

*Г. Д. Жангисина
Т. Хакимова*

**ТЕОРИЯ И МЕТОДИКА
КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ
ДЛЯ ЗАДАЧ БАЗЫ ДАННЫХ
И ГЛОБАЛЬНОЙ СЕТИ**

Учебное пособие

**Алматы
2019**

ББК 32.973.202я73

Ж 27

**Учебное пособие рекомендовано Ученым Советом Алматинский
Технологический Университет**

РЕЦЕНЗЕНТЫ:

Кутенов В. П. — доктор технических наук,
профессор Московского энергетического института;
Караев Ж. А. — доктор педагогических наук, профессор,
1-й проректор, КазГосЖенПИ;
Карымсаков К. У. — кандидат технических наук, зав. кафедры
«Инженерная графика и прикладная механика»,
Алматинского Технологического Университета.

Ж 27 Жангисина Г. Д., Хакимова Т. Х.

**Теория и методика компьютерного моделирования для задач
базы данных и глобальной сети:** Учеб. пособие.—Алматы,
«NURPRESS», 2019.— 110 с.

ISBN 9965-813-21-3

Переход на кредитную технологию обучения в высшей школе требует от студентов самостоятельной работы над изучаемым предметом. Данная работа именно предназначена для самостоятельного освоения материала, компьютерного моделирования обработки данных с помощью базы данных и информации глобальной сети.

Данная работа состоит из 4 (четырёх) разделов. Первый раздел содержит теоретический и методический материал "Базы данных". Второй раздел "Модели задач базы данных". Третий раздел "Связь информационных сетей". Четвертый раздел "Защита информации в глобальной сети Internet".

Пособие может быть рекомендовано студентам всех специальностей, изучающих компьютерное моделирование, а также будет полезно магистрантам, аспирантам, преподавателям вузов, научным работникам и работникам высшей школы.

Пособие рекомендовано для студентов вузов по курсу «ИНФОРМАТИКА».

ББК 32.973.202я73

**Ж 2404070000
00(05)-07**

ISBN 9965-813-21-3

© Жангисина Г.Д., Хакимова Т.Х., 2019

© Издательство «NURPRESS», 2019

© Дизайнер Карпун К.К., 2019

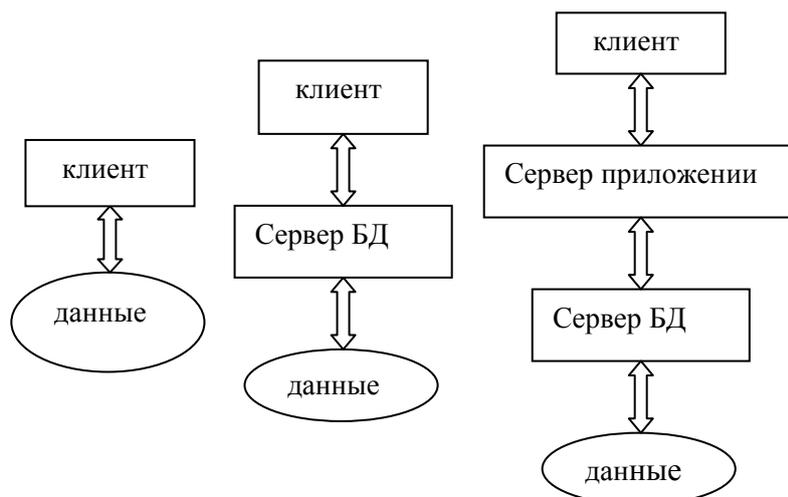
ВВЕДЕНИЕ

1. База данных

Базами данных (БД) называют электронные хранилища информации, доступ к которым осуществляется с помощью одного или нескольких компьютеров.

Базы данных обслуживаются специальными программами-системами управления базами данных (СУБД). СУБД — это программные средства, предназначенные для создания, наполнения, обновления и удаления баз данных.

По своей архитектуре СУБД делятся на одно-, двух- и многозвенные (в простом варианте — трехзвенная).



В зависимости от расположения отдельных частей СУБД различают локальные и сетевые СУБД. К сетевым относятся файл — серверные, клиент — серверные и распределенные СУБД.

Все части локальной (настольной) СУБД размещаются на компьютере пользователя базы данных и клиент обеспечивает необходимую логику управления и их визуализацию.

Настольные СУБД не содержат специальных приложений и сервисов, управляющих данными, взаимодействие с данными

осуществляется с помощью файловых сервисов операционной системы. Часто такие СУБД имеют в своем составе и средства разработки приложений. Эти средства ориентированы на работу с данными в формате этой СУБД и позволяют создать комфортный пользовательский интерфейс. Обработка данных полностью осуществляется в пользовательском (клиентском) приложении.

Недостатком чисто настольной СУБД является наличие у каждого пользователя своей копии локальной БД и частая синхронизация копии данных.

Следующий шаг настольных СУБД это — их сетевая многопользовательская версия, позволяющая обрабатывать данные, находящиеся в общедоступном хранилище, например, в одном или нескольких каталогах достаточно мощной машины, несколькими пользователями одновременно. Такой компьютер называется файл-сервером. Между локальными и файл — серверными СУБД нет особых различий, в них все части собственно СУБД (кроме данных) находятся на компьютере клиента и по архитектуре они обычно однозвенные (уровневые).

Достоинством файл — серверных СУБД является относительная простота создания и обслуживания, который сводится лишь к развертыванию локальной сети.

Недостатком файл — серверных систем является нагрузка на сеть, так как при работе к клиенту передается весь файл и затем в созданной локальной копии данных отыскивается нужная запись.

Наиболее популярными настольными СУБД являются dBase корпорации dBase, Paradox корпорации Corel, FoxPro и Access корпорации Microsoft, а также Microsoft Data Engine — по существу серверная СУБД, предоставляющая «облегченную» версию Microsoft SQL Server, но предназначенная в настольных системах и небольших группах.

Клиент — серверные (двухзвенные) схемы снижают нагрузку на сеть, так как клиент общается с данными через специализированного посредника — сервер базы данных, который размещается на машине с данными. Сервер БД принимает запрос от клиента, отыскивает в данных нужную запись и передает клиенту нужную единственную запись. Запрос к серверу формируется на процедурном языке структурированных запросов (SQL,

Structured Query Language), поэтому часто серверы БД называются SQL — серверами. Термин «непроцедурный» означает, что на этом языке можно сформулировать, что именно нужно сделать с данными, но нельзя проинструктировать, как это следует сделать. Иначе, в языке SQL отсутствуют алгоритмические конструкции, такие как метки, операторы цикла, условные переходы и т.п.

Запросом (query) к базе данных называются любые манипуляции с данными в базах данных, такие как выбор, вставка, обновление данных, изменение или выбор метаданных.

Информация о таблицах, их колонках (имена, тип данных, длина поля), первичных и внешних ключах, а также иных объектов базы данных, называются метаданными (metadata).

Серверы БД представляют собой относительно сложные программы, изготавливаемые различными фирмами. Перечислим наиболее популярные серверные СУБД, например, Oracle производства корпорации Oracle, Microsoft SQL Server производства корпорации Microsoft, СУБД Informix корпорации Informix, Sybase SQL Server корпорации Sybase, DB2 корпорации IBM и сервер InterBase корпорации Inprise, который поставляется вместе с Delphi в комплектации Enterprise.

Клиент — серверном приложений вся вычислительная нагрузка ложится на сервер БД, который обладает скудным набором средств для реализации сложной бизнес — логики (обработка данных, это — хранимые процедуры, триггеры, просмотры и т.д.). Поэтому разработчики вынуждены усложнять программный код клиентского ПО, а с усложнением клиентского ПО возрастает вероятность ошибок и усложняется его обслуживание при большом числе удаленных клиентских компьютеров (более тысячи).

Многозвенные (в простом варианте — трехзвенное распределенное) архитектура приложений баз данных вызвана необходимостью обрабатывать на стороне сервера запросы от большого числа удаленных клиентов. В многозвенной архитектуре «тонкие» клиенты представляют собой простейшие приложения, обеспечивающие лишь передачу данных, их локальное кэширование (кэш — буфер), представление средствами пользовательского интерфейса, редактирование и простейшую обработку.

ПО промежуточного слоя называется сервером приложений, принимает запросы клиентов, обрабатывает их в соответствии с запрограммированными правилами бизнес — логики, при необходимости преобразует в форму, удобную для сервера БД и отправляет серверу. Сервер БД выполняет полученные запросы и отправляет результаты серверу приложений, который адресует данные клиентам.

Таким образом, простая трехзвенная модель содержит следующие элементы: «тонкие» клиенты, сервер приложений, сервер БД.

Компоненты и объекты Delphi, обеспечивающие разработку многозвенных приложений, объединены общим названием DataSnap. В Delphi 4 и 5 эти компоненты объединялись под названием MIDAS (Multi — tier Distributed Applications Services — сервисы многозвенных распределенных приложений).

1.1. Проектирование базы данных

Жизненный цикл любого программного продукта, в том числе и системы управления базой данных, состоит из стадий проектирования, реализации и эксплуатации.

Значительным фактором в жизненном цикле приложения, работающего с базой данных, является стадия проектирования. От того, насколько тщательно продумана структура базы, насколько четко определены связи между ее элементами, зависит производительность системы и ее информационная насыщенность.

Требования к базам данных.

Спроектированная база данных:

- Удовлетворяет всем требованиям пользователей к содержанию базы данных. Перед проектированием базы необходимо провести обширные исследования требований пользователей к функционированию базы данных.

- Гарантирует непротиворечивость и целостность данных. При проектировании таблиц нужно определить их атрибуты и некоторые правила, ограничивающие возможность ввода пользователем неверных значений.

- Обеспечивает естественное, легкое для восприятия структурирование информации. Качественное построение базы по-

звolyет делать запросы к базе более “прозрачными” и легкими для понимания; снижается вероятность внесения некорректных данных и улучшается качество сопровождения базы.

- Удовлетворяет требованиям пользователей к производительности базы данных.

Основные концепции реляционных баз данных

Один из наиболее удобных типов баз данных — реляционные (relational — отношение, связь) базы данных. Реляционная модель данных была предложена Е.Ф.Коддом в 1969 г. на основе математической теории отношений, а основные концепции которых описаны в 1970 году. Согласно Кодду, реляционная база данных представляет собой хранилище данных, содержащее набор двумерных таблиц. Любая таблица реляционной базы данных состоит из строк (запись) и столбцов (поля).

Строки таблицы содержат сведения о представленных в ней фактах (документах, людях, в общем — об однотипных объектах). На пересечении столбца и строки находятся конкретные значения содержащихся в таблице данных.

Данные в таблицах удовлетворяют следующим принципам:

1. Каждое значение, содержащееся на пересечении строки и колонки, должно быть атомарным (не расчленяется на несколько значений).
2. Значения данных в одной колонке должны принадлежать к одному типу, доступному для данной СУБД.
3. Каждая запись в таблице уникальна, то есть в таблице не существует двух записей с полностью совпадающим набором значений ее полей.
4. Каждое поле имеет уникальное имя.
5. Последовательность полей и записей в таблице несущественна.

Несмотря на то, что строки таблиц считаются неупорядоченными, любая СУБД позволяет сортировать строки и колонки в выборках из нее нужным пользователю способом.

Имена колонок для данной таблицы уникальны, но не обязаны быть уникальными для всей базы данных.

Реляционная база данных представляет собой совокупность отношений, содержащих всю необходимую информацию и объединенных различными связями.

К примеру, таблица Список из базы данных Друзья состоит из следующих полей:

Перс_Ном
Фамилия
Имя
Отчество

Поскольку строки в таблице не упорядочены, нам нужна колонка или набор колонок для уникальной идентификации каждой строки. Поле или столбец, который может быть использован для однозначной идентификации конкретной строки или записи, называется первичным ключом (primary key). Первичный ключ таблицы обязан содержать уникальные непустые значения для каждой строки.

Если первичный ключ состоит более чем одной колонки, он называется составным первичным ключом (composite primary key).

Обычно база данных состоит из нескольких связанных таблиц. Например, информация о персоне в предыдущей таблице может дополняться в другой таблице Личные_Данные:

Перс_Ном
Год_Рожд
Адрес
Телефон
Место_Работы

Связанные отношениями таблицы взаимодействуют по принципу *главная (master) — детальная (detail)*. Одна и та же таблица может быть *главной (родительской)* по отношению к одной таблице БД и *детальной (дочерней)* по отношению к другой.

Установив связь по первичному ключу, мы можем уточнить адрес и фамилию расположенных в разных таблицах.



Индексы отличаются от первичных ключей тем, что не требуют неперменной уникальности значений входящих в их состав. Они устанавливаются по полям, которые часто используются при поиске и сортировке данных: индексы помогут системе быстрее найти нужные данные или отсортировать их в нужной последовательности.

Утверждение о том, что первичный ключ должен содержать уникальные непустые значения для данной таблицы является одним из правил *ссылочной целостности*. Многие СУБД контролируют уникальность первичных ключей и при попытке присвоить первичному ключу значение, уже имеющееся в другой записи, СУБД сгенерирует диагностическое сообщение.

Если две таблицы связаны соотношением master — detail, внешний ключ detail — таблицы должен содержать только те значения, которые уже имеются среди значений первичного ключа master — таблицы. Если корректность внешних ключей не контролируется СУБД, можно говорить о нарушении ссылочной целостности. При удалении из таблицы Друзья запись, имеющую с ней detail — запись в таблице Личные_Данные, приведет к тому, что в таблице Личные_Данные окажутся записи, неизвестно к кому относящиеся. Если СУБД контролирует корректность значений внешних ключей, то при попытке присвоить внешнему ключу значение, отсутствующее среди значений первичных ключей master — таблицы, либо при удалении записей master — таблицы, приводящих к нарушению ссылочной целостности, СУБД сгенерирует диагностическое сообщение.

Иными словами СУБД автоматически отслеживает вопросы “ненарушения” связей между отношениями, а именно:

- если Вы попытаетесь вставить в подчиненную таблицу запись, для внешнего ключа которой не существует соответствия

в главной таблице (например, там нет еще записи с таким первичным ключом), СУБД сгенерирует ошибку;

- если Вы попытаетесь удалить из главной таблицы запись, на первичный ключ которой имеется хотя бы одна ссылка из подчиненной таблицы, СУБД также сгенерирует ошибку.

- если Вы попытаетесь изменить первичный ключ записи главной таблицы, на которую имеется хотя бы одна ссылка из подчиненной таблицы, СУБД также сгенерирует ошибку.

Существует два подхода к удалению и изменению записей из главной таблицы:

1. Запретить удаление всех записей, а также изменение первичных ключей главной таблицы, на которые имеются ссылки подчиненной таблицы.

2. Распространить всякие изменения в первичном ключе главной таблицы на подчиненную таблицу, а именно:

- А) если в главной таблице удалена запись, то в подчиненной таблице должны быть удалены все записи, ссылающиеся на удаляемую;

- В) если в главной таблице изменен первичный ключ записи, то в подчиненной таблице должны быть изменены все внешние ключи записей, ссылающихся на изменяемую.

1.2. Шаги проектирования базы данных

I. Определение информационных потребностей БД. Включает опрос будущих пользователей, чтобы понять и задокументировать их требования. Следует выяснить следующие вопросы:

- кто будет вводить данные в базу и в какой форме; как часто будут изменяться данные;

- достаточно ли будет для Вашей предметной области одной базы или Вам потребуется несколько баз данных с различными структурами;

- какая информация является наиболее чувствительной к скорости ее извлечения и изменения.

II. Проанализировать объекты реального мира, которые необходимо смоделировать в базе данных. Сформировать из этих объектов сущности и характеристики этих сущностей (например, для сущности “деталь” характеристиками могут быть “название”, “цвет”, “вес” и т.п.) и сформировать их список.

III. Установление соответствия между сущностями и характеристиками предметной области, отношениями и атрибутами в нотации выбранной СУБД. Поскольку каждая сущность реального мира обладает некими характеристиками, в совокупности образующими полную картину ее проявления, можно поставить им в соответствие набор отношений (таблиц) и их атрибутов (полей).

Перечислив все таблицы и их поля, уже на этом этапе можно начать устранять излишние позиции. Каждое поле должно появляться только один раз; и Вы должны решить, какая таблица будет являться владельцем какого набора полей.

IV. Определяются поля, которые уникальным образом идентифицируют каждый объект. Это необходимо для того, чтобы система могла получить любую единичную строку таблицы. Вы должны определить первичный ключ для каждого из таблиц. Если нет возможности идентифицировать запись с помощью одного поля, то первичный ключ нужно сделать составным — из нескольких полей. Хорошим примером может быть первичный ключ в таблице работников, состоящий из фамилии, имени и отчества. Первичный ключ гарантирует, что в таблице не будет содержаться двух одинаковых строк. Во многих СУБД имеется возможность помимо первичного, определять еще ряд уникальных ключей. Отличие уникального ключа от первичного состоит в том, что уникальный ключ не является главным идентифицирующим фактором записи и на него не может ссылаться внешний ключ другой таблицы. Его главная задача — гарантировать уникальность значения поля.

V. Выработка правил, которые будут устанавливаться, и поддерживать целостность данных. Такие определенные правила в клиент — серверных СУБД поддерживаются автоматически — сервером баз данных; в локальных СУБД их поддержание возлагают на пользовательское приложение.

VI. Устанавливаются связи между объектами (таблицами и столбцами) и производится очень важная операция для исключения избыточности данных — нормализация таблиц.

Каждый из различных типов связей должен быть смоделирован в базе данных. Существует несколько типов связей:

- связь “один-к-одному”,

- связь “один-ко-многим”,
- связь “много-к-одному”,
- связь “многие-ко-многим”.

Связь “один-к-одному” означает, что каждая запись одной таблицы соответствует только одной записи в другой таблице. Например, если рассматривать таблицы, одна из которых содержит данные о сотрудниках предприятия, а вторая — профессиональные сведения, то можно сказать, что между этими таблицами существует отношение “один-к-одному”, поскольку для одного человека, информация о котором содержится в первой таблице, может существовать только одна запись, содержащая профессиональные сведения, во второй таблице. Такую связь удобно устанавливать тогда, когда невыгодно держать разные по размеру (или по другим критериям) данные в одной таблице.

Наиболее часто встречающимся типом отношений в базе данных является отношение “один-к-одному”. Она реализуется парой «внешний ключ — первичный ключ», т.е. когда определен внешний ключ, ссылающийся на первичный ключ другой таблицы. Именно это связь описывает механизм классификаторов. Имеется справочная таблица, содержащая названия, имена и т.п. и некие коды, причем первичным ключом является код. В таблице собирающей информацию — назовем ее информационной таблицей — определяется внешний ключ, ссылающийся на первичный ключ классификатора. После этого в нее заносится не название классификатора, а код. В качестве иллюстрации данного типа отношения можно обратиться к таблицам, содержащим информацию о клиентах предприятия и сделанных ими заказах. В качестве других примеров могут быть рассмотрены отношения между предприятием и работающими на нем сотрудниками.

Отношение “много-к-одному” аналогично рассмотренному ранее типу “один-к-одному”. Тип отношения между объектами зависит от вашей точки зрения. Например, если вы будете рассматривать отношение между сделанными заказами и клиентами, то получите отношение “много-к-одному”.

Связь “многие-ко-многим” в явном виде в реляционных базах данных не поддерживается. Однако имеется ряд способов косвенной реализации такой связи, которые с успехом возме-

щают ее отсутствие. Один из наиболее распространенных способов заключается во введении дополнительной таблицы, строки которой состоят из внешних ключей, ссылающихся на первичные ключи двух таблиц. Например, имеются две таблицы: КЛИЕНТ и ГРУППА_ИНТЕРЕСОВ. Один человек может быть включен в различные группы, в то время как группа может объединять различных людей. Для реализации такой связи “многие-ко-многим” вводится дополнительная таблица, назовем, ее КЛИЕНТЫ_В_ГРУППЕ, строка которой будет иметь два внешних ключа: один будет ссылаться на первичный ключ в таблице КЛИЕНТ, а другой — на первичный ключ в таблице ГРУППА_ИНТЕРЕСОВ. Таким образом, в таблицу КЛИЕНТЫ_В_ГРУППЕ можно записывать любое количество людей и любое количество групп.

После определения таблиц, полей, индексов и связей между таблицами следует посмотреть на проектируемую базу данных в целом и проанализировать ее, используя правила нормализации, с целью устранения логических ошибок. Важность нормализации состоит в том, что она позволяет разбить большие отношения, содержащие большую избыточность информации, на более мелкие логические единицы, группирующие только данные, объединенные “по природе”. Таким образом, идея нормализации заключается в следующем. Каждая таблица в реляционной базе данных удовлетворяет условию, в соответствии с которым в позиции на пересечении каждой строки и столбца таблицы всегда находится единственное значение, и никогда не может быть множества таких значений.

После применения правил нормализации, логические группы данных располагаются не более чем в одной таблице. Это дает следующие преимущества:

- данные легко обновлять или удалять;
- исключается возможность рассогласования копий данных;
- уменьшается возможность введения некорректных данных.

Теория нормализации основана на концепции нормальных форм. Таблица находится в данной нормальной форме, если она удовлетворяет определенному набору требований. Существует пять нормальных форм, но на практике обычно используется первые три.

Этот процесс включает:

- удаление повторяющихся групп (приведение к 1НФ);
- удаление частично зависимых атрибутов (приведение к 2НФ);
- удаление транзитивно зависимых атрибутов (приведение к 3НФ). Транзитивная зависимость — это зависимость между неключевыми полями.

Нужно понимать, что разбиение информации на более мелкие единицы способствует повышению надежности и непротиворечивости базы данных, но с другой стороны, снижает ее производительность, так как требуются дополнительные затраты процессорного времени (серверного или машины пользователя) на обратное “соединение” таблиц при представлении информации на экране. Иногда для достижения требуемой производительности нужно сделать отход от канонической нормализации, при этом необходимо обеспечить меры по предотвращению противоречивости в данных. Поэтому всякое решение о необходимости того или иного действия по нормализации можно принимать, тщательно проанализировав предметную область и класс поставленной задачи.

VII. Спланировать вопросы надежности данных и, при необходимости, сохранения секретности информации. Для этого необходимо ответить на следующие вопросы:

- кто будет иметь права (и какие) на использование базы данных;
- кто будет иметь права на модификацию, вставку и удаление данных;
- нужно ли делать различие в правах доступа;
- каким образом обеспечить общий режим защиты информации и т.п.

Объекты баз данных

Типы объектов баз данных: таблицы для хранения данных; индексы для сортировки данных и поддержки ключей; ограничения или правила для поддержки ссылочной целостности и ограничения значения данных; триггеры; хранимые процедуры для хранения исполняемого кода.

Таблицы поддерживаются всеми реляционными СУБД и в их полях хранятся данные разных типов.

Рассмотрим таблицу формата Paradox:

Имя поля в таблице формата Paradox представляет собой строку, написание которой подчиняется следующим правилам:

- Имя должно быть не длиннее 25 символов.
- Имя не начинается с пробела, но может содержать пробелы. Если собираетесь в будущем переносить базу данных в другие форматы, то лучше ограничиться девятью символами в названии поля, не включая в него пробелы.
- Имя не должно содержать квадратные, круглые или фигурные скобки, тире.

Имя не должно быть только символом #, но символ может присутствовать в имени среди других символов. Paradox поддерживает точку (.) в названии поля, но лучше ее избегать.

Поля таблиц формата Paradox могут иметь следующий тип (для ввода типа поля можно набрать только подчеркнутые буквы или цифры):

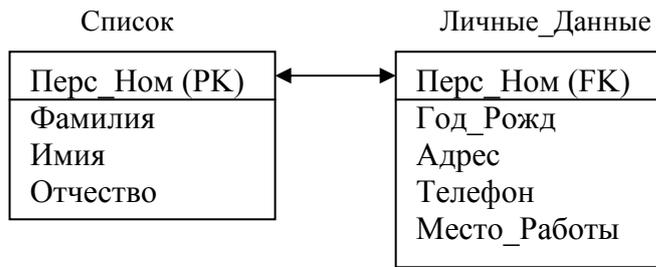
<u>A</u> lpha	строка длиной 1-255 байт, содержащая любые печатаемые символы.
<u>N</u> umber	числовое поле длиной 8 байт, значение которого может быть положительным и отрицательным. Диапазон чисел — от 10^{-308} до 10^{308} с 15 значащими цифрами.
<u>\$</u> (Money)	числовое поле, значение которого может быть положительным и отрицательным. По умолчанию, является форматированным для отображения десятичной точки и денежного знака.
<u>S</u> hort	числовое поле длиной 2 байта, которое может содержать только целые числа в диапазоне от -32768 до 32767.
<u>L</u> ong <u>I</u> nteger	числовое поле длиной 4 байта, которое может содержать целые числа в диапазоне от -2147483648 до 2147483648.

# (BCD)	числовое поле, содержащее данные в формате BCD (Binary Coded Decimal). Скорость вычислений немного меньше, чем в других числовых форматах, однако точность — гораздо выше. Может иметь 0-32 цифр после десятичной точки.
Date	поле даты длиной 4 байта, которое может содержать дату до 31 декабря 9999 г. нашей эры. Корректно обрабатывает високосные года и имеет встроенный механизм проверки правильности даты.
Time	поле времени длиной 4 байта, содержит время в миллисекундах от полуночи и ограничено 24 часами.
@ (Timestamp)	обобщенное поле даты длиной 8 байт — содержит и дату и время.
Memo	поле для хранения символов, суммарная длина которых более 255 байт. Может иметь любую длину. При этом размер, указываемый при создании таблицы, означает количество символов, сохраняемых в таблице (1-240) — остальные символы сохраняются в отдельном файле с расширением .MB.
Formatted Memo	поле, аналогичное Мемо, с добавлением возможности задавать шрифт текста. Может иметь любую длину. При этом размер, указываемый при создании таблицы, означает количество символов, сохраняемых в таблице (0-240) — остальные символы сохраняются в отдельном файле с расширением .MB. Однако, Delphi в стандартной поставке не обладает возможностью работать с полями типа Formatted Memo.
Graphic	поле, содержащее графическую информацию. Может иметь любую длину. Смысл размера — такой же, как и в Formatted Memo. Database Desktop “умеет” создавать поля типа Graphic, однако наполнять их можно только в приложении.

<u>O</u> LE	поле, содержащее OLE-данные (Object Linking and Embedding) — образы, звук, видео, документы — которые для своей обработки вызывают создавшее их приложение. Может иметь любую длину. Смысл размера — такой же, как и в Formatted Memo. Database Desktop “умеет” создавать поля типа OLE, однако наполнять их можно только в приложении. Delphi “напрямую” не умеет работать с OLE-полями, но это легко обходится путем использования потоков.
<u>L</u> ogical	поле длиной 1 байт, которое может содержать только два значения — T (true) или F (false). Допускаются строчные и прописные буквы.
<u>±</u> (Autoincrement)	поле длиной 4 байта, содержащее не редактируемое (read-only) значение типа long integer. Значение этого поля автоматически увеличивается (начиная с 1) с шагом 1 — это очень удобно для создания уникального идентификатора записи (физический номер записи не может служить ее идентификатором, поскольку в Парadoxе таковой отсутствует).
<u>B</u> inary	поле, содержащее любую двоичную информацию. Может иметь любую длину. При этом размер, указываемый при создании таблицы, означает количество символов, сохраняемых в таблице (0-240) — остальные символы сохраняются в отдельном файле с расширением .MB. Это аналог поля BLOB в InterBase.
<u>B</u> ytes	строка цифр длиной 1-255 байт, содержащая любые данные.

1.3. Создание базы данных

Для примера создадим простую БД «Друзья» состоящую из двух таблиц Список и Личные_Данные. Для начала создадим папку БД «Друзья».



Для создания объектов (таблицы) базы данных воспользуемся утилитой Database Desktop, во многом похожая на Paradox, входящей в поставку Delphi для интерактивной работы с таблицами различных форматов локальных баз данных — Paradox и dBase, а также SQL-серверных баз данных InterBase, Oracle, Informix, Sybase (с использованием SQL Links).

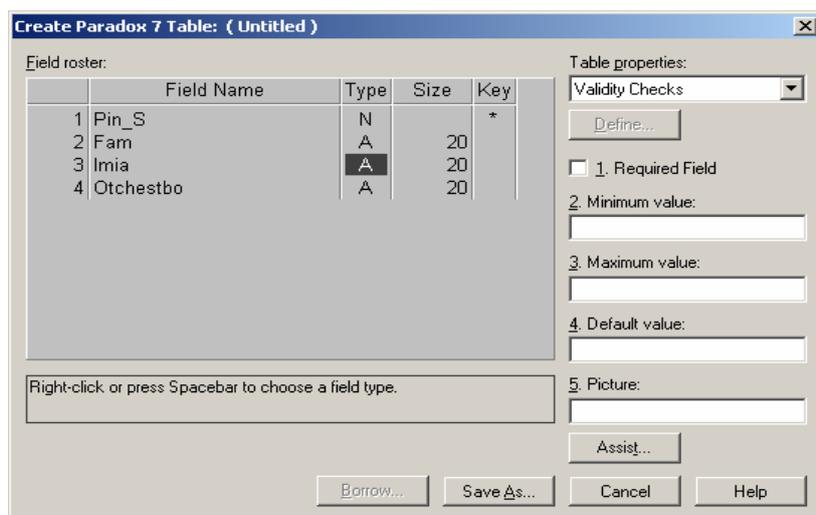
С помощью утилиты DBD в файл — серверных БД можно создавать или изменять структуру таблиц, создавать первичные ключи и индексы, создавать и изменять записи, просматривать их и т.д.

После запуска Tools->Database Desktop в среде Delphi, выберите команду File->Working Directory и установите в появившемся окне ссылку на каталог (папку БД). Но лучше следует создать Alias (псевдоним) базы данных. Псевдоним БД — это просто имя БД. Псевдоним в файл — серверных БД определяет путь доступа к файлам базы данных. При изменении месторасположения БД нам не придется исправлять этот путь в многочисленных компонентах доступа к данным — достаточно изменить его в псевдониме, и все ссылающиеся на псевдоним компоненты будут связаны с новым местом размещения данных.

Для создания алиаса в меню Tools (сервис) выбираем строку Alias Manager...(менеджер алиаса). Потом выбираем кнопку New (создать). В окне Database alias (алиас базы данных) присвоим псевдоним, например druzia. Затем нажмем на кнопку Browse...(просмотр) и выбираем папку «Друзья». Теперь, база алиас “druzia” будет служить ссылкой на папку «Друзья».

Для создания таблицы Spisok в меню File выбираем команду New — Table. В открывшемся окне запишем название полей (Field Name) таблицы, типы полей (Type), размеры полей (Size) и

в колонку ключ (Key) — символ звездочки (*), если по значению этого поля нужно построить первичный ключ.



Пункт Validity Checks в списке Table properties позволяет следить за правильностью вводимых в поля данных, где флажок Required Field потребует обязательного заполнения поля, строки Minimum и Maximum Value будут следить за диапазоном значений, строка Default Value указывает значение поля по умолчанию, а строка Picture при необходимости задает шаблон значения поля (например, ## — ## — ## для тел. номера). Пункт Lookup Table свойства таблицы выбирает данные из другой таблицы, пункт Secondary Indexes строит вторичный индекс, пункт.

Введите название первого поля Pin_S (первый символ названия полей в Paradox DBD всегда вводится прописными буквами) и перейдите к следующей колонке. Нажмите клавишу пробела, чтобы DBD показала список возможных типов, и выберите соответствующий тип. При необходимости в поле Size вставьте размерность и если по колонке Key нужен первичный ключ, нажмите клавишу пробела. Если при записи в поле обязательно должны быть помещены значения, установите флажок в ячейку Required Field — за этим будет следить BDE. Для отображения

русскоязычного текста в списке Table properties выберите пункт Table Language и щелкните на кнопке Modify. В открывшемся диалоговом окне выберите драйвер Pdox ANSI Cyrillic. Этот драйвер следует установить на все поля. После заполнения всех полей таблицы выберите команду Save as и в окне Alias выберите созданный вами псевдоним druzia, в окне «имя файла» запишите Spisok и сохраните. В файл — серверных БД имя таблицы совпадают с именем файла. Названия таблиц и полей набирайте латинскими буквами, чтобы использовать SQL запросы.

Аналогично таблице Spisok создайте и сохраните таблицу Lich_Dan.

2. Модели задач базы данных

2.1. Модель № 1. Создание однотабличной базы данных

Общие сведения

Access — это *система управления базами данных (СУБД)*. Под *системой управления* понимается комплекс программ, который позволяет не только хранить большие массивы данных в определенном формате, но и обрабатывать их, представляя в удобном для пользователей виде. Access дает возможность также автоматизировать часто выполняемые операции (например, расчет заработной платы, учет материальных ценностей и т.п.). С помощью Access можно не только разрабатывать удобные формы ввода и просмотра данных, но и составлять сложные отчеты. Access — это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных. Применение реляционной СУБД помогает упростить структуру данных и таким образом облегчить выполнение работы. Таблицу Access можно связать с данными, хранящимися на другом компьютере или на сервере, а также использовать таблицу, созданную в СУБД Paradox или Dbase. Данные Access очень просто комбинировать с данными Excel.

Все составляющие базы данных, такие, как таблицы, отчеты, запросы, формы и объекты, в Access 97 хранятся в едином дисковом файле. Основным структурным компонентом базы данных является таблица. В таблицах хранятся вводимые нами дан-

ные. Каждая запись таблицы содержит всю необходимую информацию об отдельном элементе базы данных. Например, запись о преподавателе может содержать фамилию, имя, отчество, дату рождения, должность и т.п. При разработке структуры таблицы, прежде всего, необходимо определить названия полей, из которых она должна состоять, типы полей и их размеры. Каждому полю таблицы присваивается уникальное имя, которое не может содержать более 64 символов. Имя желательно делать таким, чтобы функция поля узнавалась по его имени. Далее надо решить, данные какого типа будут содержаться в каждом поле. В Access можно выбирать любые из основных типов данных. Один из этих типов данных должен быть присвоен каждому полю. Значение типа поля может быть задано только в режиме конструктора. Ниже представлены типы данных Access и их описание.

Тип данных	Описание
Текстовый (Значение по умолчанию)	Текст или числа, не требующие проведения расчетов, например номера телефонов (до 255 знаков).
Числовой	Числовые данные различных форматов, используемые для проведения расчетов.
Дата/время	Для хранения информации о дате и времени с 100 по 9999 год включительно.
Денежный	Денежные значения и числовые данные, используемые в математических расчетах, проводящихся с точностью до 15 знаков в целой и до 4 знаков в дробной части.
Поле МЕМО	Для хранения комментариев; до 65535 символов.
Счетчик	Специальное числовое поле, в котором Access автоматически присваивает уникальный порядковый номер каждой записи. Значения полей типа счетчика обновлять нельзя

Логический	Может иметь только 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), связанный или внедренный в таблицу Access.
Гиперссылка	Строка, состоящая из букв и цифр и представляющая адрес гиперссылки. Адрес гиперссылки может состоять максимум из трех частей: текст, выводимый в поле или в элементе управления; путь к файлу (в формате пути UNC) или к странице (адрес URL). Чтобы вставить адрес гиперссылки в поле или в элемент управления, выполните команду Вставка, Гиперссылка.
Мастер подстановок	Создает поле, в котором предлагается выбор значений из списка или из поля со списком, содержащего набор постоянных значений или значений из другой таблицы. Это в действительности не тип поля, а способ хранения поля.

Задание 2.1. Создание базы данных

1. Создайте новую базу данных.
2. Создайте таблицу базы данных.
3. Определите поля таблицы в соответствии с табл.1.
4. Сохраните созданную таблицу.

Таблица.1. Таблица данных

Преподаватели

Имя поля	Тип данных	Размер поля
Код преподавателя	Счетчик	

Фамилия	Текстовый	15
Имя	Текстовый	15
Отчество	Текстовый	15
Дата рождения	Дата/время	Краткий
Должность	Текстовый	9
Дисциплина	Текстовый	11
Телефон	Текстовый	9
Зарплата	Денежный	

Технология выполнения работы

Для создания новой базы данных:

- загрузите Access, в появившемся окне выберите пункт **Новая база данных**;

- в окне "Файл новой базы данных" задайте имя вашей базы (пункт **Имя Файла**) и выберите папку (пункт **Папка**), где ваша база данных будет находиться. По умолчанию Access предлагает вам имя базы db1, а тип файла — *Базы данные Access*. Имя задайте *Преподаватели*, а тип файла оставьте прежним, так как другие типы файлов нужны в специальных случаях;

- щелкните по кнопке <Создать>.

Для создания таблицы базы данных:

- в окне базы данных выберите вкладку *Таблицы*, а затем щелкните по кнопке <Создать>;

- в окне "Новая таблица" выберите пункт **Конструктор** и щелкните по кнопке <ОК>. В результате проделанных операций открывается окно таблицы в режим конструктора (рис. 1), в котором следует определить поля таблицы.

Для определения полей таблицы:

- введите в строку столбца Имя поля имя первого поля *Код преподавателя*;

- в строке столбца "Тип данных" щелкните по кнопке списка и выберите тип данных *Счетчик*. Поля вкладки *Общие* оставьте такими, как предлагает Access.

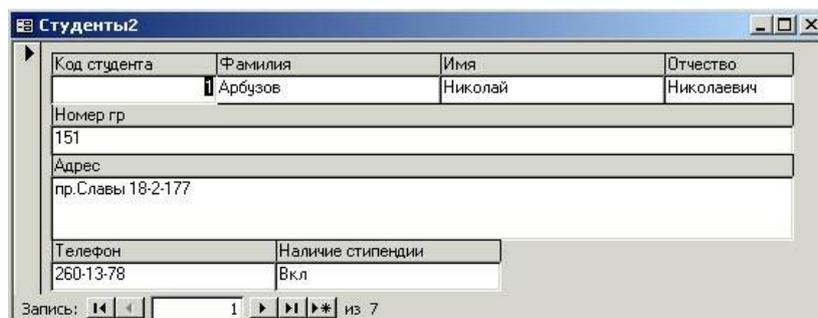


Рис. 1. Окно таблицы в режиме конструктора — в этом режиме вводятся имена и типы полей таблицы.

4. Для сохранения таблицы:

- выберите пункт меню **Файл, Сохранить**;
- в диалоговом окне "Сохранение" введите имя таблицы *Преподаватели'*,
- щелкните по кнопке <ОК>.

Задание 2.2. Заполнение базы данных

1. Введите ограничения на данные, вводимые в поле "Должность"; должны вводиться только слова *Профессор, Доцент* или *Ассистент*.

2. Задайте текст сообщения об ошибке, который будет появляться на экране при вводе неправильных данных в поле "Должность".

3. Задайте значение по умолчанию для поля "Должность" в виде слова *Доцент*.

4. Введите ограничения на данные в поле <Код>; эти данные не должны повторяться.

5. Заполните таблицу данными в соответствии с табл. 2 и проверьте реакцию системы на ввод неправильных данных в поле "Должность".

6. Измените ширину каждого поля таблицы в соответствии с шириной данных.

7. Произведите поиск в таблице преподавателя Миронова.

8. Произведите замену данных: измените заработную плату ассистенту Сергеевой с 450 р, на 470 р.

9. Произведите сортировку данных в поле "Год рождения" по убыванию,

10. Произведите фильтрацию данных по полям "Должность" и "Дисциплина".

Просмотрите созданную таблицу, как она будет выглядеть на листе бумаги при печати.

Таблица 2

Код	Фамилия	Имя	Отчество	Дата рожд.	Должность	Дисциплина	Телефон	Зарплата
1	Истомин	Ремир	Евгеньевич	23.10.54	Доцент	Информатика	110-44-68	890р.
2	Миронов	Павел	Юрьевич	25.07.40	Профессор	Экономика	312-21-40	1200р.
3	Гришин	Евгений	Сергеевич	05.12.67	Доцент	Математика	260-23-65	760р.
4	Сергеева	Ольга	Ивановна	12.02.72	Ассистент	Математика	234-85-69	450р.
5	Емец	Татьяна	Ивановна	16.02.51	Доцент	Экономика	166-75-33	890р.
6	Игнатъева	Татьяна	Павловна	30.05.66	Доцент	Информатика	210-36-98	790р.
7	Миронов	Алексей	Николаевич	30.07.48	Доцент	Физика	166-75-33	890р.

Технология выполнения работы

Для задания условия на значение для вводимых данных:

- войдите в режим *Конструктор* для проектируемой таблицы. Если вы находитесь в окне базы данных, то выберите вкладку *Таблицы* и щелкните по кнопке <Конструктор>. Если вы на-

ходите в режиме таблицы, то щелкните по кнопке на панели инструментов или выполните команду **Вид, Конструктор**;

- в верхней части окна щелкните по полю "Должность";
- в нижней части окна щелкните по строке параметра *Условие на значение*;

- щелкните по кнопке для определения условий на значение при помощи построителя выражений;

- в появившемся окне напишите слово *Профессор*, затем щелкните по кнопке (эта кнопка выполняет функцию ИЛИ), напишите *Доцент*, снова щелкните по этой же кнопке, напишите *Ассистент* и щелкните по кнопке <ОК>. Таким образом, вы ввели условие, при котором в поле "Должность" могут вводиться только указанные значения.

2. В строке *Сообщение об ошибке* введите предложение "Такой должности нет, правильно введите данные".

3. В строке *Значение по умолчанию* введите слово "Доцент".

4. Введите ограничения на данные в поле "Код". Здесь ограничения надо вводить не совсем обычным способом. Дело в том, что коды преподавателей не должны повторяться, а также должна быть обеспечена возможность их изменения (из-за последнего условия в этом поле нельзя использовать тип данных *Счетчик*, в котором данные не повторяются). Для выполнения второго условия пришлось задать в поле "Код" тип данных *Числовой*, а для выполнения первого условия сделайте следующее:

- щелкните по строке параметра *Индексированное поле*;
- выберите в списке пункт **Да (совпадения не допускаются)**;

- перейдите в режим **Таблица**, щелкнув по кнопке на панели инструментов или выполнив команду **Вид, Режим таблицы**. На вопрос о сохранении таблицы щелкните по кнопке <Да>.

5. Введите данные в таблицу в соответствии с табл. 4.3. Попробуйте в поле <Должность> любой записи ввести слово *Лаборант*. Посмотрите, что получилось. На экране должно появиться сообщение; "Такой должности нет, правильно введите данные". Введите правильное слово.

6. Для изменения ширины каждого поля таблицы в соответствии с шириной данных:

- щелкните в любой строке поля "Код";

- выполните команду **Формат, Ширина столбца**;
 - в появившемся окне щелкните по кнопке <По ширине данных>. Ширина поля изменится;
 - проделайте эту операцию с остальными полями.
7. Для поиска в таблице преподавателя Миронова:
- переведите курсор в первую строку поля "Фамилия";
 - выполните команду **Правка, Найти**;
 - в появившейся строке параметра *Образец* введите *Миронов*;
 - в строке параметра *Просмотр* должно быть слово *ВСЕ* (имеется в виду искать по всем записям);
 - в строке параметра *Совпадение* выберите из списка *С любой частью поля*;
 - в строке параметра *Только в текущем поле* установите флажок (должна стоять галочка■);
 - щелкните по кнопке <Найти>. Курсор перейдет на вторую запись и выделит слово *Миронов*;
 - щелкните по кнопке <Найти далее>. Курсор перейдет на седьмую запись и также выделит слово *Миронов*;
 - щелкните по кнопке <Закреть> для выхода из режима поиска.
8. Для замены заработной платы ассистенту Сергеевой с 450 р. на 470 р.:
- переведите курсор в первую строку поля "Зарплата";
 - выполните команду **Правка, Заменить**;
 - в появившемся окне в строке *Образец* введите 450 р.;
 - в строке *Заменить на* введите 470. Обратите внимание на остальные опции — вам надо вести поиск по всем записям данного поля;
 - щелкните по кнопке <Найти далее>. Курсор перейдет на четвертую запись, но здесь не нужно менять данные, поэтому снова щелкните по кнопке <Найти далее>. Курсор перейдет на девятую запись — это то, что нам надо;
 - щелкните по кнопке <Заменить>. Данные будут изменены;
9. Для сортировки данных в поле "Год рождения" по убыванию:
- щелкните по любой записи поля "Год рождения";

- щелкните по кнопке на панели управления или выполните команду **Записи, Сортировка, Сортировка по убыванию**. Все данные в таблице будут отсортированы в соответствии с убыванием значений в поле "Год рождения".

- Для фильтрации данных по полям "Должность" и "Дисциплина":

- щелкните по записи *Доцент* поля "Должность";

- щелкните по кнопке или выполните команду **Записи, Фильтр, Фильтр по выделенному**. В таблице останутся только записи о преподавателях — доцентах;

- щелкните по записи *Информатика* поля "Дисциплина";

- щелкните по кнопке или выполните команду **Записи, Фильтр, Фильтр по выделенному**. В таблице останутся только записи о преподавателях — доцентах кафедры информатики;

- для отмены фильтрации щелкните по кнопке на панели инструментов или выполните команду **Записи, Удалить фильтр**. В таблице появятся все данные.

Для просмотра созданной таблицы:

- щелкните по кнопке или выполните команду **Файл, Предварительный просмотр**. Вы увидите таблицу как бы на листе бумаги;

- закройте окно просмотра.

Задание 2.3. Ввод и просмотр данных посредством формы

1. С помощью Мастера форм создайте форму *Состав преподавателей* (тип — форма один столбец).

2. Найдите запись о доценте Гришине, находясь в режиме формы.

3. Измените зарплату ассистенту Сергеевой с 470 р. на 490 р.

4. Произведите сортировку данных в поле "Фамилия" по убыванию.

5. Произведите фильтрацию данных по полю "Должность".

6. Измените название поля "Дисциплина" на "Преподаваемая дисциплина".

7. Просмотрите форму с точки зрения того, как она будет выглядеть на листе бумаги.

Технология работы

1. Для создания формы Состав преподавателей:

- откройте вкладку *Формы* в окне базы данных;
 - щелкните по кнопке <Создать>;
 - в появившемся окне выберите (подведите курсор мыши и щелкните левой кнопкой) пункт **Мастер форм**;
 - щелкните по значку списка в нижней части окна;
 - выберите из появившегося списка таблицу *Преподаватель*;
 - щелкните по кнопке <ОК>;
 - в появившемся окне выберите поля, которые будут присутствовать в форме. В данном примере присутствовать будут все поля, поэтому щелкните по кнопке ;
 - щелкните по кнопке <Далее>;
 - в появившемся окне уже выбран вид *Форма в один столбец*, поэтому щелкните по кнопке <Далее>;
 - в появившемся окне выберите стиль оформления. Для этого щелкните по словам, обозначающим стили, либо перемещайте выделение стрелками вверх или вниз на клавиатуре. После выбора стиля щелкните по кнопке <Далее>;
 - в появившемся окне задайте имя формы, набрав на клавиатуре параметр *Состав преподавателей*. Остальные параметры в окне оставьте без изменений;
 - щелкните по кнопке <Готово>. Перед вами откроется форма в один столбец. Столбец слева — это названия полей, столбец справа — данные первой записи (в нижней части окна в строке параметра *Запись* стоит цифра "1"). Для перемещения по записям надо щелкнуть по кнопке (в сторону записей с большими номерами) или (в сторону записей с меньшими номерами). Кнопка — это переход на первую запись, кнопка — переход на последнюю запись.
- Для поиска преподавателя Миронова:
- переведите курсор в первую строку поля "Фамилия";
 - выполните команду **Правка, Найти**;
 - в появившемся окне в строке *Образец* введите фамилию *Миронов*;
 - в строке параметра *Просмотр* должно быть слово *ВСЕ* (имеется в виду искать по всем записям);
 - в строке параметра *Совпадение* выберите из списка параметр *С любой частью поля*;

- в строке параметра *Только в текущем поле* установите флажок (должна стоять "галочка");
 - щелкните по кнопке <Найти>. Курсор перейдет на вторую запись и выделит слово *Миронов*;
 - щелкните по кнопке <Найти далее>. Курсор перейдет на седьмую запись и также выделит слово *Миронов*;
 - щелкните по кнопке <Заккрыть> для выхода из режима поиска;
 - Для замены зарплаты ассистенту Сергеевой с 470 р. на 490 р.:
 - переведите курсор в первую строку поля "Зарплата";
 - выполните команду **Правка, Заменить**;
 - в появившемся окне в строке параметра *Образец* введите 470 р.;
 - в строке параметра *Заменить на* введите 490 р. Обратите внимание на остальные опции — вам надо вести поиск по всем записям данного поля;
 - щелкните по кнопке <Найти далее>. Курсор перейдет на четвертую запись, но здесь не нужно менять данные, поэтому снова щелкните по кнопке <Найти далее>. Курсор перейдет на девятую запись — это то, что нам надо;
 - щелкните по кнопке <Заменить>. Данные будут изменены;
 - щелкните по кнопке <Заккрыть>.
4. Для сортировки данных в поле "Год рождения" по убыванию:
- щелкните по любой записи поля "Год рождения";
 - щелкните по кнопке на панели управления или выполните команду **Записи, Сортировка, Сортировка по убыванию**. Все данные в таблице будут отсортированы в соответствии с убыванием значений в поле "Год рождения".
5. Для фильтрации данных по полю "Должность":
- щелкните по записи *Доцент* поля "Должность";
 - щелкните по кнопке или выполните команду **Записи, Фильтр, Фильтр по выделенному**. В форме останутся только записи о преподавателях — доцентах;
 - щелкните по записи *Информатика* поля "Дисциплина";
 - щелкните по кнопке или выполните команду **Записи, Фильтр, Фильтр по выделенному**. В форме останутся только записи о преподавателях — доцентах кафедры информатики;

- для отмены фильтра щелкните по кнопке на панели инструментов или выполните команду **Записи, Удалить фильтр**. В таблице появятся все данные;

6. Измените название поля "Дисциплина" на "Преподаваемая дисциплина". Для этого:

- перейдите в режим конструктора, щелкнув по кнопке на панели инструментов или выполнив команду **Вид, Конструктор**;

- щелкните правой кнопкой мыши в поле "Дисциплина" (на названии поля — оно слева, а строка справа с именем *Дисциплина* — это ячейка для данных, свойства которых мы не будем менять). В появившемся меню выберите пункт **Свойства**. На экране откроется окно свойств для названия поля "Дисциплина";

- щелкните по строке с именем *Подпись*, т.е. там, где находится слово *Дисциплина*;

- сотрите слово "Дисциплина" и введите "Преподаваемая дисциплина";

- для просмотра результата перейдите в режим формы, выполнив команду **Вид, Режим формы**.

7. Для просмотра созданной формы:

- щелкните по кнопке или выполните команду **Файл, Предварительный просмотр**. Вы увидите форму как бы на листе бумаги;

- закройте окно просмотра.

2.2. Модель № 2. Формирование запросов и отчетов для однотабличной базы данных

Формирование запросов на выборку

Запросы являются мощным средством обработки данных, хранимых в таблицах Access 97. С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц. Они также используются в качестве источника данных для форм и отчетов. Запросы позволяют вычислять итоговые значения и выводить их в компактном формате, подобном формату электронной таблицы, а также выполнять вычисления над группами записей. Самостоятельно разработать запросы можно в режиме **конструктора**.

В Access 97 можно создавать следующие типы запросов:

1. запрос на выборку;
2. запрос с параметрами;
3. перекрестный запрос;
4. запрос на изменение (запрос на удаление, обновление и добавление записей на создание таблицы);
5. запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы).

Задание 2.4. Формирование запросов на выборку

1. На основе таблицы *Преподаватели* создайте простой запрос на выборку, в котором должны отображаться фамилии, имена, отчества преподавателей и их должность.

2. Данные запроса отсортируйте по должностям.

3. Сохраните запрос.

4. Создайте запрос на выборку с параметром, в котором должны отображаться фамилии, имена, отчества преподавателей и преподаваемые ими дисциплины, а в качестве параметра задайте фамилию преподавателя и выполните этот запрос для преподавателя *Гришина*.

Технология работы

1. Для создания простого запроса:

- в окне базы данных откройте вкладку *Запросы*;
- в открывшемся окне щелкните по кнопке <Создать>;
- из появившихся пунктов окна "Новый запрос" выберите

Простой запрос и щелкните по кнопке <ОК>;

• в появившемся окне в строке *Таблицы/запросы* выберите таблицу *Преподаватели* (если других таблиц или запросов не было создано, она будет одна в открывающемся списке);

• в окне "Доступные поля" переведите выделение на параметр *Фамилия*,

• щелкните по кнопке. Слово *Фамилия* перейдет в окно "Выбранные поля";

• аналогично в окно "Выбранные поля" переведите поля "Имя", "Отчество", "Должность" (порядок важен — в таком порядке данные и будут выводиться);

• щелкните по кнопке <Далее>;

• в строке параметра *Задайте имя запроса* введите новое имя *Должности преподавателей*;

- щелкните по кнопке <Готово>. На экране появится таблица с результатами запроса.
2. Для сортировки данных:
- щелкните в любой строке поля "Должность";
 - отсортируйте данные по убыванию. Для этого щелкните по кнопке  на панели инструментов или выполните команду **Записи, Сортировка, Сортировка по убыванию**.
3. Для сохранения запроса:
- щелкните по кнопке  или выполните команду **Файл, Сохранить**;
 - закройте окно запроса.
4. Для создания запроса на выборку с параметром:
- создайте запрос на выборку для следующих полей таблицы *Преподаватели*: "Фамилия", "Имя", "Отчество", "Преподаваемая дисциплина". Запрос создавайте аналогично тому, как это делалось в п.1;
 - задайте имя запросу *Преподаваемые дисциплины*;
 - щелкните по кнопке <Готово>. На экране появится таблица с результатами запроса;
 - перейдите в режиме конструктора, щелкнув по кнопке или выполнив команду **Вид, Конструктор**;
 - в строке параметра *Условия отбора* для поля "Фамилия" введите фразу (скобки тоже вводить): [*Введите фамилию преподавателя*];
- выполните запрос, щелкнув по кнопке  на панели инструментов или выполнив команду **Запрос, Запуск**;
 - в появившемся окне введите фамилию *Гришин* и щелкните по кнопке <ОК>. На экране появится таблица с данными о преподавателе *Гришине* — его имя, отчество и преподаваемая им дисциплина;
 - сохраните запрос;
 - закройте окно запроса.

Задание 2.5. На основе таблицы *Преподаватели* создайте отчет с группированием данных по должностям

Технология работы
Для создания отчета:

- откройте вкладку *Отчеты* и щелкните по кнопке <Создать>;
- в открывшемся окне выберите пункт Мастер отчетов;
- щелкните по значку раскрывающегося списка в нижней части окна;
 - выберите из появившегося списка таблицу *Преподаватели*;
 - щелкните по кнопке <ОК>, в появившемся окне выберите поля, которые будут присутствовать в форме. В данном примере присутствовать будут все поля из таблицы, поэтому щелкните по кнопке;
 - щелкните по кнопке <Далее>;
 - в появившемся окне присутствует перечень полей. Проведите выделение на поле "Должность";
 - щелкните по кнопке. Таким образом вы задаете группировку данных по должности;
 - щелкните по кнопке <Далее>;
 - параметры появившегося окна оставим без изменений, поэтому щелкните по кнопке <Далее>;
 - в появившемся окне выберите стиль оформления отчета;
 - щелкните по кнопке <Далее>;
 - в появившемся окне введите название отчета *Преподаватели*;
 - щелкните по кнопке <Готово>. На экране появится сформированный отчет;
 - просмотрите, а затем закройте отчет.

2.3. Модель № 3. Разработка инфологической модели и создание структуры реляционной базы данных

Организация данных

Слово "реляционная" происходит от английского *relation* — отношение. *Отношение* — тематическое понятие, но в терминологии моделей данных отношения удобно изображать в виде таблицы. При этом строки таблицы соответствуют кортежам отношения, а столбцы — атрибутам. Ключом называют любую функцию от атрибутов кортежа, которая может быть использована для идентификации кортежа. Такая функция может быть значением одного, из атрибутов (простой ключ), задаваться ал-

гебраическим выражением, включающим значения нескольких атрибутов (составной ключ). Это означает, что данные в строках каждого из столбцов составного ключа могут повторяться, но комбинация данных каждой строки этих столбцов является уникальной. Например, в таблице Студенты есть столбцы *Фамилии* и *Год рождения*. В каждом из столбцов есть некоторые повторяющиеся данные, т.е. одинаковые фамилии и одинаковые года рождения. Но если студенты, имеющие одинаковые фамилии, имеют разные года рождения, то эти столбцы можно использовать в качестве составного ключа. Как правило, ключ является уникальным, т.е. каждый кортеж определяется значением ключа однозначно, но иногда используют и неуникальные ключи (ключи с повторениями). В локализованной (русифицированной) версии Access 97 вводится термин *ключевое поле*, которое можно трактовать как *первичный ключ*.

В Access 97 можно выделить три типа ключевых полей: простой ключ, составной ключ и внешний ключ.

Одно из важнейших достоинств реляционных баз данных состоит в том, что вы можете хранить логически сгруппированные данные в разных таблицах и задавать связи между ними, объединяя их в единую базу. Для задания связи таблицы должны иметь поля с одинаковыми именами или хотя бы с одинаковыми форматами данных. Связь между таблицами устанавливает отношения между совпадающими значениями в этих полях. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов. Поясним это на примере. Допустим, вам в базе надо хранить, данные о студентах (фамилия, изучаемая дисциплина) и преподавателях (фамилия, номер кафедры, ученая степень, преподаваемая дисциплина). Если хранить данные в одной таблице, то в строке с фамилией студента, изучающего конкретную дисциплину, будут храниться все атрибуты преподавателя, читающего эту дисциплину. Это же огромная избыточность данных. А если хранить данные о студенте в одной таблице, о преподавателе — в другой и установить связь между полями "Читаемая дисциплина" — "Изучаемая дисциплина" (фактически это одинаковые поля), то избыточность хранимых данных многократно уменьшится без ущерба для логической организации информации.

В Access 97 можно задать три вида связей между таблицами; *Один-ко-многим*, *Многие-ко-многим* и *Один-к-одному*.

Связь *Один-ко-многим* — наиболее часто используемый тип связи между таблицами. В такой связи каждой записи в таблице А может соответствовать несколько записей в таблице В (поля с этими записями называют *внешними ключами*), а запись в таблице В не может иметь более одной соответствующей ей записи в таблице А.

При связи *Многие-ко-многим* одной записи в таблице А может соответствовать несколько записей в таблице В, а одной записи в таблице В — несколько записей в таблице А. Такая схема реализуется только с помощью третьей (связующей) таблицы, ключ которой состоит по крайней мере из двух полей, одно из которых является общим с таблицей А, а другое — общим с таблицей В.

При связи *Один-к-одному* запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением *Один-к-одному* применяют для разделения очень широких таблиц, для отделения части таблицы в целях ее защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

Тип создаваемой связи зависит от полей, для которых определяется связь:

- связь *Один-ко-многим* создается в том случае, когда только одно из полей является ключевым или имеет уникальный индекс, т.е. значения в нем не повторяются;
- связь *Один-к-одному* создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы;
- связь *Многие-ко-многим* фактически представляет две связи типа *один-ко-многим* через третью таблицу, ключ которой состоит, по крайней мере, из двух полей, общих для двух других таблиц.

Целостность данных

Целостность данных означает систему правил, используемых в СУБД Access для поддержания связей между записями в свя-

занных таблицах (таблиц, объединенных с помощью связи), а также обеспечивает защиту от случайного удаления или изменения связанных данных. Контролировать целостность данных можно, если выполнены следующие условия:

- связанное поле (поле, посредством которого осуществляется связь) одной таблицы является ключевым полем или имеет уникальный индекс;
- связанные поля имеют один тип данных. Здесь существует исключение. Поле счетчика может быть связано с числовым полем, если оно имеет тип *Длинное целое*;
- обе таблицы принадлежат одной базе данных Access. Если таблицы являются связанными, то они должны быть таблицами Access. Для установки целостности данных база данных, в которой находятся таблицы, должна быть открыта. Для связанных таблиц из баз данных других форматов установить целостность данных невозможно.

Задание 2.6. Создание инфологической и логической моделей базы данных

1. Разработайте информационно-логическую модель реляционной базы данных.
2. Разработайте логическую модель реляционной базы данных.

Технология работы

1. Перед разработкой информационно-логической модели реляционной базы данных рассмотрим, из каких информационных объектов должна состоять эта база данных. Можно выделить три объекта, которые не будут обладать избыточностью, — *Студенты*, *Дисциплины* и *Преподаватели*. Представим состав реквизитов этих объектов в виде "название объекта (перечень реквизитов)": *Студенты* (код студента, фамилия, имя, отчество, номер группы, дата рождения, стипендия, оценки). *Дисциплины* (код дисциплины, название дисциплины), *Преподаватели* (код преподавателя, фамилия, имя, отчество, дата рождения, телефон, заработная плата).

Рассмотрим связь между объектами *Студенты* и *Дисциплины*. Студент изучает несколько дисциплин, что соответствует многозначной связи и отражено на рис. 4.9 двойной стрелкой.

Понятно, что каждая дисциплина изучается множеством студентов. Это тоже многозначная связь, обозначаемая двойной стрелкой (связь "один" обозначена одинарной стрелкой). Таким образом, связь между объектами *Студенты* и *Дисциплины* — *Многие-ко-многим* (M : N).

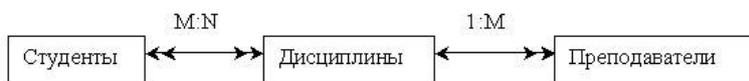


Рис. 4.9. Типы связей между объектами *Студенты*, *Дисциплины* и *Преподаватели*.

Множественные связи усложняют управление базой данных, например, в СУБД Access 97 при множественных связях нельзя использовать механизм каскадного обновления. Поэтому, использовать такие связи нежелательно и нужно строить реляционную модель, не содержащую связей типа *Многие-ко-многим*. В Access 97 для контроля целостности данных с возможностью каскадного обновления и удаления данных необходимо создать вспомогательный объект связи, который состоит из ключевых реквизитов связываемых объектов и который может быть дополнен описательными реквизитами. В нашем случае таким новым объектом для связи служит объект *Оценки*, реквизитами которого являются код студента, код дисциплины и оценки. Каждый студент имеет оценки по нескольким дисциплинам, поэтому связь между объектами *Студенты* и *Оценки* будет *Один-ко-многим* (1:M). Каждую дисциплину сдает множество студентов, поэтому связь между объектами *Дисциплины* и *Оценки* также будет *Один-ко-многим* (1:M). В результате получаем информационно-логическую модель базы данных, приведенную на рис. 4.10.

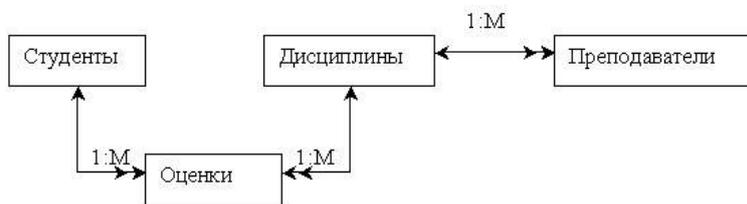


Рис. 4.10. Информационно-логическая модель реляционной базы данных.

2. В реляционной базе данных в качестве объектов рассматриваются отношения, которые можно представить в виде таблиц. Таблицы между собой связываются посредством общих полей, т.е. одинаковых по форматам и, как правило, по названию, имеющихя в обеих таблицах. Рассмотрим, какие общие поля надо ввести в таблицы для обеспечения связности данных. В таблицах *Студенты* и *Оценки* таким полем будет "Код студента", в таблицах *Дисциплины* и *Оценки* — "Код дисциплины", в таблицах *Преподаватели* и *Дисциплины* — "Код дисциплины". Выбор цифровых кодов вместо фамилий или названий дисциплин обусловлен меньшим объемом информации в таких полях: например, число "2" по количеству символов значительно меньше слова "математика". В соответствии с этим логическая модель базы данных представлена на рис. 4.11, где жирными буквами выделены ключевые поля.



Рис. 4.11. Логическая модель базы данных.

Задание 2. Создание реляционной базы данных

1. Создайте базу данных *Деканат*.
2. Создайте структуру таблицы *Студенты*.
3. Создайте структуру таблицы *Дисциплины*.
4. Измените структуру таблицы *Преподаватели*.
5. Создайте структуру таблицы *Оценки*.
6. Разработайте схему данных, т.е. создайте связи между таблицами.

Технология работы

1. Создайте базу данных *Деканат*, выполнив следующие действия:

- загрузите Access, в появившемся окне выберите пункт *Новая база данных*, затем щелкните по кнопке <ОК>;
- в окне <Файл новой базы данных> задайте имя (пункт *Имя файла*) и выберите папку (пункт *Папка*), где ваша база будет находиться. По умолчанию Access предлагает имя базы *dbl*, а тип файла — *Базы данных Access*. Имя задайте *Деканат*, а тип файла оставьте прежним, так как другие типы файлов нужны в специальных случаях;
- щелкните по кнопке <Создать>.

2. Создайте структуру таблицы *Студенты*. Для этого:

- в окне базы данных выберите вкладку *Таблицы*, а затем щелкните по кнопке <Создать>;
- в окне "Новая таблица" выберите пункт *Конструктор* и щелкните по кнопке <ОК>. В результате проделанных операций открывается окно таблицы в режиме конструктора, в котором следует определить поля таблицы;
- определите поля таблицы в соответствии с табл. 4.3;

Таблица 4.3.

Имя поля	Тип данных	Размер поля
Код студента	Числовой	Целое
Фамилия	Текстовый	15
Имя	Текстовый	12
Отчество	Текстовый	15
Номер группы	Числовой	Целое
Телефон	Текстовый	9
Стипендия	Логический	Да/Нет

- в качестве ключевого поля задайте "Код студента". Для этого щелкните по полю "Код студента" и по кнопке на панели инструментов или выполните команду **Правка, Ключевое поле**;
- закройте таблицу, задав ей имя *Студенты*.

Примечание. Заполнять таблицу данными пока не требуется, это будет сделано в режиме формы.

3. Создайте структуру *таблицы Дисциплины* аналогично п. 2 в соответствии с табл. 4.4.

Таблица 4.4

Имя поля	Тип данных	Размер поля
Код дисциплины	Числовой	Целое
Название дисциплины	Текстовый	30

В качестве ключевого поля задайте "Код дисциплины". Заполняться эта таблица будет также в режиме формы.

4. Структура таблицы *Преподаватели* уже создана в работе 1 и заполнена данными, поэтому для работы используйте эту таблицу с одним лишь изменением — в соответствии с рис. 4.11 в структуру таблицы надо добавить поле "Код дисциплины" и заполнить его в соответствии с данными табл. 4.4.

5. Создайте структуру таблицы *Оценки* аналогично п. 2 в соответствии с табл. 4.5.

Таблица 4.5

Имя поля	Тип данных	Размер поля
Код студента	Числовой	Целое
Код дисциплины	Числовой	Целое
Оценки	Числовой	Байт

В этой таблице задавать ключевое поле не надо, так как данные во всех полях могут повторяться. Эта таблица, аналогично предыдущим, будет заполняться в режиме формы.

6. Разработайте схему данных, т.е. создайте связи между таблицами. Для этого:

- щелкните по кнопке на панели инструментов или выполните команду **Сервис, Схема данных**. На экране появится окно "Схема данных";
- щелкните по кнопке на панели инструментов или выполните команду **Связи, Добавить таблицу**;
- в появившемся окне будет выделено название одной таблицы. Щелкните по кнопке <Добавить>;
- переведите выделение на имя следующей таблицы и щелкните по кнопке <Добавить>. Аналогично добавьте оставшиеся две таблицы;
- закройте окно, щелкнув по кнопке <Закреть>;

- создайте связь между таблицами *Дисциплины* и *Оценки*. Для этого подведите курсор мыши к полю "Код дисциплины" в таблице *Дисциплины* щелкните левой кнопкой мыши и, не отпуская ее, перетащите курсор на поле "Код дисциплины" в таблице *Оценки*, а затем отпустите кнопку мыши. На экране откроется окно "Связи";

- установите флажок ("галочку") в свойстве *Обеспечение целостности данных*, щелкнув по нему;

- установите флажок в свойстве *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*;

Примечание. Задание каскадного обновления связанных полей и каскадного удаления связанных записей позволит вам отредактировать записи только в *таблице Дисциплины*, а в таблице *Оценки* эти действия будут со связанными записями выполняться автоматически. Например, если вы удалите из таблицы *Дисциплины* один предмет, то в таблице *Оценки* удалятся все строки, связанные с этим предметом.

- щелкните по кнопке <Создать>. Связь будет создана;

- аналогично создайте связи между полем "Код дисциплины" в *таблице Дисциплины* и полем "Код дисциплины" в *таблице Преподаватели*, а также между полем "Код студента" в *таблице Студенты* и полем "Код студента" в *таблице Оценки*. Результат представлен на рис. 4.12;

- закройте окно схемы данных, ответив *ДА* на вопрос о сохранении макета.

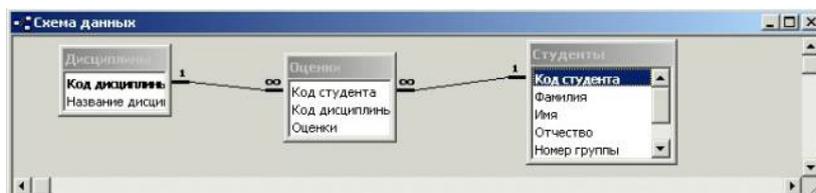


Рис. 4.12. Структура таблицы *Студенты*

Задание 2.7. Создание форм для ввода данных в таблицы

1. Создайте форму *Студенты*.

2. Заполните данными таблицу *Студенты* посредством формы *Студенты*.
3. Создайте форму *Дисциплины*.
4. Заполните данными таблицу *Дисциплины* посредством формы *Дисциплины*.
5. Создайте форму *Оценки*.
6. Заполните данными таблицу *Оценки* посредством формы *Оценки*.

Технология работы

1. Для создания формы *Студенты*:
 - откройте вкладку *Формы*;
 - щелкните по кнопке <Создать>;
 - в открывающемся списке выберите таблицу *Студенты*;
 - выберите пункт Автоформа: ленточная;
 - щелкните по кнопке <ОК>. Форма для ввода данных создана.

Примечание. Если вас не удовлетворяет макет, вы можете перейти в режим конструктора и изменить макет, передвигая и изменяя размеры элементов — заголовков полей и ячеек для ввода данных. Достаточно щелкнуть по элементу — он выделяется прямоугольной рамкой, и вы можете изменять размеры и двигать элемент. Если вы хотите изменить другие параметры элемента, надо, по выделенному элементу щелкнуть правой клавишей мыши, и откроется окно свойств элемента. В силу ограниченности объема раздела описать все свойства нет возможности, но их можно изучить самостоятельно по справочной системе, а многие свойства понятны уже из своего названия.

2. Заполните данными, приведенными в табл. 4.6, таблицу *Студенты* посредством формы.

Таблица 4.6.

Код студента	Фамилия	Имя	Отчество	Номер группы	Телефон	Стипендия
1	Арбузов	Николай	Николаевич	151	260-15-63	Да
2	Кириши	Петр	Валерьевич	151	110-67-82	Да
3	Кривинский	Сергей	Николаевич	151	172-97-21	Нет
4	Крылова	Елена	Петровна	151	130-31-87	Да
5	Кудачий	Григорий	Викторович	151	269-53-75	Да
6	Патрикеев	Олег	Борисович	152	234-11-63	Нет
7	Перлов	Кириш	Николаевич	152	312-21-33	Нет
8	Соколова	Наталья	Петровна	152	166-87-24	Нет
9	Степанская	Ольга	Витальевна	152	293-43-77	Да
10	Тимофеев	Сергей	Трофимович	152	260-11-57	Да

Примечание. Переход между ячейками лучше выполнять клавишей <Tab>, либо мышью. Существуют и другие варианты перехода по строкам или полям с помощью различных клавиш и их комбинаций. Обычно их используют опытные пользователи, не любящие работать с мышью.

Закройте форму, задав ей имя *Студенты*.

3. Создайте форму Дисциплины аналогично п. 1.

4. Заполните данными, приведенными в табл. 4.7, таблицу Дисциплины посредством, формы и закройте форму, задав ей имя Дисциплины.

5. Создайте форму Дисциплины аналогично п. 1.

6. Заполните данными, приведенными в табл. 4.8, таблицу Оценки. Посредством формы закройте форму, задав ей имя Оценки.

Таблица 4.7.

Код дисциплины	Название дисциплины
1	Информатика
2	Математика
3	Физика
4	Экономика

Таблица 4.8.

Код студента	Код дисциплины	Оценки	Код студента	Код дисциплины	Оценки
1	1	4	6	1	5
1	2	5	6	2	4
1	3	4	6	3	5
1	4	4	6	4	4
2	1	5	7	1	4
2	2	5	7	2	3
2	3	4	7	3	4
2	4	4	7	4	3
3	1	3	8	1	3
3	2	5	8	2	5
3	3	4	8	3	5
3	4	3	8	4	4
4	1	4	9	1	4
4	2	4	9	2	4
4	3	5	9	3	4
4	4	4	9	4	4
5	1	5	10	1	5
5	2	5	10	2	5
5	3	5	10	3	5
5	4	5	10	4	5

2.4. Модель № 4. Формирование сложных запросов

В *перекрестном запросе* отображаются результаты статистических расчетов (такие, как суммы, количество записей, средние значения), выполненных по данным из одного поля таблицы. Эти результаты группируются по двум наборам данных, один из которых расположен в левом столбце таблицы, а второй — в верхней строке. Например, нам надо узнать средний стаж работы ассистентов, доцентов и профессоров на разных кафедрах (на основе таблицы *Преподаватели*). Перекрестный запрос позволит легко решить эту задачу, создав таблицу, в которой заголовками строк будут служить должности, заголовками столбцов — названия кафедр, а в ячейках будут рассчитаны средние значения стажа преподавателей.

Запрос на изменение — это запрос, который за одну операцию вносит изменения в несколько записей. Существует четыре

типа запросов на изменение; на удаление, обновление и добавление записей, а также на создание таблицы.

Запрос на удаление удаляет группу записей, удовлетворяющих заданным условиям, из одной или нескольких таблиц. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.

Запрос на обновление записей вносит общие изменения в группу записей одной или нескольких таблиц. Например, на 10 процентов увеличилась заработная плата ассистентов, Запрос на обновление позволит быстро внести эти изменения в таблицу *Преподаватели*.

Запрос на добавление добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Например, появилось несколько новых преподавателей, а также база данных, содержащая сведения о них. Чтобы не вводить все данные вручную, их можно добавить в таблицу *Преподаватели*.

Запрос на создание таблицы создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Например, на основе таблицы *Преподаватели* можно создать новую таблицу, содержащую данные только о профессорах.

Запрос SQL — это запрос, создаваемый при помощи инструкций SQL. Этот тип запросов довольно сложен для начинающих пользователей и используется обычно опытными пользователями, имеющими навыки программирования и общения с серверами баз данных. Из-за сложности и специфики рассмотреть запрос SQL в данной главе не будем.

Задание 2.8.

1. Разработайте запрос с параметрами о студентах заданной группы, в котором при вводе в окно параметров номера группы (в примере это 151 или 152) на экран должен выводиться состав этой группы.
2. Создайте запрос, в котором выводятся оценки студентов заданной группы по заданной дисциплине.
3. Создайте перекрестный запрос, в результате которого создастся выборка, отражающая средний балл по дисциплинам в группах.
4. Разработайте запрос на увеличение на 10% заработной платы тех преподавателей, кто получает менее 500 руб.

5. Создайте запрос на удаление отчисленных студентов.
6. Разработайте запрос на создание базы данных отличников.
7. Для всех созданных вами запросов разработайте формы.

Технология работы

1. Для создания запроса с параметрами о студентах заданной группы:

- откройте вкладку *Запросы*;
- щелкните по кнопке <Создать>;
- в появившемся окне выберите *Простой запрос* и щелкните по кнопке <ОК>;
- в появившемся окне в строке *Таблицы/запросы* выберите из списка таблицу *Студенты*;
- перенесите все поля из окна "Доступные поля" в окно "Выбранные поля";
- щелкните по кнопке <Далее>. Выводить надо все поля, поэтому еще раз щелкните по кнопке <Далее>;
- в появившемся окне введите имя запроса *Группа*;
- щелкните по кнопке <Готово>. На экране появится таблица с данными запроса. Но вам надо, чтобы при выполнении запроса выяснялся номер группы. Для этого перейдите в режим конструктора;
- в строке *Условия отбора* для поля "Номер группы" введите фразу (скобки то же вводить): *[Введите номер группы]*;
- выполните запрос, щелкнув по кнопке на панели инструментов, или выполните команду Запрос, Запуск;
- в появившемся окне введите *151* и щелкните по кнопке <ОК>. На экране появится таблица с данными о студентах 151-й группы;
- сохраните запрос и закройте таблицу запроса.

2. Для создания запроса, в котором выводятся оценки студентов заданной группы по заданной дисциплине:

- на вкладке *Запросы* щелкните по кнопке <Создать>;
- выберите *Простой запрос* и щелкните по кнопке <ОК>;
- выберите таблицу *Студенты* и перенесите поля "Фамилия", "Имя", "Отчество", "Номер группы" в окно "Выделенные поля" (выделяя нужное поле и щелкая по кнопке).

Внимание! В дальнейшем под фразой *В таблице ... выберите поле ...* будем понимать выбор таблицы, выбор поля и перенос его в окно "Выделенные поля".

- в таблице *Дисциплины* выберите поле "Название дисциплины";
- в таблице *Оценки* выберите поле "Оценки". Вы сформировали шесть полей запроса — они связаны между собой посредством схемы данных;
 - щелкните по кнопке <Далее>, затем в появившемся окне снова щелкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Оценки группы*, затем щелкните по ячейке *Изменение структуры запроса* (в ней должна появиться черная точка) — это позволит сразу перейти в режим конструктора;
 - щелкните по кнопке <Готово>;
 - в строке *Условия отбора* для поля "Номер группы" введите фразу: *[Введите номер группы]*,
 - в строке *Условия отбора* для поля "Название дисциплины" введите фразу: *[Введите название дисциплины]*;
 - выполните запрос;
 - в первом появившемся окне введите *152*, затем щелкните по кнопке <ОК>, во втором — введите *Информатика* и щелкните по кнопке <ОК>. На экране появится таблица со списком 152-й группы и оценками по информатике;
 - сохраните запрос и закройте таблицу запроса.

3. Создайте перекрестный запрос о среднем балле в группах по дисциплинам. Но такой запрос строится на основе одной таблицы или одного запроса, в связи с чем надо сначала сформировать запрос, в котором были бы поля "Номер группы", "Название дисциплины" и "Оценки". Для этого:

 - на вкладке *Запросы* щелкните по кнопке <Создать>;
 - выберите *Простой запрос* и щелкните по кнопке <ОК>;
 - выберите из таблицы *Студенты* поле "Номер группы";
 - выберите из таблицы *Дисциплины* поле "Название дисциплины";
 - выберите из таблицы *Оценки* поле "Оценки";
 - щелкните по кнопке <Далее>, затем в появившемся окне снова щелкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Дисциплины оценки группы*;
 - щелкните по кнопке <Готово>;

- сохраните запрос и закройте таблицу запроса. Теперь можно создавать перекрестный запрос. Для этого:
 - на вкладке *Запросы* щелкните по кнопке <Создать>;
 - выберите *Перекрестный запрос* и щелкните по кнопке <ОК>;
 - щелкните по ячейке *Запросы*, выберите *Дисциплины оценки группы* и щелкните по кнопке <Далее>;
 - выберите поле "Название дисциплины" и щелкните по кнопке <Далее>;
 - выберите поле "Номер группы" и щелкните по кнопке <Далее>;
 - выберите функцию AVG, т.е. среднее (она по умолчанию уже выделена), и щелкните по кнопке <Далее>;
 - введите название запроса *Средние оценки* и щелкните по кнопке <Готово>. Откроется таблица перекрестного запроса. Обратите внимание на то, что Access создает еще итоговое значение средних оценок по дисциплинам;
 - закройте таблицу запроса.
- 4. Для создания запроса на изменение заработной платы преподавателей:
 - на вкладке *Запросы* щелкните по кнопке <Создать>;
 - выберите *Простой запрос*;
 - в таблице *Преподаватели* выберите поле <Зарплата>;
 - щелкните по кнопке <Далее>, затем в появившемся окне снова щелкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Изменение зарплаты*;
 - щелкните по ячейке *Изменение структуры запроса*;
 - щелкните по кнопке <Готово>;
 - в строке *Условия отбора* введите <1>500;
 - откройте пункт меню *Запрос* и выберите *Обновление*;
 - в строке конструктора запроса *Обновление* в поле "Зарплата" введите: $[Зарплата] * 1,1$;
 - выполните запрос, подтвердив готовность на обновление данных;
 - закройте запрос, подтвердив его сохранение;
 - откройте форму *Преподаватели*;

- просмотрите изменение заработной платы у преподавателей, получающих меньше 500 р.;
 - закройте форму.
5. Для создания запроса на отчисление студента гр. 152 *Перлова Кирилла Николаевича*:
- на вкладке *Запросы* щелкните по кнопке <Создать>;
 - выберите *Простой запрос*;
 - в таблице *Студенты* выберите поля "Фамилия", "Имя", "Отчество", "Номер группы";
 - щелкните по кнопке <Далее>, затем в появившемся окне снова щелкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Отчисленные студенты*;
 - щелкните по ячейке *Изменение структуры запроса*;
 - щелкните по кнопке <Готово>;
 - в строке *Условия отбора* введите; в поле "Фамилия" — *Перлов*, в поле "Имя" — *Кирилл*, в поле "Отчество" — *Николаевич*, в поле "Номер группы" — *152*;
 - откройте пункт меню *Запрос* и выберите *Удаление*;
 - просмотрите удаляемую запись, щелкнув по кнопке или выполнив команду **Вид, Режим таблицы**; если отчисляемый студент выбран правильно, то перейдите в режим конструктора и выполните запрос. Если условия отбора сделаны неправильно, измените их;
 - закройте запрос;
 - откройте форму *Студенты* и удостоверьтесь в удалении записи о студенте *Перлове*;
 - закройте форму.
6. Для создания запроса на создание базы данных отличников:
- на вкладке *Запросы* щелкните по кнопке <Создать>;
 - выберите *Простой запрос*;
 - в таблице *Студенты* выберите поля "Фамилия", "Имя", "Отчество" и "Номер группы", а в таблице *Оценки* — поле "Оценки";
 - щелкните по кнопке <Далее>, затем в появившемся окне вновь щелкните по кнопке <Далее>;
 - в появившемся окне введите имя запроса *Отличники*;

- щелкните по ячейке *Изменение структуры запроса*;
- щелкните по кнопке <Готово>;

Примечание. Для создания этого запроса надо воспользоваться операцией группировки. Будем считать отличниками тех студентов, которые набрали за четыре экзамена 20 баллов. Операция группировки позволит просуммировать оценки студентов по всем экзаменационным дисциплинам.

- для выполнения групповых операции щелкните на панели инструментов по кнопке или выполните команду Вид, Групповые операции;

- в строке *Групповые операции* поля "Оценки" щелкните по ячейке *Групповые операции*. Откройте раскрывающийся список и выберите функцию SUM;

- в строке *Условия отбора* поля "Оценки" введите 20;
- просмотрите создаваемую базу, щелкнув по кнопке или выполнив команду **Вид, Режим таблицы**;
- перейдите в режим конструктора;
- выполните команду Запрос, Создание таблицы;
- введите имя таблицы *Студенты-отличники* и щелкните по кнопке <ОК>;
- подтвердите создание таблицы;
- закройте с сохранением запрос;
- откройте вкладку *Таблицы*;
- откройте таблицу *Студенты-отличники*. Удостоверьтесь в правильности создания таблицы. Закройте таблицу.

7. Для каждого из созданных запросов создайте форму (можно рекомендовать автоформу в столбец или ленточную автоформу) для удобного просмотра данных. При создании этих форм воспользуйтесь рекомендациями в работе 3.

2.5. Модель № 5. Создание сложных форм и отчетов

Кнопочное меню представляет собой форму, на которой расположены элементы управления — кнопки с поясняющими надписями. Щелчок по кнопке открывает соответствующую таблицу, запрос, форму или отчет. Меню — это очень удобный инструмент работы с базами данных, и он практически всегда присутствует в базах, созданных для предприятий или фирм.

Кнопочное меню можно создать вручную (в режиме конструктора) или воспользовавшись *диспетчером кнопочных форм*. Меню, созданные вручную, могут обладать большими возможностями и выглядеть более привлекательно, чем меню, созданные *диспетчером*. Однако для создания красивого меню вручную требуется достаточно много времени даже для опытного разработчика, а создать простое меню с помощью *диспетчера* можно за несколько минут, тем более, что впоследствии, в режиме конструктора можно дополнить и изменить это меню. Остановимся на варианте создания меню с помощью *диспетчера кнопочных форм*.

Задание 2.9. Создание сложных форм

1. Разработайте сложную форму, в которой с названиями дисциплин была бы связана подчиненная форма *Студенты* и подчиненная форма *Оценки студентов*.

2. Измените расположение элементов в форме в соответствии с рис. 4.12.

3. Вставьте в форму диаграмму, графически отражающую оценки студентов.

4. Отредактируйте вид осей диаграммы.

Технология работы

1. Для создания сложной формы;

- на вкладке *Формы* щелкните по кнопке <Создать>;
- выберите *Мастер форм* и, не выбирая таблицу или запрос, щелкните по кнопке <ОК>;
- в *таблице Дисциплины* выберите поле "Название дисциплины";
- в *таблице Студенты* выберите поля "Код студента", "Фамилия", "Имя", "Отчество", "Номер группы";
- в *таблице Оценки* выберите поле "Оценки" и щелкните по кнопке <Далее>;
- в появившемся окне вариант построения формы нас удовлетворяет, поэтому щелкните по кнопке <Далее>;
- оставьте табличный вариант подчиненной формы и щелкните по кнопке <Далее>;
- выберите нужный вам стиль оформления формы и щелкните по кнопке <Далее>;

- введите название *формы Дисциплины и оценки*;
- щелкните по кнопке <Готово> и просмотрите полученную форму.

2. Нам не удовлетворяет расположение полей на экране. Измените их в соответствии с рис. 4.13, оставив место для диаграммы.

Студент	Фамилия	Имя	Отчество	Номер г	Оце
1	Арбузов	Николай	Николаевич	151	4

Рис. 4.13. Форма Дисциплины и оценки

Для этого:

- перейдите в режим конструктора;
- стандартными средствами Windows (технология *drag-and-drop*) измените размеры подчиненной формы так, чтобы были видны все данные. Для этого надо (как правило, многократно) переключаться из режима конструктора в режим формы, смотреть на полученный результат и, если он не подходит, снова корректировать в режиме конструктора. Ширину столбцов в подчиненной форме можно изменить только в режиме формы.

3. Для того чтобы вставить в форму диаграмму оценок студентов по заданным дисциплинам, необходимо:

- переключиться в режим конструктора;
- выполнить команду Вид, Панель элементов;
- на этой панели щелкнуть по кнопке <Aa>;
- создать прямоугольник для надписи — заголовка диаграммы. Для этого переведите курсор в левый верхний угол будущего прямоугольника, нажмите левую кнопку мыши и, не отпуская ее, доведите до правого нижнего угла, затем отпустите кнопку;
- ввести надпись *Диаграмма оценок*;
- выполнить команду Вставка, Диаграмма;

- на свободном месте формы растянуть прямоугольник для диаграммы (нажмите левую кнопку мыши в левом верхнем углу и, не отпуская ее, растяните прямоугольник до правого нижнего угла, затем отпустите кнопку);
 - выбрать таблицу *Оценки* и щелкнуть по кнопке <Далее>;
 - выбрать поля "Код студента" и "Оценки";
 - щелкнуть по кнопке <Далее>;
 - выбрать вид диаграммы *Гистограмма* (по умолчанию он и стоит) и щелкнуть по кнопке <Далее>;
 - дважды щелкнуть по надписи *Сумма_оценки*, выбрать *Отсутствует* и щелкнуть по кнопке <ОК>;
 - щелкнуть по кнопке <Далее>;
 - вновь щелкнуть по кнопке <Далее>, так как в строке *Поля формы* и в строке *Поля диаграммы* по умолчанию находится *Код дисциплины* (что нам и нужно);
 - стереть название диаграммы *Оценки* (так как мы уже задали надпись для диаграммы) и щелкнуть по кнопке <Далее>.
- 4. Отредактируйте вид осей диаграммы. Для этого:
 - дважды щелкните по диаграмме;
 - дважды щелкните по значениям вертикальной оси;
 - выберите вкладку *Шкала*;
 - уберите "галочку" у надписи *Минимальное значение*, а в ячейке справа от этого названия введите 1;
 - уберите "галочку" у надписи *Максимальное значение*, а в ячейке справа от этого названия введите 5;
 - уберите "галочку" у надписи *Цена основных делений*, а в ячейке справа от этого названия введите 1 и щелкните по кнопке <ОК>;
 - расширьте область диаграммы, перетащив правую границу окна диаграммы несколько правее (подведя курсор к правой границе до появления двойной стрелки и нажав левую кнопку мыши);
 - откройте окно "Microsoft Graph", выбрав в меню Файл пункт *Выход и возврат в дисциплины и оценки: форма*;
 - перейдите в режим формы (она представлена на рис. 4.14) и просмотрите форму для разных дисциплин (щелкая по кнопке перехода к следующей записи в нижней части формы). Вы увидите изменение названий дисциплин, а также оценок студентов

по этим дисциплинам и изменение диаграмм, отображающих эти оценки;

- закройте форму.

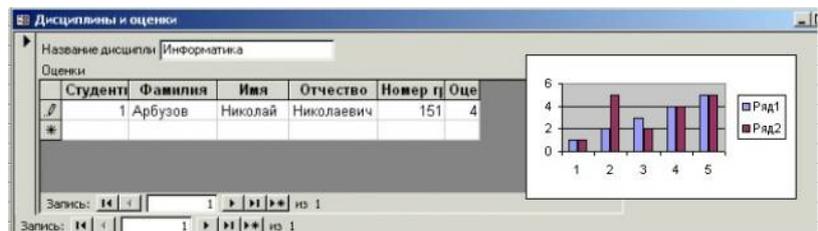


Рис. 4.14. Форма *Дисциплины и оценки* с включенной в нее диаграммой

Задание 2.10. Создание сложных отчетов

1. Создайте запрос, на основе которого будет формироваться отчет. В запросе должны присутствовать: из таблицы *Студенты* — поля "Фамилия", "Имя", "Отчество" и "Номер группы", из таблицы *Дисциплины* — поле "Название дисциплины", из таблицы *Оценки* — поле "Оценки".

2. Создайте отчет по итогам сессии. В отчете оценки студентов должны быть сгруппированы по номерам групп и дисциплинам. Для каждого студента должна вычисляться средняя оценка в сессию, а для каждой группы — среднее значение оценок по всем предметам.

Технология работы

1. Для создания запроса:

- на вкладке *Запросы* щелкните по кнопке <Создать>;
- выберите *Простой запрос* и щелкните по кнопке <ОК>;
- из таблицы *Студенты* выберите поля "Фамилия", "Имя", "Отчество" и "Номер группы", из таблицы *Дисциплины* — поле "Название дисциплины", из таблицы *Оценки* — поле "Оценки" и щелкните по кнопке <Далее>;
- щелкните еще раз по кнопке <Далее>;
- введите название запроса *Сессия* и щелкните по кнопке <Готово>;

• закройте запрос.

2. Для создания итогового отчета выполните следующее:

- на вкладке *Отчеты* щелкните по кнопке <Создать>;

- выберите *Мастер отчетов*, из раскрывающегося списка — запрос *Сессия* и щелкните по кнопке <ОК>;
- выберите все поля запроса и щелкните по кнопке <Далее>;
- тип представления данных нас удовлетворяет, поэтому щелкните по кнопке <Далее>;
- добавьте уровень группировки по номеру группы, выбрав в левом окне *Номер группы* и перенеся его в правое окно, щелкнув по кнопке;
- щелкните по кнопке <Далее>;
- щелкните по кнопке <Итоги>, так как надо вычислять средний балл;
- поставьте "галочку" в ячейке поля "AVG" (эта функция вычисляет среднее) и щелкните по кнопке <ОК>;
- щелкните по кнопке <ОК>, так как сортировка не требуется, потому что данными являются название дисциплины и оценки, порядок которых не столь важен;
- выберите макет отчета. Рекомендуем ступенчатый, так как он занимает меньше места и в нем наглядно представлены данные (хотя это дело вкуса). Щелкните по кнопке <Далее>; выберите стиль отчета и щелкните по кнопке <Далее>;
- введите название отчета *Итоги сессии* и щелкните по кнопке <Готово>. На экране появится отчет. Его можно просмотреть, изменяя масштаб (щелкнув по листу) и перелистывая страницы (в нижней части экрана). Его можно также распечатать, выполнив команду Файл, Печать. После завершения необходимых вам операций закройте окно просмотра отчета.

Задание 2.11. Разработайте кнопочную форму-меню для работы с базами данных, в которой должны быть созданные вами формы и отчет

Технология работы

Для создания кнопочного меню выполните следующие действия:

- выполните команду Сервис, Настройки, Диспетчер кнопочных форм;
- подтвердите создание кнопочной формы, щелкнув по кнопке <Да>;

- Access предложит вам работать с главной кнопочной формой или создать дополнительно новую. Создайте свою форму, щелкнув по кнопке <Создать>;
- введите имя *Меню* и щелкните по кнопке <ОК>;
- в окне выберите *Меню* и щелкните по кнопке <Изменить>;
- создайте элементы данной кнопочной формы, щелкнув по кнопке <Создать>;
- в строке *Текст* введите поясняющую надпись к первой создаваемой кнопке *Преподаватели*;
- в строке *Команда* выберите из списка *Открытие формы* в режиме редактирования;

Примечание. Диспетчер напрямую может связать кнопку с открытием формы или отчета. Чтобы открыть таблицу или запрос, надо создать соответствующий макрос и указать это в диспетчере.

- в строке *Форма* выберите из списка форму *Преподаватели* и щелкните по кнопке <ОК>; введите в меню все созданные формы и отчет, повторяя п. 6—9;
- закройте окно кнопочной формы, щелкнув по кнопке <Закреть>;
- щелкните по кнопке <По умолчанию>;
- закройте диспетчер кнопочных форм, щелкнув по кнопке <Закреть>;
- на вкладке *Формы* подведите курсор мыши к надписи *Кнопочная форма*, щелкните правой кнопкой мыши, выберите пункт *Переименовать* и введите новое имя *Форма меню*, затем нажмите клавишу <Enter>;
- откройте эту форму и просмотрите возможности открытия форм и отчета из меню.

Примечание. Для возврата из любой открытой формы или отчета в меню достаточно закрыть их.

2.6. Модель № 6. Создание базы данных

Задание 2.12.

1. Создать БД с именем.
2. В этой БД создать структуру таблицы 1 в Режиме конструктора согласно варианту. Ввод данных осуществить в режиме *Формы-Автоформа* в столбец.

3. В этой БД создать таблицу 2, осуществив связи между двумя созданными таблицами для конструирования Запроса-таблицы 3, объединяющей необходимые поля из первых двух таблиц. Создать вычисляемое поле.

4. Оформить работу.

Таблица 1.

Номенклатура-ценник

Артикул	Наименование материала	Единица измерения	Цена
12345	Сатин	м	44,15
12346	Габардин	м	345,00
12347	Крепдешин	м	288,70
12343	Драп	м	645,78
12300	Велюр	м	900,00
12401	Ситец	м	65,56
12307	Штапель	м	150,50
12301	Шифон	м	500,90
12340	Вельвет	м	120,00
12346	Шелк	м	155,45

Таблица 2.

Отпущено в производство

Номер ателье	Артикул	Количество
1	12345	1050
3	12346	1000
4	12347	2500
2	12343	7000
1	12300	2000
5	12401	2750
4	12307	2850
3	12301	3050
2	12340	4500
1	12346	5750

Таблица 3.

Ведомость учета поступления тканей

Но- мер ате- лье	Ар- ти- кул	Наиме- нование материа- ла	Единица измере- ния	Цена	Коли- чество	Сумма
---------------------------	-------------------	-------------------------------------	---------------------------	------	-----------------	-------

*Сумма = Цена * Количество*

Задание 2.13.

1. Создать БД с именем.
2. В этой БД создать структуру таблицы 1 в Режиме конструктора согласно варианту. Ввод данных осуществить в режиме Формы-Автоформа в столбец.
3. В этой БД создать таблицу 2, осуществив связи между двумя созданными таблицами для конструирования Запроса-таблицы 3, объединяющей необходимые поля из первых двух таблиц. Создать вычисляемое поле.
4. Оформить работу.

Таблица 1.

Справочник подетальных норм расхода материалов

Код детали	Наименование детали	Единица измерения	Норма расхода
1725	Поршень	шт	120
1870	Кольца	кг	2780
1900	Гильза	т	157,80
1727	Втулка	ц	130,70
1802	Граната	м	290,90
1500	Колодка	м	100,70
1700	Шайба	т	575,00
1950	Амортизатор	шт	800,00
1726	Пружина	кг	905,00
1877	Сальник	ц	455,00

Таблица 2.

Расход материалов на производство деталей

Код детали	Количество
1725	2,80
1870	10,40
1900	640,80
1727	451,10
1802	188,90
1500	974,80
1800	200,00
1850	38,80
1778	18,00
1870	165,00

Таблица 3.

Ведомость потребностей в материалах на выпуск деталей

Код детали	Наименование детали	Единица измерения	Количество	Норма расхода	Потребность
------------	---------------------	-------------------	------------	---------------	-------------

*Потребность = Норма расхода * Количество*

Задание 2.14.

1. Создать БД с именем.
2. В этой БД создать структуру таблицы 1 в Режиме конструктора согласно варианту. Ввод данных осуществить в режиме Формы-Автоформа в столбец.
3. В этой БД создать таблицу 2, осуществив связи между двумя созданными таблицами для конструирования Запроса-таблицы 3, объединяющей необходимые поля из первых двух таблиц. Создать вычисляемое поле.
4. Оформить работу.

Таблица 1.

Справочник материалов

Код материала	Наименование материала	Единица измерения	Цена
111367	Трубы д20	м	700
111385	Кирпич к-30	шт	900
124536	Трубы д30	м	900
124815	Плитка	шт	800
124857	Бетон	т	950
176101	Кирпич к-40	шт	750
176198	Цемент	т	500
202196	Вагонка №3	шт	950
115367	Доски	м	800
124816	Кафель	шт	700

Таблица 2.

Сведения о наличии материалов на складах

Номер склада	Код материала	Количество
1	111367	3500
2	111385	5400
3	124536	3600
4	124815	16000
5	124857	1900
6	176101	6000
7	176198	1500
8	202196	1900
9	115367	12000
10	124816	6000

Таблица 3.

Ведомость о наличии материалов на складах

№ склада	Код материала	Наименование материала	Единица измерения	Цена	Количество	Количество
----------	---------------	------------------------	-------------------	------	------------	------------

*Сумма = Цена * Количество*

Задание 2.15.

1. Создать БД с именем.
2. В этой БД создать структуру таблицы 1 в Режиме конструктора согласно варианту. Ввод данных осуществить в режиме Формы-Автоформа в столбец.
3. В этой БД создать таблицу 2, осуществив связи между двумя созданными таблицами для конструирования Запроса-таблицы 3, объединяющей необходимые поля из первых двух таблиц. Создать вычисляемое поле.
4. Оформить работу.

Таблица 1.

Ведомость наличия денежного довольствия

Код подразделения	Ф И О	Оклад по должности
1111	Антонов В.В.	10000,00
1112	Иванов О.Б.	10000,50
1113	Нигай Р.А.	7000,00
1114	Сидоров Е.А.	4800,89
1115	Ахметова Д.К.	5600,00
1116	Оспанов М.Р.	15000,00
1117	Досжанов М.Г.	15000,00
1118	Нам Н.Н.	4800,89
1119	Петров А.А.	2000,90
1120	Сидоров Р.Н.	2450,90

Таблица 2.

Ведомость удержаний с денежного довольствия

Код подразделения	Ф И О	Удержание за питание	Прочие удержания
1111	Антонов В.В.	350	100,90
1112	Иванов О.Б.	500	200
1113	Нигай Р.А.	300	100
1114	Сидоров Е.А.	200	45
1115	Ахметова Д.К.	355	100
1116	Оспанов М.Р.	750	250
1117	Досжанов М.Г.	750	250

1118	Нам Н.Н.	200	50
1119	Петров А.А	100	20
1120	Сидоров Р.Н.	100	20

Таблица 3.

Платежная ведомость

Код подразделения	Ф И О	Оклад	Сумма удержаний	Сумма к выдаче	Роспись
-------------------	-------	-------	-----------------	----------------	---------

Сумма удержаний = Удержание за питание + Прочие удержания

Сумма к выдаче = Оклад-Сумма удержаний

Задание 2.16.

1. Создать БД с именем.
2. В этой БД создать структуру таблицы 1 в Режиме конструктора согласно варианту. Ввод данных осуществить в режиме Формы-Автоформа в столбец.
3. В этой БД создать таблицу 2, осуществив связи между двумя созданными таблицами для конструирования Запроса-таблицы 3, объединяющей необходимые поля из первых двух таблиц. Создать вычисляемое поле.
4. Оформить работу.

Таблица 1.

Справочник товаров

Артикул	Наименование товара	Код группы товаров	Единица измерения	Цена
4201	Макароны	1	Кг	15,00
4202	Соломка	2	Кг	20,00
4203	Рожки	3	Кг	40,00
4204	Вермишель	4	Кг	40,00
4205	Лапша	5	Кг	45,00
4206	Сахар-песок	6	Кг	50,00
4207	Сахарная пудра	7	Шт	25,00

4208	Кофе натураль- ный	8	Шт	120,00
4209	Кофе с цикорием	9	Шт	135,00
4210	Чай	10	Шт	50,00

Таблица 2.

Сведения о товарах

Код группы товаров	Наименование группы товаров	Количество
1	А	250
1	В	100
1	Б	150
1	Е	250
1	Г	150
2	Д	270
2	А	250
3	Б	100
5	Г	100
4	Д	50

Таблица 3.

Ведомость о наличии товаров

Наименование группы товаров	Код группы товаров	Арти- кул	На- имено- вание товара	Сум- ма	Цена	Коли- чество
-----------------------------------	--------------------------	--------------	----------------------------------	------------	------	-----------------

$$\text{Сумма} = \text{Цена} * \text{Количество}$$

3. Связь информационных сетей

3.1. Internet — глобальная компьютерная сеть

Internet — глобальная компьютерная сеть, охватывающая весь мир. Internet состоит из множества локальных и глобальных сетей, принадлежащих различным компаниям и предприятиям, связанных между собой различными линиями связи. Internet можно представить себе в виде мозаики, сложенной из небольших сетей разной величины, которые активно взаимодействуют одна с другой, пересылая файлы, сообщения и т.п.

Кроме того, Internet предоставляет уникальные возможности дешевой, надежной и конфиденциальной глобальной связи по всему миру. Это оказывается очень удобным для фирм имеющих свои филиалы по всему миру, транснациональных корпораций и структур управления. Обычно, использование инфраструктуры Internet для международной связи обходится значительно дешевле прямой компьютерной связи через спутниковый канал или через телефон.

Электронная почта — самая распространенная услуга сети Internet. Посылка письма по электронной почте обходится значительно дешевле посылки обычного письма. Кроме того сообщение, посланное по электронной почте дойдет до адресата за несколько часов, в то время как обычное письмо может добраться до адресата несколько дней, а то и недель.

3.2. Протоколы сети Internet

Основное, что отличает Internet от других сетей — это его протоколы — TCP/IP. Термин TCP/IP означает все, что связано с протоколами взаимодействия между компьютерами в Internet. Он охватывает целое семейство протоколов, прикладные программы, и даже саму сеть. TCP/IP — это технология межсетевое взаимодействия, технология internet. Сеть, которая использует технологию internet, называется "internet". Если речь идет о глобальной сети, объединяющей множество сетей с технологией internet, то ее называют Internet.

Название протокол TCP/IP получил от двух коммуникационных протоколов (или протоколов связи). Это Transmission Control Protocol (TCP) и Internet Protocol (IP). Несмотря на то,

что в сети Internet используется большое число других протоколов, сеть Internet часто называют TCP/IP-сетью, так как эти два протокола, являются важнейшими.

В Internet существует 7 уровней взаимодействия между компьютерами: физический, логический, сетевой, транспортный, уровень сеансов связи, представительский и прикладной уровень. Каждому уровню взаимодействия соответствует набор протоколов (т.е. правил взаимодействия).

Протоколы физического уровня определяют вид и характеристики линий связи между компьютерами. В Internet используются практически все известные в настоящее время способы связи: от простого провода (витая пара) до волоконно-оптических линий связи (ВОЛС).

Для каждого типа линий связи разработан соответствующий протокол логического уровня, занимающийся управлением передачей информации по каналу. К протоколам логического уровня для телефонных линий относятся протоколы SLIP (Serial Line Interface Protocol) и PPP (Point to Point Protocol). Для связи по кабелю локальной сети — это пакетные драйверы плат ЛВС.

Протоколы сетевого уровня отвечают за передачу данных между устройствами в разных сетях, то есть занимаются маршрутизацией пакетов в сети. К протоколам сетевого уровня принадлежат IP (Internet Protocol) и ARP (Address Resolution Protocol).

Протоколы транспортного уровня управляют передачей данных из одной программы в другую. К протоколам транспортного уровня принадлежат TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

Протоколы уровня сеансов связи отвечают за установку, поддержание и уничтожение соответствующих каналов. В Internet этим занимаются уже упомянутые TCP и UDP протоколы, а также протокол UUCP (Unix to Unix Copy Protocol).

Протоколы представительского уровня занимаются обслуживанием прикладных программ. К программам представительского уровня принадлежат программы, запускаемые, к примеру, на Unix-сервере, для предоставления различных услуг абонентам. К таким программам относятся: telnet-сервер, FTP-сервер, Gopher-сервер, NFS-сервер, NNTP (Net News Transfer Protocol),

SMTP (Simple Mail Transfer Protocol), POP2 и POP3 (Post Office Protocol) и т.д.

К протоколам прикладного уровня относятся сетевые услуги и программы их предоставления.

3.3. Услуги предоставляемые сетью

Все услуги предоставляемые сетью Internet можно делить на две категории: обмен информацией между абонентами сети и использование баз данных сети.

К числу услуг связи между абонентами принадлежат:

Telnet — удаленный доступ. Дает возможность абоненту работать на любой ЭВМ сети Internet, как на своей собственной, то есть запускать программы, менять режим работы и т.д.

FTP (File Transfer Protocol) — протокол передачи файлов. Дает возможность абоненту обмениваться двоичными и текстовыми файлами с любым компьютером сети. Установив связь с удаленным компьютером, пользователь может скопировать файл с удаленного компьютера на свой или скопировать файл со своего компьютера на удаленный.

NFS (Network File System) — распределенная файловая система. Дает возможность абоненту пользоваться файловой системой удаленного компьютера, как своей собственной.

Электронная почта — обмен почтовыми сообщениями с любым абонентом сети Internet. Существует возможность отправки как текстовых, так и двоичных файлов. На размер почтового сообщения в сети Internet накладывается следующее ограничение — размер почтового сообщения не должен превышать 64 килобайт.

Новости — получение сетевых новостей и электронных досок объявлений сети и возможность помещения информации на доски объявлений сети. Электронные доски объявлений сети Internet формируются по тематике. Пользователь может по своему выбору подписаться на любые группы новостей.

Rsh (Remote Shell) — удаленный доступ. Аналог Telnet, но работает только в том случае, если на удаленном компьютере стоит ОС UNIX.

Rexec (Remote Execution) — выполнение одной команды на удаленной UNIX-машине.

Lpr — сетевая печать. Отправка файла на печать на удаленном (сетевом) принтере.

Lpq — сетевая печать. Показывает файлы, стоящие в очереди на печать, на сетевом принтере.

Ping — проверка доступности удаленной ЭВМ по сети.

Talk — дает возможность открытия "разговора" с пользователем удаленной ЭВМ. При этом на экране одновременно виден вводимый текст и ответ удаленного пользователя.

Iptunnel — дает возможность доступа к серверу ЛВС NetWare с которым нет непосредственной связи по ЛВС, а имеется лишь связь по сети Internet.

Whois — адресная книга сети Internet. По запросу абонент может получить информацию о принадлежности удаленного компьютера, о пользователях.

Finger — получение информации о пользователях удаленного компьютера.

Кроме этих услуг, сеть Internet предоставляет также следующие специфические услуги.

Webster — сетевая версия толкового словаря английского языка.

Факс-сервис — дает возможность пользователю отправлять сообщения по факсимильной связи, пользуясь факс-сервером сети.

Электронный переводчик — производит перевод присланного на него текста с одного языка на другой. Обращение к электронным переводчикам происходит посредством электронной почты.

Шлюзы — дают возможность абоненту отправлять сообщения в сети, не работающие с протоколами TCP/IP (FidoNet, Goldnet, AT50).

К системам автоматизированного поиска информации в сети Internet принадлежат следующие системы.

Gopher — наиболее широко распространенное средство поиска информации в сети Internet, позволяющее находить информацию по ключевым словам и фразам. Работа с системой Gopher напоминает просмотр оглавления, при этом пользователю предлагается пройти сквозь ряд вложенных меню и выбрать нужную тему.

Gopher позволяет получить информацию без указания имен и адресов авторов, пользователь не тратит много времени. Он просто сообщит системе Gopher, что именно ему нужно, и система находит соответствующие данные.

WAIS — еще более мощное средство получения информации, чем Gopher, поскольку оно осуществляет поиск ключевых слов во всех текстах документов. Запросы посылаются в WAIS на упрощенном английском языке. При работе с WAIS пользователям не нужно тратить много времени, чтобы найти необходимые им материалы.

WWW — система для работы с гипертекстом. Она является наиболее мощным средством поиска. Гипертекст соединяет различные документы на основе заранее заданного набора слов. Например, когда в тексте встречается новое слово или понятие, система, работающая с гипертекстом, дает возможность перейти к другому документу, в котором это слово или понятие рассматривается более подробно.

WWW часто используется в качестве интерфейса к базам данных WAIS, но отсутствие гипертекстовых связей ограничивает возможности WWW до простого просмотра.

Практически все услуги сети построены на принципе клиент-сервер. Сервером в сети Internet называется компьютер, способный предоставлять клиентам (по мере прихода от них запросов) некоторые сетевые услуги. Взаимодействие клиент-сервер строится обычно следующим образом. По приходу запросов от клиентов сервер запускает различные программы предоставления сетевых услуг. По мере выполнения запущенных программ сервер отвечает на запросы клиентов.

Все программное обеспечение сети также можно делить на клиентское и серверное. При этом, программное обеспечение сервера занимается предоставлением сетевых услуг, а клиентское программное обеспечение обеспечивает передачу запросов серверу и получение ответов от него.

3.4. Гипертекстовая технология WWW, URL, HTML

World Wide Web переводится на русский язык как “Всемирная Паутина”. WWW является одним из самых совершенных инструментов для работы в глобальной мировой сети Internet.

Главное отличие WWW от остальных инструментов для работы с Internet заключается в том, что WWW позволяет работать практически со всеми доступными на компьютере видами документов: это могут быть текстовые файлы, иллюстрации, звуковые и видео-ролики, и т.д.

Что такое WWW? Это попытка организовать всю информацию в Internet, и любую локальную информацию по выбору, как набор гипертекстовых документов. Можно перемещаться по сети, переходя от одного документа к другому по ссылкам. Все эти документы написаны на специально разработанном для этого языке, который называется HyperText Markup Language (HTML). Он напоминает язык, использующийся для написания текстовых документов, только HTML проще. Причем, можно использовать не только информацию, предоставляемую Internet, но и создавать собственные документы.

Польза гипертекста в создании гипертекстовых документов. Также в одном документе возможно делать ссылки на другие, написанные другими авторами или расположенные на другом сервере. Это представляется как одно целое.

Гипермедиа — это надмножество гипертекста. В гипермедиа производятся операции не только над текстом, но и над звуком, изображениями, анимацией.

Существуют WWW-серверы для Unix, Macintosh, MS Windows и VMS, большинство из них распространяются свободно. Установив WWW-сервер, можно решить две задачи:

1. Предоставить информацию внешним потребителям — сведения о фирме, каталоги продуктов и услуг, техническую или научную информацию.

2. Предоставить сотрудникам удобный доступ к внутренним информационным ресурсам организации. Это могут быть последние распоряжения руководства, внутренний телефонный справочник, ответы на часто задаваемые вопросы для пользователей прикладных систем, техническая документация. Информация, которую нужно предоставить пользователям WWW, оформляется в виде файлов на языке HTML. HTML — простой язык разметки, который позволяет помечать фрагменты текста и задавать ссылки на другие документы, выделять заголовки нескольких уровней, разбивать текст на абзацы, центрировать их и

т. п., превращая простой текст в отформатированный гипермедиальный документ. Легко создать html-файл вручную, однако, имеются специализированные редакторы и преобразователи файлов из других форматов.

Для просмотра документов используются специальные просмотрщики, такие как Mosaic, Netscape, Internet Explorer, lynx, www и другие. Mosaic и Netscape удобно использовать на графических терминалах.

Архитектура WWW-технологии

Архитектура взаимодействия программного обеспечения в системе World Wide Web. WWW построена по известной схеме “клиент-сервер”. На схеме 1.2 показано, как разделены функции в этой схеме. Программа-клиент выполняет функции интерфейса пользователя и обеспечивает доступ практически ко всем информационным ресурсам Internet. Клиент — это интерпретатор HTML. Клиент в зависимости от команд (разметки) выполняет различные функции.

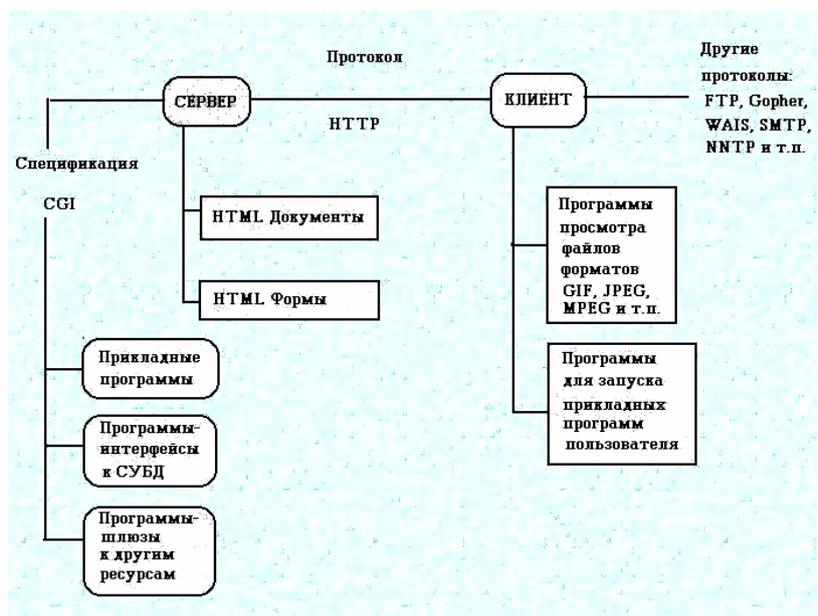


Схема 1.2. Структура “клиент-сервер”.

При анализе URL-спецификации или по командам сервера клиент запускает дополнительные внешние программы для работы с документами в форматах, отличных от HTML, например GIF, JPEG, MPEG, Postscript и т. п. Другую часть программного комплекса WWW составляет сервер протокола HTTP, базы данных документов в формате HTML, управляемые сервером, и программное обеспечение, разработанное в стандарте спецификации CGI. До образования Netscape использовались два HTTP-сервера: сервер CERN и сервер NCSA. Но в настоящее время число базовых серверов расширилось. Появился сервер для MS-Windows и Ararchie-сервер для Unix-платформ. Существуют и другие, но два последних можно выделить по доступности использования.

База данных HTML-документов — это часть файловой системы, которая содержит текстовые файлы в формате HTML и связанные с ними графику и другие ресурсы. Прикладное программное обеспечение, работающее с сервером, можно разделить на программы-шлюзы и прочие. Шлюзы — это программы, обеспечивающие взаимодействие сервера с серверами других протоколов, например ftp, или с распределенными на сети серверами Oracle. Прочие программы — это программы, принимающие данные от сервера и выполняющие какие-либо действия: получение текущей даты, реализацию графических ссылок, доступ к локальным базам данных или просто расчеты.

Имея доступ в Internet, можно создать свой WWW-сервер, или посмотреть информацию с других серверов.

Основные компоненты технологии World Wide Web

HTML — пример решения проблемы построения гипертекстовой системы при помощи специального средства управления отображением.

Гипертекстовые системы имеют специальные программные средства построения гипертекстовых связей. Сами гипертекстовые ссылки хранятся в специальных форматах или составляют специальные файлы. В HTML гипертекстовые ссылки встроены в тело документа и хранятся как его часть. В системах применяют специальные форматы хранения данных для повышения эффективности доступа. В WWW документы — это обычные ASCII-файлы, которые можно подготовить в любом текстовом редакторе.

В качестве базы, для разработки языка гипертекстовой разметки, был выбран SGML (Standard Generalised Markup Language). В настоящее время HTML развивается в сторону создания стандартного языка разработки интерфейсов как локальных, так и распределенных систем.

Вторым — WWW стала универсальная форма адресации информационных ресурсов. Universal Resource Identification (URI) представляет собой стройную систему, учитывающую опыт адресации и идентификации e-mail, Gopher, WAIS, telnet, ftp и т. п. URL используется в гипертекстовых ссылках и обеспечивает доступ к распределенным ресурсам сети. В URL можно адресовать как другие гипертекстовые документы формата HTML, так и ресурсы e-mail, telnet, ftp, Gopher, WAIS. Различные интерфейсные программы по разному осуществляют доступ к этим ресурсам.

Третьим в списке стоит протокол обмена данными в World Wide Web-HyperText Transfer Protocol. Данный протокол предназначен для обмена гипертекстовыми документами и учитывает специфику обмена. Управление в HTTP реализовано в виде ASCII-команд.

Последняя составляющая технологии WWW — это работы группы NCSA — спецификация Common Gateway Interface. CGI была специально разработана для расширения возможностей WWW за счет подключения всевозможного внешнего программного обеспечения. При реализации CGI заняли методы доступа, описанные в HTTP.

4. Защита информации в глобальной сети Internet

Проблемы защиты информации

Internet и информационная безопасность несовместны по самой природе Internet. Она как чисто корпоративная сеть. В настоящее время, с помощью единого стека протоколов TCP/IP и адресного пространства объединяют не только корпоративные и ведомственные сети (образовательные, государственные, коммерческие, военные и т.д.), являющиеся сетями с ограниченным доступом, но и рядовых пользователей, которые имеют возможность получить прямой доступ в Internet с домашних компьютеров с помощью модемов и телефонной сети общего пользования.

Чем проще доступ в Сеть, тем хуже ее информационная безопасность, так как пользователь может не узнать, что у него были скопированы — файлы и программы, есть возможности их порчи и корректировки.

Платой за пользование Internet является всеобщее снижение информационной безопасности, поэтому для предотвращения несанкционированного доступа к своим компьютерам все корпоративные и ведомственные сети, а также предприятия, использующие технологию Internet, ставят фильтры (fire-wall) между внутренней сетью и Internet, что фактически означает выход из единого адресного пространства. Еще большую безопасность даст отход от протокола TCP/IP и доступ в Internet через шлюзы.

Для решения этих и других вопросов при переходе к новой архитектуре Internet нужно предусмотреть следующее:

Во-первых, ликвидировать физическую связь между будущей Internet (которая превратится во Всемирную информационную сеть общего пользования) и корпоративными и ведомственными сетями, сохранив между ними лишь информационную связь через систему World Wide Web.

Во-вторых, заменить маршрутизаторы на коммутаторы, исключив обработку в узлах IP-протокола и заменив его на режим трансляции кадров Ethernet, при котором процесс коммутации сводится к простой операции сравнения MAC-адресов.

В-третьих, перейти в новое единое адресное пространство на базе физических адресов доступа к среде передачи (MAC-уровень), привязанное к географическому расположению сети.

Безопасность данных является одной из главных проблем в Internet. В банковской сфере проблема безопасности информации осложняется двумя факторами: во-первых, почти все ценности, с которыми имеет дело банк (кроме наличных денег и еще кое-чего), существуют лишь в виде той или иной информации. Во-вторых, банк не может существовать без связей с внешним миром: без клиентов, корреспондентов и т. п.

4.1. Информационная безопасность и информационные технологии

На раннем этапе автоматизации внедрение банковских систем (и вообще средств автоматизации банковской деятельности) не повышало открытость банка.

Положение изменилось, как на рынке финансовых услуг стали появляться продукты, возникновение которых было немыслимо без информационных технологий. В первую очередь это — пластиковые карточки. Обслуживание по карточкам шло в режиме голосовой авторизации, открытость информационной системы банка повышалась незначительно, но затем появились банкоматы, POS-терминалы, другие устройства самообслуживания — то есть средства, принадлежащие к информационной системе банка, но расположенные вне ее и доступные посторонним для банка лицам.

Повысившаяся открытость системы потребовала специальных мер для контроля и регулирования обмена информацией: дополнительных средств идентификации и аутентификации лиц, которые запрашивают доступ к системе (PIN-код, информация о клиенте на магнитной полосе или в памяти микросхемы карточки, шифрование данных, контрольные числа и другие средства защиты карточек), средств криптозащиты информации в каналах связи и т. д.

Большой сдвиг баланса “защищенность-открытость” связан с телекоммуникациями. Системы электронных расчетов между банками защитить несложно, так как субъектами электронного обмена информацией выступают сами банки.

Общая тенденция развития телекоммуникаций и массового распространения вычислительной техники привела к тому, что на рынке банковских услуг во всем мире появились новые, чис-

то телекоммуникационные продукты, и в первую очередь системы Home Banking (отечественный аналог — “клиент-банк”). Это потребовало обеспечить клиентам круглосуточный доступ к автоматизированной банковской системе для проведения операций, причем на совершение банковских транзакций получил непосредственно клиент. Степень открытости информационной системы банка возросла до предела. Требуется особые, специальные меры для того, чтобы не упала ее защищенность.

Эпоха “информационной супермагистрали”: взрывообразное развитие сети Internet и связанных услуг. Вместе с новыми возможностями сеть принесла и новые опасности. Каким образом клиент связывается с банком: по коммутируемой линии, приходящей на модемный пул банковского узла связи, или по IP-протоколу через Internet? Однако в первом случае, максимально возможное количество подключений ограничивается техническими характеристиками модемного пула, во втором же — возможностями Internet, которые могут быть выше. Кроме того, сетевой адрес банка общедоступен, тогда как телефонные номера модемного пула могут сообщаться лишь заинтересованным лицам. Значит, открытость банка, чья информационная система связана с Internet, значительно выше, чем в первом случае.

Все это вызывает необходимость пересмотра подходов к обеспечению информационной безопасности банка. Подключаясь к Internet, следует заново провести анализ риска и составить план защиты информационной системы, а также конкретный план ликвидации последствий, возникающих в случае тех или иных нарушений конфиденциальности, сохранности и доступности информации.

Вторая проблема: во многих банках безопасность автоматизированной банковской системы не анализируется и не обеспечивается. Мало имеется необходимого набора организационных документов (анализ риска, план защиты и план ликвидации последствий).

Поэтому, необходимо заботиться об информационной безопасности.

Некоторые рекомендации:

1. Необходим комплексный подход к информационной безопасности. Информационная безопасность должна рассмат-

ваться как составная часть общей безопасности банка — причем как важная и неотъемлемая ее часть. Разработка концепции информационной безопасности должна обязательно проходить при участии управления безопасности банка. В этой концепции следует предусматривать не только меры, связанные с информационными технологиями (криптозащиту, программные средства администрирования прав пользователей, их идентификации и аутентификации, “брандмауэры” для защиты входов — выходов сети и т. п.), но и меры административного и технического характера, включая жесткие процедуры контроля физического доступа к автоматизированной банковской системе, а также средства синхронизации и обмена данными между модулем администрирования безопасности банковской системы и системой охраны.

2. Необходимо участие сотрудников управления безопасности на этапе выбора — приобретения — разработки автоматизированной банковской системы. Управление безопасности должно контролировать наличие надлежащих средств разграничения доступа к информации в приобретаемой системе, безопасность в сети Internet.

4.2. Средства защиты информации

Одним из наиболее распространенных механизмов защиты является применение межсетевых экранов — **брандмауэров (firewalls)**.

Технология работы в глобальных сетях Solstice FireWall-1

Разработано множество средств для обеспечения информационной безопасности, предназначенных для использования на различных компьютерах с разными ОС. Одно из направлений межсетевые экраны (firewalls), призванные контролировать доступ к информации со стороны пользователей внешних сетей.

Здесь рассматриваются основные понятия экранирующих систем, а также требования, предъявляемые к ним. На примере пакета Solstice FireWall-1 рассматриваются несколько типичных случаев использования таких систем, особенно применительно к вопросам обеспечения безопасности Internet-подключений. Рассмотрено также несколько уникальных особенностей Solstice FireWall-1, позволяющих оценить его лидерство.

Назначение экранирующих систем и требования к ним

Межсетевое экранирование формулируется следующим образом. Пусть имеются две информационные системы или два множества информационных систем. Экран (firewall) — это средство разграничения доступа клиентов из одного множества систем к информации, хранящейся на серверах в другом множестве.

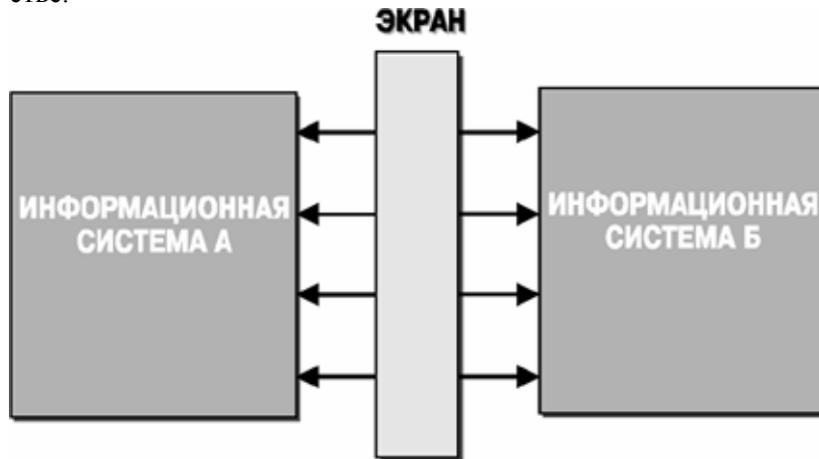


Рис. 2.2.1. Экран FireWall.

Экран выполняет свои функции, контролируя все информационные потоки между этими двумя множествами информационных систем, работая как некоторая “информационная мембрана”. Здесь экран можно представлять как набор фильтров, анализирующих проходящую через них информацию и, на основе заложенных в них алгоритмов, принимающих решение: пропустить ли эту информацию или отказать в ее пересылке. Кроме того, такая система может выполнять регистрацию событий, связанных с процессами разграничения доступа, в частности, фиксировать все “незаконные” попытки доступа к информации и, дополнительно, сигнализировать о ситуациях, требующих немедленной реакции.

Экранирующие системы делают несимметричными. Для экранов определяются понятия “внутри” и “снаружи”, и задача

экрана состоит в защите внутренней сети от “потенциально враждебного” окружения. Примером потенциально враждебной внешней сети является Internet.

Какие проблемы возникают при построении экранирующих систем? При этом нужно рассматривать не только проблему безопасного подключения к Internet, но и разграничение доступа внутри корпоративной сети организации.

Первое. Требование к таким системам, это обеспечение безопасности внутренней (защищаемой) сети и полный контроль над внешними подключениями и сеансами связи.

Во-вторых, экранирующая система должна обладать мощными и гибкими средствами управления для простого и полного воплощения.

В-третьих, экранирующая система должна работать незаметно для пользователей локальной сети и не затруднять выполнение ими легальных действий.

В-четвертых, экранирующая система должна работать достаточно эффективно и успевать обрабатывать весь входящий и исходящий трафик в “пиковых” режимах. Это необходимо для того, чтобы firewall нельзя было, “забросать” большим количеством вызовов, которые привели бы к нарушению ее работы.

Пятое. Система обеспечения безопасности должна быть сама надежно защищена от любых несанкционированных воздействий, поскольку она является ключом к конфиденциальной информации в организации.

Шестое. Если у организации имеется несколько внешних подключений, в том числе и в удаленных филиалах, система управления экранами должна иметь возможность централизованно обеспечивать для них проведение единой безопасности.

Седьмое. Система Firewall должна иметь средства авторизации доступа пользователей через внешние подключения. Например, когда часть персонала организации должна выезжать в командировки, и в процессе работы требуется доступ, по крайней мере, к некоторым ресурсам внутренней компьютерной сети организации. Система должна уметь распознавать таких пользователей и предоставлять им необходимый доступ к информации.

Структура системы Solstice FireWall-1

Примером, является программный комплекс Solstice FireWall-1. Рассмотрим основные компоненты Solstice FireWall-1 и функции, которые они реализуют (рис. 2.2.2).

Центральным для системы FireWall-1 является модуль управления всем комплексом. С этим модулем работает администратор безопасности сети.

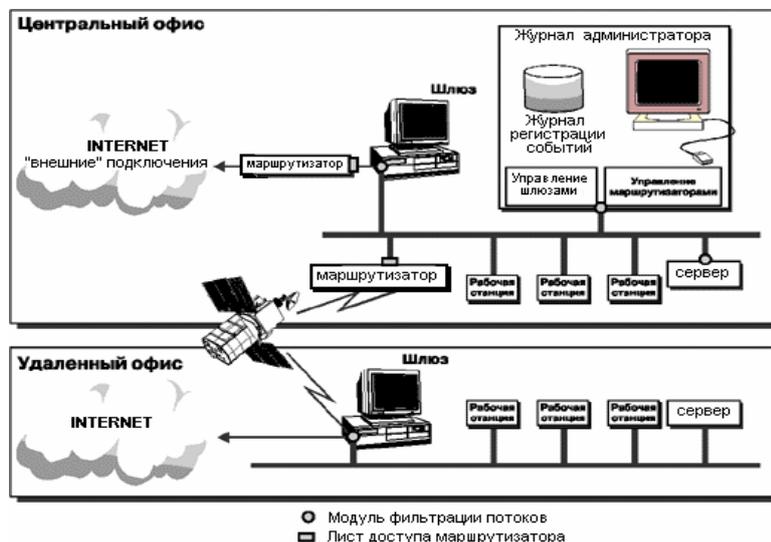


Рис. 2.2.2. Основные компоненты Solstice FireWall-1.

Администратору безопасности сети для конфигурирования комплекса FireWall-1 необходимо выполнить следующий ряд действий:

- Определить объекты, участвующие в процессе обработки информации. Имеются в виду пользователи и группы пользователей, компьютеры и их группы, маршрутизаторы и различные подсети локальной сети организации.
- Описать сетевые протоколы и сервисы, с которыми будут работать приложения.
- С помощью введенных понятий описывается политика разграничения доступа в следующих терминах: "Группе пользователей А разрешен доступ к ресурсу Б с помощью сервиса или протокола С, но об этом сделать пометку в регистрационном

журнале”. Совокупность таких записей компилируется в исполнимую форму блоком управления и далее, передается на исполнение в модули фильтрации.

Модули фильтрации могут располагаться на компьютерах — шлюзах или выделенных серверах — или в маршрутизаторах, как часть конфигурационной информации. Модули фильтрации просматривают все пакеты, поступающие на сетевые интерфейсы и в зависимости от заданных правил, пропускают или отбрасывают эти пакеты, с соответствующей записью в регистрационном журнале. Эти модули, работая непосредственно с драйверами сетевых интерфейсов, обрабатывают весь поток данных, располагая полной информацией о передаваемых пакетах.

Пример реализации политики безопасности

Рассмотрим процесс практической реализации политики безопасности организации с помощью программного пакета FireWall-1. (рис. 2.2.3).

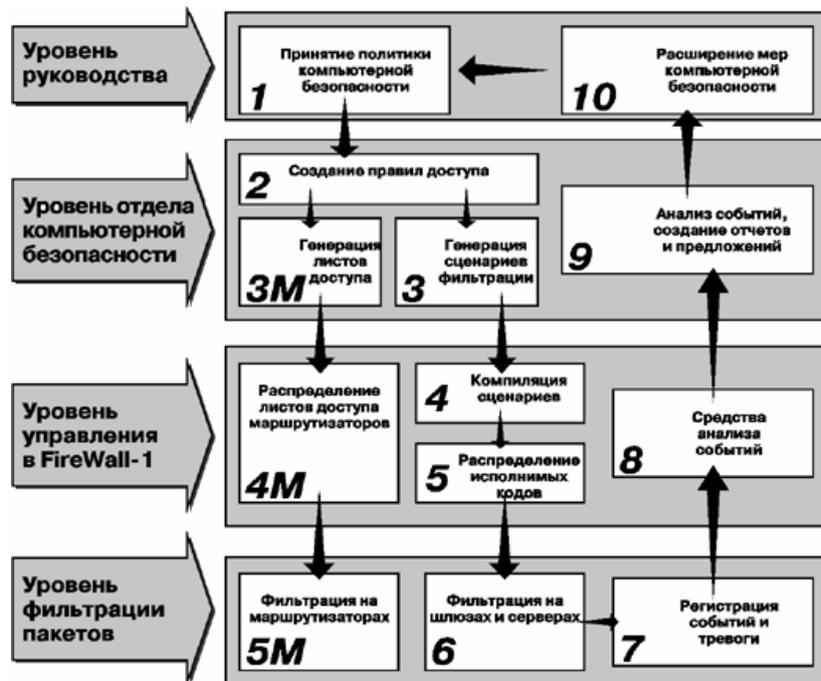


Рис. 2.2.3. Реализация политики безопасности FireWall.

1. Разрабатываются и утверждаются на уровне руководства организации правила политики безопасности.

2. После утверждения эти правила надо воплотить в жизнь. Для этого их нужно перевести в структуру типа “откуда, куда и каким способом доступ разрешен или, наоборот, запрещен. Такие структуры, легко переносятся в базы правил системы FireWall-1.

3. На основе этой базы правил формируются списки доступа для маршрутизаторов и сценарии работы фильтров на сетевых шлюзах. Списки и сценарии далее переносятся на физические компоненты сети, после чего правила политики безопасности “вступают в силу”.

4. В процессе работы фильтры пакетов на шлюзах и серверах генерируют записи обо всех событиях, которые им приказали отслеживать, а также, запускают механизмы “тревоги”, требующие от администратора немедленной реакции.

5. На основе анализа записей, сделанных системой, отдел компьютерной безопасности организации может разрабатывать предложения по изменению и дальнейшему развитию политики безопасности.

Рассмотрим простой пример реализации следующих правил:

1. Из локальных сетей подразделений, возможно удаленных, разрешается связь с любой локальной сетью организации после аутентификации, например, по UNIX-паролю.

2. Всем запрещается доступ к сети финансового департамента, за исключением генерального директора и директора этого департамента.

3. Из Internet разрешается только отправлять и получать почту. Обо всех других попытках связи необходимо делать подробную запись.

Все эти правила представляются средствами графического интерфейса Редактора Правил FireWall-1 (рис. 2.2.4).

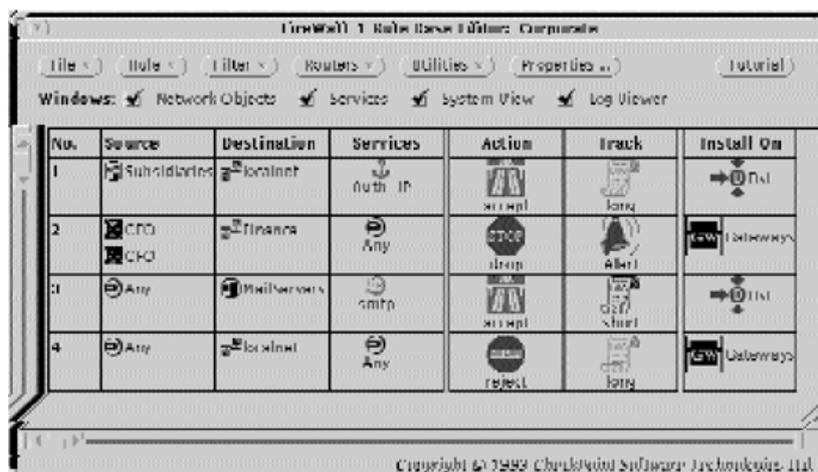


Рис. 2.2.4. Графический интерфейс Редактора Правил FireWall-1.

После загрузки правил FireWall-1 для каждого пакета, передаваемого по сети, последовательно просматривает список правил до нахождения элемента, соответствующего текущему случаю.

Важна защита системы, на которой размещен административно-конфигурационный модуль FireWall-1. Рекомендуется запретить средствами FireWall-1 все виды доступа к машине, или строго ограничить список пользователей, которым это разрешено, принять меры по физическому ограничению доступа и по защите обычными средствами ОС UNIX.

Управление системой FireWall-1

На рис. 2.2.5 показаны основные элементы управления системой FireWall-1.

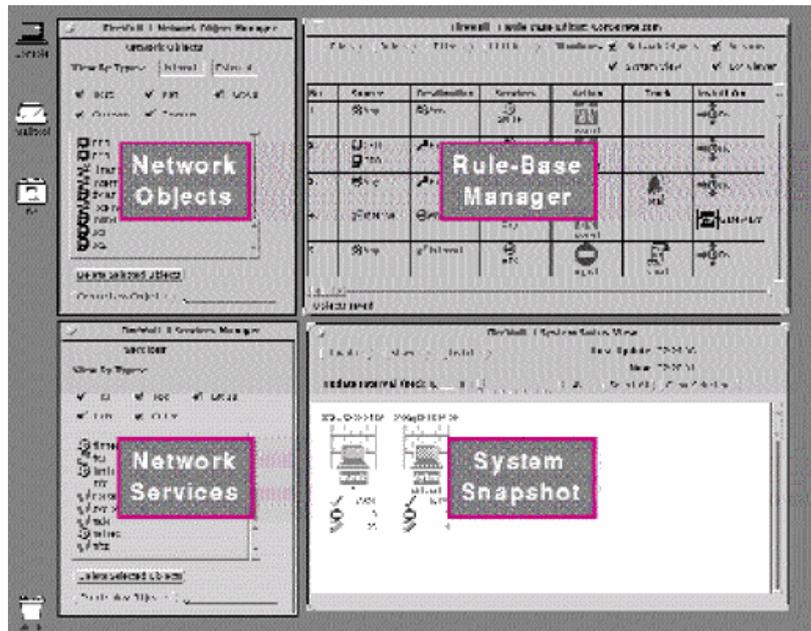


Рис. 2.2.5. Основные элементы управления системой FireWall-1.

Слева расположены редакторы баз данных об объектах, существующих в сети и о протоколах или сервисах, с помощью которых происходит обмен информацией. Справа вверху показан редактор правил доступа.

Справа внизу располагается интерфейс контроля текущего состояния системы, в котором для всех объектов, которые занес туда администратор, отображаются данные о количестве разрешенных коммуникаций (галочки), о количестве отвергнутых связей (знак “кирпич”) и о количестве коммуникаций с регистрацией (иконка карандаш). Кирпичная стена за символом объекта (компьютера) означает, что на нем установлен модуль фильтрации системы FireWall-1.

Пример реализации политики безопасности

Случай, когда первоначальная конфигурация сети меняется, а вместе с ней меняется и политика безопасности.

Установить в организации несколько общедоступных серверов для предоставления информационных услуг. Это могут быть, например, серверы World Wide Web, FTP или другие информационные серверы. Поскольку такие системы обособлены от работы всей остальной сети организации, для них часто выделяют свою собственную подсеть, имеющую выход в Internet через шлюз (рис. 2.2.6).

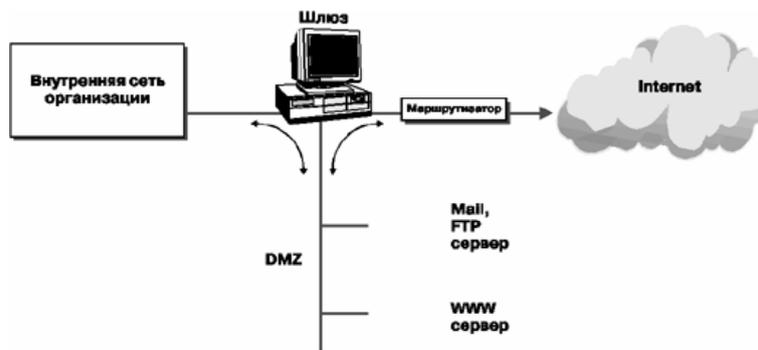


Рис. 2.2.6. Схема шлюза Internet.

В предыдущем примере локальная сеть была уже защищена, значит разрешен соответствующий доступ в выделенную подсеть. Это делается с помощью одной дополнительной строки в редакторе правил. Это является типичной при изменении конфигурации FireWall-1. Для этого требуется изменение одной или небольшого числа строк в наборе правил доступа.

Аутентификация пользователей при работе с FTP

Solstice FireWall-1 позволяет администратору установить различные режимы работы с интерактивными сервисами FTP и telnet для различных пользователей и групп пользователей. При установленном режиме аутентификации, FireWall-1 заменяет стандартные FTP и telnet домены UNIX на свои собственные,

располагая их на шлюзе, закрытом с помощью модулей фильтрации пакетов. Пользователь, желающий начать интерактивную сессию по FTP или telnet (это должен быть разрешенный пользователь и в разрешенное для него время), может сделать это только через вход на такой шлюз, где и выполняется вся процедура аутентификации. Она задается при описании пользователей или групп пользователей и может проводиться следующими способами:

- Unix-пароль;
- программа S/Key генерации одноразовых паролей;
- карточки SecurID с аппаратной генерацией одноразовых паролей.

Гибкие алгоритмы фильтрации UDP-пакетов, динамическое экранирование

UDP-протоколы, входящие в состав набора TCP/IP, представляют проблему для обеспечения безопасности. На их основе создано множество приложений. Они являются протоколами “без состояния”, что приводит к отсутствию различий между запросом и ответом, приходящим извне защищаемой сети.

Пакет FireWall-1 решает эту проблему созданием контекста соединений поверх UDP сессий, запоминая параметры запросов. Пропускаются назад только ответы внешних серверов на высланные запросы, которые однозначно отличаются от любых других UDP-пакетов (например: незаконных запросов), поскольку их параметры хранятся в памяти FireWall-1.

Можно отметить, что данная возможность присутствует в немногих программах экранирования, распространяемых в настоящий момент.

Подобные механизмы задействуются для приложений, использующих RPC, и для FTP сеансов. Возникают аналогичные проблемы, связанные с динамическим выделением портов для сеансов связи, которые FireWall-1 отслеживает аналогичным образом, запоминая необходимую информацию при запросах на такие сеансы и обеспечивая только “законный” обмен данными.

Данные возможности пакета Solstice FireWall-1 выделяют его среди всех остальных межсетевых экранов.

Язык программирования

Система Solstice FireWall-1 имеет собственный встроенный объектно-ориентированный язык программирования, применяемый для описания поведения модулей — Фильтров системы. Результатом работы графического интерфейса администратора системы является сгенерированный сценарий работы именно на этом внутреннем языке. Он не сложен для понимания, что допускает непосредственное программирование на нем. Однако на практике данная возможность почти не используется, поскольку графический интерфейс системы и так позволяет сделать практически все, что нужно.

Прозрачность и эффективность

FireWall-1 полностью прозрачен для пользователей. Еще одним свойством системы Solstice FireWall-1 является очень высокая скорость работы. Фактически модули системы работают на сетевых скоростях передачи информации, что обусловлено компиляцией сгенерированных сценариев работы перед подключением их непосредственно в процесс фильтрации.

Solstice FireWall-1 — эффективное средство защиты корпоративных сетей и их сегментов от внешних угроз, а также от несанкционированных взаимодействий локальных пользователей с внешними системами.

Solstice FireWall-1 обеспечивает высокоуровневую поддержку политики безопасности организации по отношению ко всем протоколам семейства TCP/IP.

Solstice FireWall-1 характеризуется прозрачностью для легальных пользователей и высокой эффективностью.

4.3. Ограничения доступа в WWW серверах

Рассмотрим два из них:

- Ограничить доступ по IP адресам клиентских машин;
- ввести идентификатор получателя с паролем для данного вида документов.

Такого рода ввод ограничений стал использоваться достаточно часто, т.к. многие стремятся в Internet, чтобы использовать его коммуникации для доставки информации потребителю. С помощью такого рода механизмов по разграничению прав доступа удобно производить саморассылку информации на получение которой существует договор.

Ограничения по IP адресам

Доступ к приватным документам можно разрешить, либо наоборот запретить используя IP адреса конкретных машин или сетей, например:

123.456.78.9

123.456.79.

В этом случае доступ будет разрешен (или запрещен в зависимости от контекста) для машины с IP адресом 123.456.78.9 и для всех машин подсетки 123.456.79.

Ограничения по идентификатору получателя

Доступ к приватным документам можно разрешить, либо наоборот запретить используя присвоенное имя и пароль конкретному пользователю, причем пароль нигде не хранится.

Пример: Агентство печати предоставляет свою продукцию, только своим подписчикам, которые заключили договор и оплатили подписку. WWW Сервер находится в сети Internet и общедоступен.

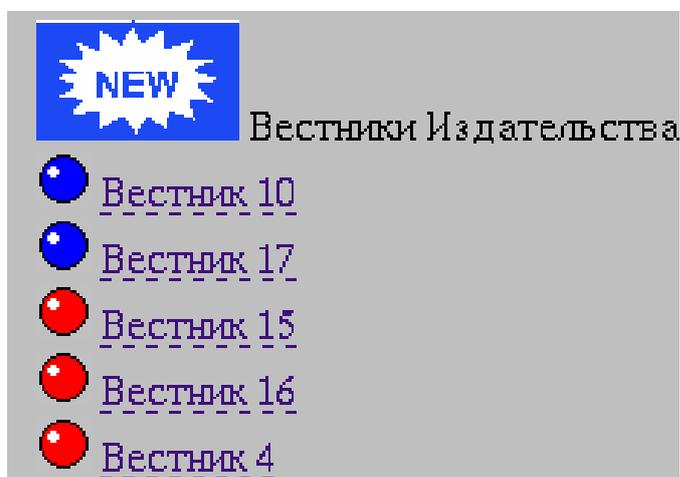


Рис. 2.2.7. Пример списка вестников издательства.

Выберем Вестник предоставляемый конкретному подписчику. На клиентском месте подписчик получает сообщение:

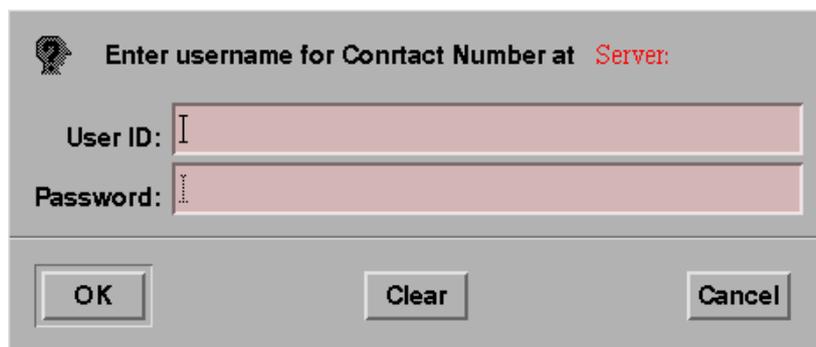


Рис. 2.2.8. Окно ввода пароля.

Если он правильно написал свое имя и пароль, то он допускается до документа, в противном случае — получает сообщение:

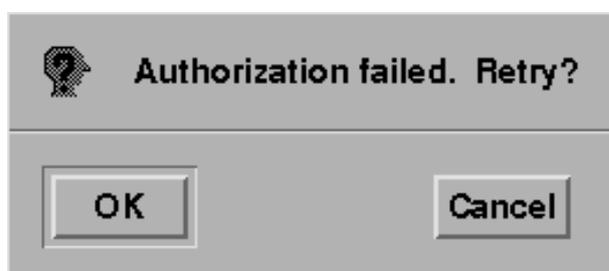


Рис. 2.2.9. Окно неправильного ввода пароля.

4.4. Информационная безопасность в Internet

Архитектура Internet подразумевает подключение к внешним открытым сетям, использование внешних сервисов и предоставление собственных сервисов вовне, что предъявляет повышенные требования к защите информации.

В Internet-системах используется подход клиент-сервер, а главная роль отводится Web-сервису. Web-серверы должны поддерживать традиционные защитные средства, такие как аутентификация и разграничение доступа; кроме того, необходимо

обеспечение новых свойств, в особенности безопасности программной среды и на серверной, и на клиентской сторонах.

Формирование режима информационной безопасности — проблема комплексная.

Меры по ее решению можно разделить на четыре уровня:

- законодательный (законы, нормативные акты, стандарты и т.п.);
- административный (действия общего характера, предпринимаемые руководством организации);
- процедурный (конкретные меры безопасности, имеющие дело с людьми);
- программно-технический (конкретные технические меры).

Законодательный уровень

В настоящее время наиболее подробным законодательным документом в области информационной безопасности является Уголовный кодекс.

Уголовный кодекс стоит на страже всех аспектов информационной безопасности — доступности, целостности, конфиденциальности, предусматривая наказания за “уничтожение, блокирование, модификацию и копирование информации, нарушение работы ЭВМ, системы ЭВМ или их сети”.

Разработка сетевых аспектов политики безопасности

Политика безопасности определяется как совокупность документированных управленческих решений, направленных на защиту информации и ассоциированных с ней ресурсов.

При разработке и проведении руководствоваться следующими принципами:

- невозможность миновать защитные средства;
- усиление самого слабого звена;
- невозможность перехода в небезопасное состояние;
- минимизация привилегий;
- разделение обязанностей;
- эшелонированность обороны;
- разнообразие защитных средств;
- простота и управляемость информационной системы;
- обеспечение всеобщей поддержки мер безопасности.

К межсетевым экранам данный принцип означает, что все информационные потоки в защищаемую сеть и из нее должны проходить через экран. Не должно быть “тайных” модемных входов или тестовых линий, идущих в обход экрана.

Надежность определяется слабым звеном. Слабым звеном оказывается не компьютер или программа, а человек, и тогда проблема обеспечения информационной безопасности приобретает нетехнический характер.

Принцип невозможности перехода в небезопасное состояние означает, что при любых обстоятельствах, в том числе нештатных, защитное средство либо полностью выполняет свои функции, либо полностью блокирует доступ.

Принцип минимизации привилегий предписывает выделять пользователям и администраторам только те права доступа, которые необходимы им для выполнения служебных обязанностей.

Принцип разделения обязанностей предполагает распределение ролей и ответственности, при котором один человек не может нарушить критически важный для организации процесс. Это важно, чтобы предотвратить злонамеренные или некомпетентные действия системного администратора.

Принцип эшелонированности обороны предписывает защитный рубеж. За средствами физической защиты должны следовать программно-технические средства, за идентификацией и аутентификацией — управление доступом и, как последний рубеж, — протоколирование и аудит. Эшелонированная оборона способна задержать злоумышленника, а наличие такого рубежа, как протоколирование и аудит, существенно затрудняет незаметное выполнение злоумышленных действий.

Принцип разнообразия защитных средств рекомендует организовывать различные по характеру оборонительные рубежи, чтобы от потенциального злоумышленника требовалось овладение разнообразными и, по возможности, несовместимыми между собой навыками (например умением преодолевать высокую ограду и знанием слабостей нескольких операционных систем).

Очень важен принцип простоты и управляемости информационной системы в целом и защитных средств в особенности. В простой и управляемой системе можно проверить согласован-

ность конфигурации разных компонентов и осуществить централизованное администрирование. Важно отметить интегрирующую роль Web-сервиса, скрывающего разнообразие обслуживаемых объектов и предоставляющего единый, наглядный интерфейс. Если объекты некоторого вида (например, таблицы базы данных) доступны через Web, необходимо заблокировать прямой доступ к ним, поскольку в противном случае система будет сложной и трудно управляемой.

Последний принцип — всеобщая поддержка мер безопасности — носит нетехнический характер. Анализ рисков — важнейший этап выработки политики безопасности. При оценке рисков, которым подвержены Internet-системы, нужно учитывать следующие обстоятельства:

- новые угрозы по отношению к старым сервисам, вытекающие из возможности пассивного или активного прослушивания сети. Пассивное прослушивание означает чтение сетевого трафика, а активное — его изменение (кражу, дублирование или модификацию передаваемых данных). Например, аутентификация удаленного клиента с помощью пароля многократного использования не может считаться надежной в сетевой среде, независимо от длины пароля;

- новые (сетевые) сервисы и ассоциированные с ними угрозы.

В Internet-системах следует придерживаться принципа “все, что не разрешено, запрещено”, поскольку “лишний” сетевой сервис может предоставить канал проникновения в корпоративную систему.

Процедурные меры

В целом Internet-технология не предъявляет каких-либо специфических требований к мерам процедурного уровня. Рассматриваются два обстоятельства:

- описание должностей, связанных с определением, наполнением и поддержанием корпоративной гипертекстовой структуры официальных документов;

- поддержка жизненного цикла информации, наполняющей Internet.

При описании должностей исходить из аналогии между Internet и издательством. В издательстве существует директор,

определяющий общую направленность деятельности. В Internet ему соответствует Web-администратор, решающий, какая корпоративная информация должна присутствовать на Web-сервере и как следует структурировать дерево (точнее, граф) HTML-документов.

В многопрофильных издательствах существуют редакции, занимающиеся конкретными направлениями (математические книги, книги для детей и т.п.). Аналогично, в Internet целесообразно выделить должность публикатора, ведающего появлением документов отдельных подразделений и определяющего перечень и характер публикаций.

У каждой книги есть титульный редактор, отвечающий перед издательством за свою работу. В Internet редакторы занимаются вставкой документов в корпоративное дерево, их коррекцией и удалением. В больших организациях “слой” редактор может состоять из нескольких уровней.

В издательстве, и в Internet должны быть авторы, создающие документы. Они не должны иметь прав на модификацию корпоративного дерева и отдельных документов. Их работа — передать свой труд редактору.

Кроме официальных, корпоративных, в Internet могут присутствовать групповые и личные документы, порядок работы с которыми (роли, права доступа) определяется, соответственно, групповыми и личными интересами.

К вопросам поддержки жизненного цикла Internet-информации, необходимо использование средств конфигурационного управления. Достоинство Intranet-технологии состоит в том, что основные операции конфигурационного управления — внесение изменений (создание новой версии) и извлечение старой версии документа — естественным образом вписываются в рамки Web-интерфейса.

Управление доступом путем фильтрации информации

Рассмотрим меры программно-технического уровня, направленные на обеспечение информационной безопасности систем, построенных в технологии Internet. На первое место среди таких мер, межсетевые экраны — средство разграничения доступа.

Бороться с угрозами, присущими сетевой среде, средствами универсальных операционных систем не представляется воз-

можным. Универсальная ОС — это огромная программа, содержащая, помимо явных ошибок, некоторые особенности, которые могут быть использованы для получения нелегальных привилегий. Современная технология программирования не позволяет сделать большие программы безопасными. Разработка специализированных защитных средств допускают формальную или неформальную верификацию. Межсетевой экран, является средством, допускающим декомпозицию, связанную с обслуживанием различных сетевых протоколов.

Межсетевой экран — это полупроницаемая мембрана, которая располагается между защищаемой (внутренней) сетью и внешней средой (внешними сетями или другими сегментами корпоративной сети) и контролирует все информационные потоки во внутреннюю сеть и из нее (Рис. 2.3.1). Контроль информационных потоков состоит в их фильтрации, с выполнением некоторых преобразований и извещением отправителя, что его данным в пропуске отказано. Фильтрация осуществляется на основе набора правил, предварительно загруженных в экран и являющихся выражением сетевых аспектов политики безопасности организации.

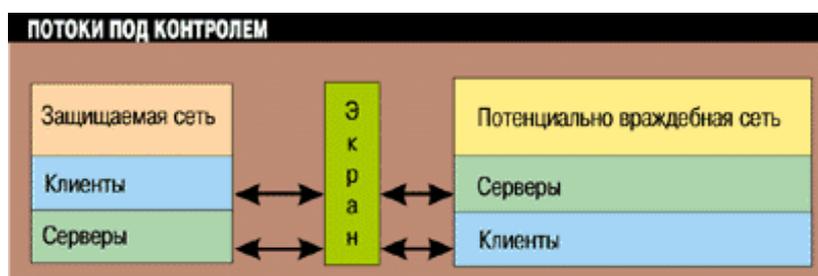


Рис. 2.3.1. Межсетевой экран как средство контроля информационных потоков

Разделить случаи, когда экран устанавливается на границе с внешней (обычно общедоступной) сетью или на границе между сегментами одной корпоративной сети.

При общении с внешними сетями используется семейство протоколов TCP/IP. Внешний межсетевой экран должен учиты-

вать специфику этих протоколов. Для внутренних экранов, следует принимать помимо TCP/IP протоколы SPX/IPX, применяемые в сетях Novell NetWare.

Когда корпоративная сеть содержит лишь один внешний канал, является исключением, чем правилом. Напротив, типична, при которой корпоративная сеть состоит из нескольких территориально разнесенных сегментов, каждый из которых подключен к сети общего пользования (Рис. 2.3.2). В этом случае каждое подключение должно защищаться своим экраном. Можно считать, что корпоративный внешний межсетевой экран является составным, и требуется решать задачу согласованного администрирования (управления и аудита) всех компонентов.

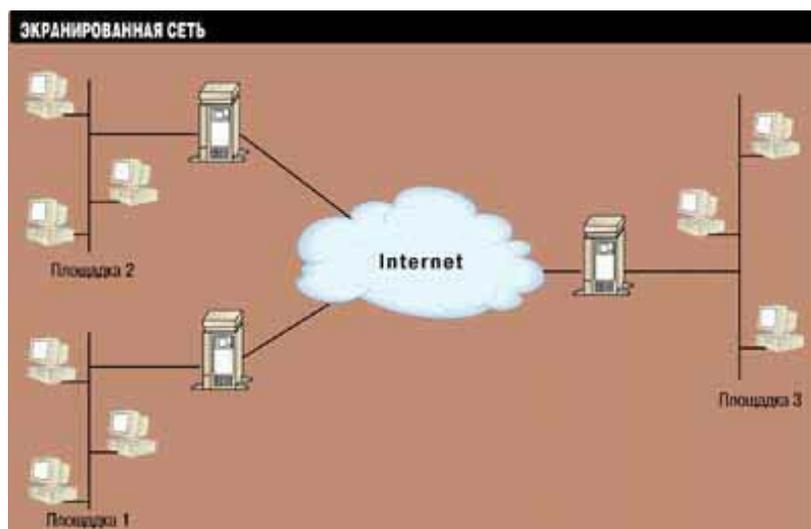


Рис. 2.3.2. Экранирование корпоративной сети, состоящей из нескольких территориально разнесенных сегментов, каждый из которых подключен к сети общего пользования

При рассмотрении любого вопроса, касающегося сетевых технологий, основой служит семиуровневая эталонная модель ISO/OSI. Межсетевые экраны также целесообразно классифицировать по тому, на каком уровне производится фильтрация —

канальном, сетевом, транспортном или прикладном. Соответственно, можно говорить об экранирующих концентраторах (уровень 2), маршрутизаторах (уровень 3), о транспортном экранировании (уровень 4) и о прикладных экранах (уровень 7). Существуют также комплексные экраны, анализирующие информацию на нескольких уровнях.

Здесь рассматриваются экранирующие концентраторы, они мало отличаются от экранирующих маршрутизаторов.

При принятии решения “пропустить/не пропустить”, межсетевые экраны могут использовать не только информацию, содержащуюся в фильтруемых потоках, но и данные, полученные из окружения, например текущее время.

Итак, возможности меж сетевого экрана непосредственно определяются тем, какая информация может использоваться в правилах фильтрации и какова может быть мощность наборов правил. Наиболее типичным является сочетание экранирующих маршрутизаторов и прикладного экрана (Рис. 2.3.3).

Приведенная конфигурация называется экранирующей подсетью. Как правило, сервисы, которые организация предоставляет для внешнего применения (например “представительский” Web-сервер), целесообразно выносить как раз в экранирующую подсеть.

Качество меж сетевого экрана определяется еще двумя характеристиками — простотой применения и собственной защищенностью. Для простоты использования первостепенное значение имеют наглядный интерфейс при задании правил фильтрации и возможность централизованного администрирования составных конфигураций.

Защищенность меж сетевого экрана обеспечивается средствами, что и защищенность универсальных систем. При выполнении централизованного администрирования следует о защите информации от пассивного и активного прослушивания сети, то есть обеспечить ее (информации) целостность и конфиденциальность.

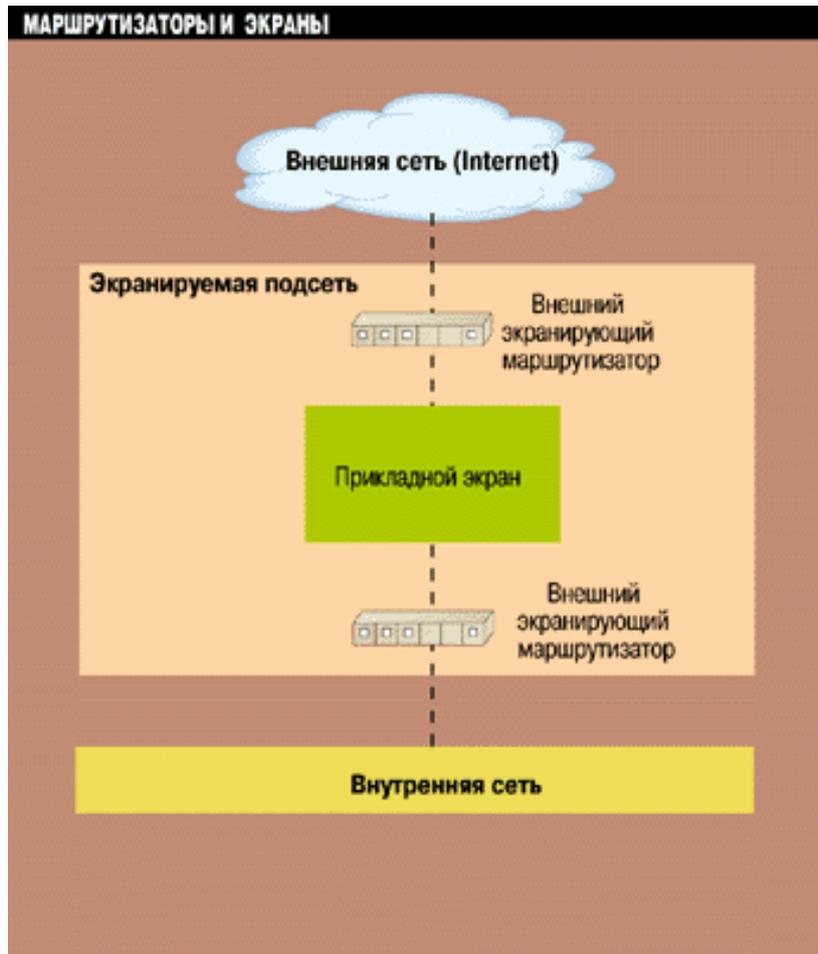


Рис. 2.3.3. Сочетание экранирующих маршрутизаторов и прикладного экрана

Экранирования (фильтрации), механизм безопасности. Помимо блокирования потоков данных, нарушающих политику безопасности, межсетевой экран может скрывать информацию о защищаемой сети, затрудняя действия злоумышленников. Прикладной экран осуществляет действия от субъектов внутренней

сети, в результате из внешней сети кажется, что имеет место взаимодействие с межсетевым экраном (Рис. 2.3.4). При таком подходе топология внутренней сети скрыта от внешних пользователей, поэтому задача злоумышленника усложняется.



Рис. 2.3.4. Истинные и кажущиеся информационные потоки

Скрытие информации о топологии защищаемой сети является трансляция “внутренних” сетевых адресов, которая попутно решает проблему расширения адресного пространства, выделенного организации.

Ограничивающий интерфейс можно рассматривать как разновидность экранирования. На невидимый объект трудно напасть, особенно с помощью фиксированного набора средств. Здесь Web-интерфейс обладает естественной защитой, в том случае, когда гипертекстовые документы формируются динамически.

Экранирующая роль Web-сервиса проявляется и тогда, когда этот сервис осуществляет посреднические (точнее, интегрирующие) функции при доступе к другим ресурсам, в частности таблицам базы данных. Здесь не только контролируются потоки запросов, но и скрывается реальная организация баз данных.

Безопасность программной среды

Перемещение программ на серверы. Пример реализации подхода — это хранимые процедуры в реляционных СУБД. Для Web-серверов аналогом хранимых процедур являются программы, обслуживающие общий шлюзовый интерфейс (Common Gateway Interface — CGI).

CGI-процедуры располагаются на серверах и используются для динамического порождения HTML-документов. Политика безопасности организации и процедурные меры должны определять, кто имеет право помещать на сервер CGI-процедуры. Контроль необходим, выполнение сервером некорректной программы может привести к тяжелым последствиям. Мера технического характера состоит в минимизации привилегий пользователя, от имени которого выполняется Web-сервер.

В технологии Internet, возникает вопрос в перемещении программ с Web-серверов на клиентские компьютеры — для создания анимации, выполнения семантического контроля при вводе данных и т.д. Активные агенты — неотъемлемая часть технологии Internet.

Перемещение программы по сети, представляют повышенную опасность, т.к. программа, полученная из ненадежного источника, может содержать внесенные ошибки или целенаправленно созданный зловредный код. Такая программа потенциально угрожает всем основным аспектам информационной безопасности:

- доступности (программа может поглотить все наличные ресурсы);
- целостности (программа может удалить или повредить данные);
- конфиденциальности (программа может прочитать данные и передать их по сети).

Система программирования Java впервые предложила концепцию решения.

Java предлагает три оборонительных рубежа:

- надежность языка;
- контроль при получении программ;
- контроль при выполнении программ.

Существует еще одно средство обеспечения информационной безопасности — беспрецедентная открытость Java-системы. Исходные тексты Java-компилятора и интерпретатора доступны для проверки, велика вероятность, что ошибки и недочеты первыми будут обнаруживать честные специалисты.

Java — это язык для написания клиентских частей приложений, одним из требований является мобильность, загруженная

программа может обслуживать только пользовательский интерфейс и осуществлять сетевое взаимодействие с сервером. Программа не может работать с файлами хотя бы потому, что на Java-терминале их, возможно, не будет.

Защита Web-серверов

В Web-серверах объектами доступа выступают универсальные локаторы ресурсов (URL — Uniform (Universal) Resource Locator). За этими локаторами могут стоять различные сущности — HTML-файлы, CGI-процедуры и т.п. Субъекты доступа идентифицируются по IP-адресам или именам компьютеров и областей управления. Кроме того, может использоваться парольная аутентификация пользователей или более сложные схемы, основанные на криптографических технологиях.

В большинстве Web-серверов права разграничиваются с точностью до каталогов (директорий) с применением произвольного управления доступом. Могут предоставляться права на чтение HTML-файлов, выполнение CGI-процедур и т.д.

Для раннего выявления попыток нелегального проникновения в Web-сервер важен регулярный анализ регистрационной информации.

Защита системы, на которой функционирует Web-сервер, должна следовать универсальным рекомендациям, главной из которых является максимальное упрощение. Все ненужные сервисы, файлы, устройства должны быть удалены. Число пользователей, имеющих прямой доступ к серверу, должно быть сведено к минимуму, а их привилегии — упорядочены в соответствии со служебными обязанностями.

Минимизировать объем информации о сервере, которую могут получить пользователи. Многие серверы в случае обращения по имени каталога и отсутствия файла index.HTML в нем, выдают HTML-вариант оглавления каталога. В этом оглавлении могут встретиться имена файлов с исходными текстами CGI-процедур или с иной конфиденциальной информацией.

Аутентификация в открытых сетях

Методы, применяемые в открытых сетях для подтверждения и проверки подлинности субъектов, должны быть устойчивы к пассивному и активному прослушиванию сети. Суть их сводится к следующему.

- Субъект демонстрирует знание секретного ключа, при этом ключ либо вообще не передается по сети, либо передается в зашифрованном виде.
- Субъект демонстрирует обладание программным или аппаратным средством генерации одноразовых паролей или средством, работающим в режиме “запрос-ответ”.
- Субъект демонстрирует подлинность своего местоположения, при этом используется система навигационных спутников.

Виртуальные частные сети

Одной из важнейших задач является защита потоков корпоративных данных, передаваемых по открытым сетям. Открытые каналы могут быть надежно защищены лишь одним методом — криптографическим.

Можно отметить, что выделенные линии не обладают особыми преимуществами перед линиями общего пользования в плане информационной безопасности. Выделенные линии частично будут располагаться в неконтролируемой зоне, где их могут повредить. Реальное достоинство — это гарантированная пропускная способность выделенных линий.

Представляется естественным возложить на межсетевой экран задачу шифрования и дешифрования корпоративного трафика на пути во внешнюю сеть и из нее. Что такое шифрование/дешифрование? Это начальное распределение ключей. Современные криптографические технологии предлагают для этого целый ряд методов.

После того, как межсетевые экраны осуществили криптографическое закрытие корпоративных потоков данных, территориальная разнесенность сегментов сети проявляется лишь в разной скорости обмена с разными сегментами.

Простота и однородность архитектуры

Важнейшим аспектом информационной безопасности является управляемость системы. Управляемость — это и поддержание высокой доступности системы.

Проблема управляемости встает на клиентских рабочих местах и на стыке клиентской и серверной частей информационной системы. Причина проста — клиентских мест гораздо больше,

чем серверных. Технология Internet за счет простоты и однородности архитектуры позволяет стоимость администрирования клиентского рабочего места нулевой. Замена и повторный ввод в эксплуатацию клиентского компьютера могут быть осуществлены быстро.

На стыке клиентской и серверной частей Intranet-системы находится Web-сервер. Это позволяет иметь единый механизм регистрации пользователей и наделения их правами доступа с последующим централизованным администрированием. Взаимодействие с многочисленными разнородными сервисами оказывается скрытым не только от пользователей, но и в значительной степени от системного администратора.

Задача информационной безопасности в Internet, это-однородность и простота архитектуры Internet.

4.5. Список специальных терминов

Английские термины

Alta Vista — один из мощнейших поисковых серверов Internet.

Archie — архив. Система для определения местонахождения файлов в публичных архивах сети Internet.

ARP (Address Resolution Protocol) — протокол определения адреса, преобразует адрес компьютера в сети Internet в его физический адрес.

ARPA (Advanced Research Projects Agency) — бюро проектов передовых исследований министерства обороны США.

ARPANET — экспериментальная сеть, работавшая в семидесятые годы, на которой проверялись теоретическая база и программное обеспечение, положенные в основу Internet. В настоящее время не существует.

Bps (bit per second) — бит в секунду. Единица измерения пропускной способности линии связи. Пропускная способность линии связи определяется количеством информации, передаваемой по линии за единицу времени.

Cisco — маршрутизатор, разработанный фирмой Cisco-Systems.

DNS (Domain Name System) — доменная система имен. распределенная система баз данных для перевода имен компьютеров в сети Internet в их IP-адреса.

Ethernet — тип локальной сети. Хороша разнообразием типов проводов для соединений, обеспечивающих пропускные способности от 2 до 10 миллионов bps(2-10 Mbps). Довольно часто компьютеры, использующие протоколы TCP/IP, через Ethernet подсоединяются к Internet.

FTP (File Transfer Protocol) — протокол передачи файлов, протокол, определяющий правила пересылки файлов с одного компьютера на другой.

FAQ (Frequently Asked Questions) — часто задаваемые вопросы. Раздел публичных архивов сети Internet в котором хранится информация для "начинающих" пользователей сетевой инфраструктуры.

Gopher — интерактивная оболочка для поиска, присоединения и использования ресурсов и возможностей Internet. Интерфейс с пользователем осуществлен через систему меню.

HTML (Hypertext Markup Language)- язык для написания гипертекстовых документов. Основная особенность — наличие гипертекстовых связей между документами находящимися в различных архивах сети; благодаря этим связям можно непосредственно во время просмотра одного документа переходить к другим документам.

Internet — глобальная компьютерная сеть.

internet — технология сетевого взаимодействия между компьютерами разных типов.

IP (Internet Protocol) — протокол межсетевого взаимодействия, самый важный из протоколов сети Internet, обеспечивает маршрутизацию пакетов в сети.

IP-адрес — уникальный 32-битный адрес каждого компьютера в сети Internet.

Iptunnel — одна из прикладных программ сети Internet. Дает возможность доступа к серверу ЛВС NetWare с которым нет непосредственной связи по ЛВС, а имеется лишь связь по сети Internet.

Lpr — сетевая печать. Команда отправки файла на печать на удаленном принтере.

Lpq — сетевая печать. Показывает файлы стоящие в очереди на печать.

NetBlazer — маршрутизатор, разработанный фирмой Telebit.

NetWare — сетевая операционная система, разработанная фирмой Novell; позволяет строить ЛВС основанную на принципе взаимодействия клиент-сервер. Взаимодействие между сервером и клиентом в ЛВС NetWare производится на основе собственных протоколов (IPX), тем не менее протоколы TCP/IP также поддерживаются.

NFS (Network File System) — распределенная файловая система. Предоставляет возможность использования файловой системы удаленного компьютера в качестве дополнительного НЖМД.

NNTP (Net News Transfer Protocol) — протокол передачи сетевых новостей. Обеспечивает получение сетевых новостей и электронных досок объявлений сети и возможность помещения информации на доски объявлений сети.

Ping — утилита проверки связи с удаленной ЭВМ.

POP (Post Office Protocol) — протокол "почтовый офис". Используется для обмена почтой между хостом и абонентами. Особенность протокола — обмен почтовыми сообщениями по запросу от абонента.

PPP (Point to Point Protocol) — протокол канального уровня, позволяющий использовать для выхода в Internet обычные модемные линии. Относительно новый протокол, является аналогом SLIP.

RAM (Random Access Memory) — оперативная память.

RFC (Requests For Comments) — запросы комментариев. Раздел публичных архивов сети Internet в котором хранится информация о всех стандартных протоколах сети Internet.

Rexec (Remote Execution) — выполнение одной команды на удаленной UNIX-машине.

Rsh (Remote Shell) — удаленный доступ. Аналог Telnet, но работает только в том случае, если на удаленном компьютере стоит ОС UNIX.

SLIP (Serial Line Internet Protocol) — протокол канального уровня позволяющий использовать для выхода в Internet обычные модемные линии.

SMTP (Simple Mail Transfer Protocol) — простой протокол передачи почты. Основная особенность протокола SMTP — обмен почтовыми сообщениями происходит не по запросу одного из хостов, а через определенное время (каждые 20 — 30 минут). Почта между хостами в Internet передается на основе протокола SMTP.

Talk — одна из прикладных программ сети Internet. Дает возможность открытия "разговора" с пользователем удаленной ЭВМ. При этом на экране одновременно печатается вводимый текст и ответ удаленного пользователя.

Telnet — удаленный доступ. Дает возможность абоненту работать на любой ЭВМ сети Internet как на своей собственной.

TCP/IP — под TCP/IP обычно понимается все множество протоколов поддерживаемых в сети Internet.

TCP (Transmission Control Protocol) — протокол контроля передачи информации в сети. TCP — протокол транспортного уровня, один из основных протоколов сети Internet. Отвечает за установление и поддержание виртуального канала (т.е. логического соединения), а также за безошибочную передачу информации по каналу.

UDP (User Datagram Protocol) — протокол транспортного уровня, в отличие от протокола TCP не обеспечивает безошибочной передачи пакета.

Unix — многозадачная операционная система, основная операционная среда в сети Internet. Имеет различные реализации: Unix-BSD, Unix-Ware, Unix-Interactive.

UUCP — протокол копирования информации с одного Unix-хоста на другой. UUCP — не входит в состав протоколов TCP/IP, но тем не менее все-еще широко используется в сети Internet. На основе протокола UUCP — построены многие системы обмена почтой, до сих пор используемые в сети.

VERONICA (Very Easy Rodent-Oriented Netwide Index to Computer Archives) — система поиска информации в публичных архивах сети Internet по ключевым словам.

WAIS (Wide Area Information Servers) — мощная система поиска информации в базах данных сети Internet по ключевым словам.

WWW (World Wide Web) — всемирная паутина. Система распределенных баз данных, обладающих гипертекстовыми связями между документами.

Whois — адресная книга сети Internet.

Webster — сетевая версия толкового словаря английского языка.

Русские термины

Драйвер — загружаемая в оперативную память программа, управляющая обменом данными между прикладными процессами и внешними устройствами.

Гипертекст — документ, имеющий связи с другими документами через систему выделенных слов (ссылок). Гипертекст соединяет различные документы на основе заранее заданного набора слов. Например, когда в тексте встречается новое слово или понятие, система, работающая с гипертекстом, дает возможность перейти к другому документу, в котором это слово или понятие рассматривается более подробно.

ЛВС — локальная вычислительная сеть.

Маршрутизатор (router) — компьютер сети, занимающийся маршрутизацией пакетов в сети, то есть выбором кратчайшего маршрута следования пакетов по сети.

Модем — устройство, преобразующее цифровые сигналы в аналоговую форму и обратно. Используется для передачи информации между компьютерами по аналоговым линиям связи.

НЖМД — накопители на жестком магнитном диске.

Протокол — совокупность правил и соглашений, регламентирующих формат и процедуру между двумя или несколькими независимыми устройствами, или процессами. Стандартные протоколы позволяют связываться между собой компьютерам разных типов, работающим в разных операционных системах.

Ресурс — логическая или физическая часть системы, которая может быть выделена пользователю или процессу.

Сервер — программа для сетевого компьютера, позволяющая предоставить услуги одного компьютера другому компьютеру. Обслуживаемые компьютеры сообщаются с сервер-программой при помощи пользовательской программы (клиент-программы). Компьютер в сети, предоставляющий свои услуги

другим, то есть выполняющий определенные функции по запросам других.

Узел — компьютер в сети, выполняющий основные сетевые функции (обслуживание сети, передача сообщений и т.п.).

Хост — сетевая рабочая машина; главная ЭВМ. Сетевой компьютер, который помимо сетевых функций (обслуживание сети, передача сообщений) выполняет пользовательские задания (программы, расчеты, вычисления).

Шлюз — станция связи с внешней или другой сетью. Может обеспечивать связь несовместимых сетей, а также взаимодействие несовместимых приложений в рамках одной сети.

Электронная почта — обмен почтовыми сообщениями с любым абонентом сети Internet.

ЛИТЕРАТУРА

1. Встроенная справка Microsoft Excel.
2. <http://www.microsoft.com/rus/> — Российская страница компании Microsoft.
3. «Microsoft Office 8.0», Эд Ботт.— М.: БИНОМ, 1998 г.
4. Каратыгин С. А. Access 97 (серия «Без проблем!»).— М.: Восточная Книжная Компания, 1997.— 368 с.
5. Минаси М. Windows 98: Полное руководство.— СПб.: BHV, 1999.— 800 с.
6. Сигел Ч. Изучи сам Access 97 / Перев. с англ. Ю. В. Климен.— ПМинск.: ООО «Поппури», 1998.— 352 с.
7. Хансен Г. и др. Базы данных: разработка и управление.— М.: Бинном, 1999.— 704 с.
8. Макарова Н. В., Информатика: Учебник / под ред. проф. Макаровой Н. В.— М.: Финансы и статистика, 1997.
9. Девис У. Операционные системы.— М.: Мир, 1980 г.
10. Хафкемейер Х. "Интернет. Путешествие по всемирной компьютерной сети", 1999 г.
11. Грошев С. В. Современный самоучитель профессиональной работы на компьютере, 1998 г.
12. Хакимова Т. Практикум по курсу "Основы информатики": Учебное пособие.— Алматы: Научно-издательский центр «Фылым», 2001.— 117 с.
13. Хакимова Т. Компьютерлік өңдеудің әдістемелері.: Оқу құралы.— Алматы: Фылым, 2002.— 160 б.
14. Хакимова Т. Специальные программы для работы на персональном компьютере.: Учебное пособие.— Алматы: Қазақ университеті, 2004.— 31 с.
15. Хакимова Т. Х. Практикум самостоятельных работ по обучению автоматизации обработки данных.: Учебное пособие.— Алматы: Қазақ университеті, 2005.— 85 с.
16. Жангисина Г. Д., Байсалбаева К. Н. «Лабораторный практикум по компьютерной графике».— А.: АТУ, 2006.— 81 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. База данных.....	3
1.1. Проектирование базы данных.....	6
1.2. Шаги проектирования базы данных.....	10
1.3. Создание базы данных.....	17
2. Модели задач базы данных.....	20
2.1. Модель № 1.Создание однотобличной базы данных..	20
2.2. Модель № 2. Формирование запросов и отчетов для однотобличной базы данных.....	31
2.3. Модель № 3. Разработка инфологической модели и создание структуры реляционных базы данных.....	34
2.4. Модель № 4.Формирование сложных запросов.....	45
2.5. Модель № 5. Создание сложных форм и отчетов.....	51
2.6. Модель № 6. Создание базы данных.....	57
3. Связь информационных сетей.....	65
3.1. Internet — глобальная компьютерная сеть.....	65
3.2. Протоколы сети Internet.....	65
3.3. Услуги предоставляемые сетью.....	67
3.4. Гипертекстовая технология WWW, URL, HTML.....	69
4. Защита информации в глобальной сети Internet	74
4.1. Информационная безопасность и информационные технологии.....	75
4.2. Средства защиты информации.....	77
4.3. Ограничения доступа в WWW серверах.....	87
4.4. Информационная безопасность в Internet.....	89
4.5. Список специальных терминов.....	102
ЛИТЕРАТУРА.....	108

*Жангисина Гульнур Давлетжановна
Хакимова Тийшитик*

**ТЕОРИЯ И МЕТОДИКА
КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ
ДЛЯ ЗАДАЧ БАЗЫ ДАННЫХ
И ГЛОБАЛЬНОЙ СЕТИ**
Учебное пособие

Главный редактор Н.Н. Жансеитов
Корректор Н.Н. Жансеитов
Компьютерная верстка Г.Ж. Жансеитова
Дизайнер К.К. Карпун

Подписано в печать 25.07.2019 г.
Формат 60x84/16. Бумага офсетная.
Гарнитура «Таймс»
Усл. печ. л. 6,8. Тираж 500 экз.

Издательство «NURPRESS»050057
г. Алматы, ул. Джандосова, д. 58, оф. 204.
Тел.: +7(727) 226-03-29
Тел.: +7(707) 226-03-29
E-mail: nurpress@mail.ru