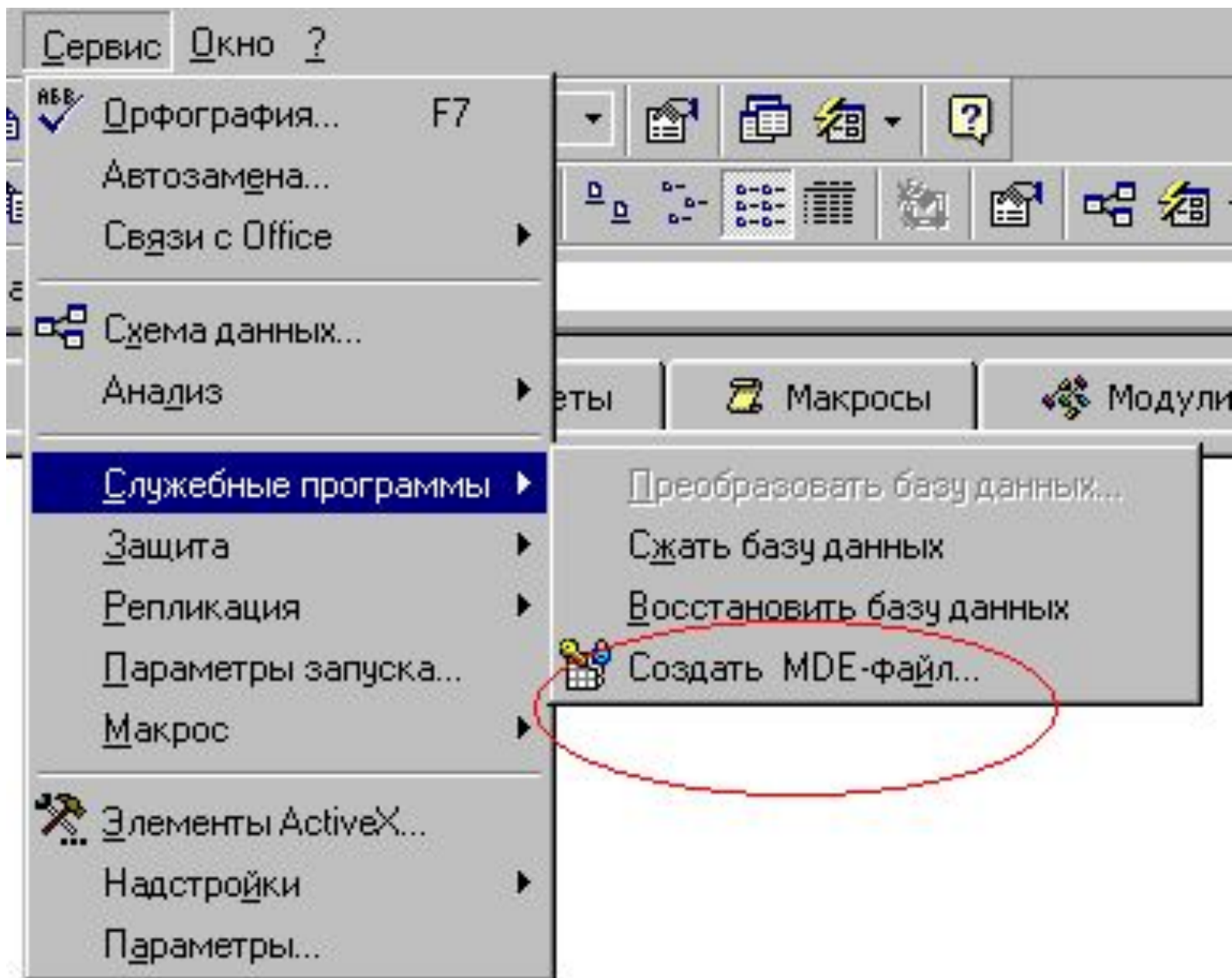
# Показ всех объектов



# Защита базы данных

Microsoft Access обеспечивает два способа защиты базы данных:

- **установка пароля**, требуемого при открытии базы данных,
- **защита на уровне пользователей**, которая позволяет ограничить, к какой части базы данных пользователь будет иметь доступ или какую ее часть он сможет изменять.
- Кроме того, можно удалить программные модули из БД, чтобы предотвратить изменения структуры форм, отчетов и модулей, сохранив базу данных как **файл MDE**.

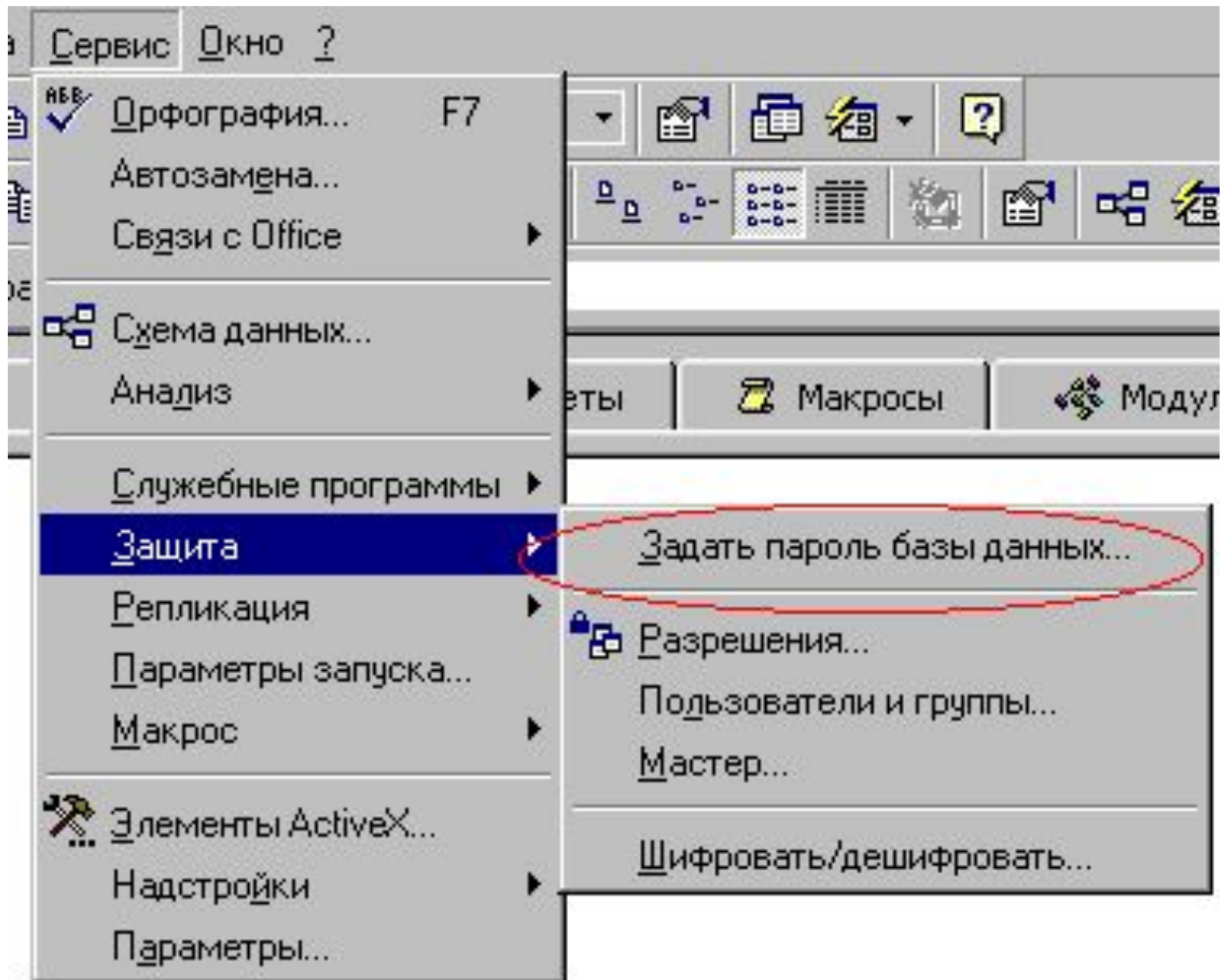


# Установка пароля

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

Этот способ достаточно надежен (Access шифрует пароль), но он действует только при открытии БД. После открытия БД все объекты становятся доступными для пользователя.

Для БД, которая совместно используется небольшой группой пользователей или на



# пользователей

Наиболее гибкий способ защиты БД - защита на уровне пользователей.

От пользователей требуется идентифицировать себя и ввести пароль, когда они запускают Microsoft Access.

Внутри файла рабочей группы они идентифицируются как члены группы. Access по умолчанию создает две группы: администраторы (группа «Admins») и пользователи (группа «Users»).

Допускается также определение других групп

Группам и пользователям предоставляются разрешения на доступ, определяющие возможность их доступа к каждому объекту базы данных.

Например, члены группы «Users» могут иметь разрешения на просмотр, ввод или изменение данных в таблице «Клиенты», но им не будет разрешено изменять структуру этой таблицы.

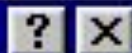
Члены группы «Admins» имеют все разрешения на доступ ко всем объектам базы данных. Имеется возможность установить более разветвленную структуру управления,

# Преимущества защиты на уровне пользователей

- Защищается программа как интеллектуальная собственность.
- Приложение защищается от повреждения из-за неумышленного изменения пользователями программ или объектов, от которых зависит работа приложения.
- Защищаются конфиденциальные сведения в базе данных.



# Пользователи и группы



Пользователи | Группы | Изменение пароля

Пользователь

имя:

Admin

Создать...

Удалить

Снять пароль

Группы

Имеющиеся группы:

Admins  
Users

Добавить >>

<< Удалить

Участие в группе:

Admins  
Users

Отчет о защите

OK

Отмена

Применить