

Раздел 6. Информационные ресурсы распределенных информационных систем

Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач из любой области. Они помогают анализировать проблемы и создавать новые продукты.

Информационной системой (ИС), либо автоматизированной ИС (**АИС**), будем называть программно-аппаратную систему, предназначенную для автоматизации целенаправленной деятельности конечных пользователей, обеспечивающую, в соответствии с заложенной в нее логикой обработки, возможность получения, модификации и хранения информации.

Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации компьютера. В крупных организациях наряду с персональными компьютерами в состав технической базы информационной системы может входить мэйнфрейм или суперЭВМ. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

Необходимо понимать разницу между компьютерами и информационными системами. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом для информационных систем. Информационная система немыслима также без персонала, взаимодействующего с компьютерами и телекоммуникациями.

Информационная система является средой, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технические и программные средства связи и т.д. Основная цель информационной системы – организация хранения и передачи информации. Информационная система представляет собой человеко-компьютерную систему обработки информации.

В результате эволюционного развития технологии хранения и использования данных утвердилась концепция *автоматизированных информационных систем (АИС) или банков и баз данных*. Широкое распространение получили базы данных, построенные на основе различных подходов к организации данных.

6.1. Базы данных

6.1.1. Общие понятия

Научно-технические исследования, направленные на поиск путей построения АИС общего назначения с изменяющейся информационной моделью объектов управления привели к возникновению концепции базы данных.

Системы, основанные на концепции баз данных, в наибольшей степени отвечают современным требованиям по построению информационных систем. Эта концепция предусматривает коллективное использование данных. Названная концепция отличается высоким универсализмом и пригодна для создания систем различных типов - от персональных до глобальных. База данных может быть локальной (централизованной) или распределенной.

Достоинством баз данных являются: хорошая структуризация информации, поддержание ее целостности и непротиворечивости, небольшая избыточность представления в памяти компьютера; снижения трудоемкости сбора и обновления данных (однократная подготовка и многократное применение данных для решения различных задач различными должностными лицами). В идеале любая единица данных может храниться в единственном экземпляре, а некоторая разумная избыточность вводится для улучшения эксплуатационных характеристик информационной системы.

При затребовании нужной информационной единицы пользователь не связан необходимостью строгого указания пути к этой единице. Он только формулирует запрос на данные, а удовлетворение этого запроса полностью возлагается на систему управления базами данных.

Информационные системы, построенные на основе баз данных, отличаются гибкостью, хорошей приспособленностью к наращиванию выполняемых функций, позволяют оперировать разнородной информацией и не требуют высокой квалификации пользователей. Расплачиваться за это приходится увеличением сроков их разработки. При проектировании и внедрении необходим сугубо профессиональный подход.

Документальные системы также используют концепцию баз данных, но в базах данных хранятся не структурированные совокупности характеристик объектов, а целые документы без выделения их структурных единиц. Применяются, например, для хранения, пополнения и поиска нормативных актов, отвечающих определенным условиям.

Развитием баз данных является технология *хранилищ (складов) данных*. В отличие от базы хранилище содержит не только и не столько информацию о современном состоянии моделируемой части реального мира, но и накапливает ее во времени. На основании таких накоплений

возникает возможность построения тенденций поведения объектов, сведения о которых находятся в хранилище. Благодаря этому свойству подобные информационные системы хорошо приспособлены для углубленного ретроспективного и прогностического анализа предметной области, информационной поддержки задач, связанных с принятием решений и т.д.

По характеру своего размещения хранилища, также как и базы данных, могут быть локальными или распределенными. В информационных системах они могут использоваться самостоятельно или совместно с базами данных в зависимости от целевого назначения системы.

6.1.2. Состав автоматизированной информационной системы

Информационное обслуживание управленческого персонала и решаемых ими задач реализуется с помощью технических средств, программного обеспечения и **информационной базы (ИБ)**. Под информационной базой принято понимать совокупность показателей, документов, словарей, массивов информации, а также методов организации их хранения и контроля, обеспечивающих решение задач в системе управления.

Различают **внемашинную ИБ** - совокупность всех документированных данных и сообщений, используемых в системе, и **внутримашинную ИБ** - совокупность всех данных на машинных носителях, сгруппированных по определенному признаку.

В дальнейшем внимание будет сосредоточено на внутримашинной ИБ, построенной на основе концепции базы данных. Такие АИС получили наименование *банка данных*. Эта концепция предусматривает выделение в составе АИС двух принципиально важных компонент:

базы данных (БД) как совокупности формализованных данных;
системы управления базой данных (СУБД) как самостоятельной системы, включающей основные процедуры информационного обслуживания.

Именно четкое выделение в составе банка данных двух компонентов определяет назначение, возможности и функции АИС, обеспечивает необходимую независимость задач управления и реализующих их программ от структур и характеристик хранимых данных, свойств среды размещения. Поэтому часто АИС рассматривают в узком смысле как совокупность БД и СУБД.

Практически все современные информационные системы строятся на основе рассматриваемой концепции. Ее сущность состоит в интеграции данных и централизации управления ими для обеспечения многоаспектного использования. Этим обеспечивается необходимый уровень независимости между техническими, программными и информационными средствами систем, что позволяет адаптировать

последние к текущим требованиям пользователей, а также совершенствовать в процессе эксплуатации.

База данных представляет собой идентифицированную, структурированную, коллективно используемую совокупность данных, связанных определенным образом и относящуюся к конкретной предметной области. Здесь понятия:

«идентифицированная» означает, что компоненты БД имеют свои имена и операции над ними оформляются путем указания их имен, а не адресов;

«структурированная» – данные имеют четкую структуру, т.е. информация хранится в формализованном виде в заранее установленных форматах, определяющих вид данных (например, числовые, текстовые), размерность и другие характеристики. Состав и связи компонентов данных отражают свойства и отношения объектов управления. В базе данных может храниться и неформализованная информация в виде обычного текста, изображений (например, фотографий сотрудников);

«коллективное использование» предполагает централизованное накопление и многоаспектное применение данных (при этом данные вводятся однократно, а используются при решении различных задач в интересах различных пользователей). Понятно, что для персональных ИС не предусматривается применение данных различными пользователями.

В базе данных выделяют следующие категории данных: *проблемные (первичные)* – описывающие предметную область и необходимые пользователям для решения их задач, и *вторичные* – обеспечивающие эффективное хранение и доступ к первичным данным.

В состав БД могут входить следующие **массивы** данных:

- 1) *основные* – используемые при пополнении, корректировке, поиске, и контроле данных;
- 2) *массивы для восстановления базы* – страховые копии;
- 3) *массивы словарей*, используемые при контроле вводимых данных, их кодировании и декодировании;
- 4) *массивы для учета и разграничения доступа* – таблицы паролей, учетные журналы;
- 5) *массивы статистических данных* о работе базы и др.

Система управления базой данных предназначена для реализации типовых процедур информационного обслуживания при создании АИС и входе ее эксплуатации. СУБД работает под управление ОС ЭВМ и расширяет ее возможности по управлению данными.

В общем случае АИС может включать несколько БД и соответственно СУБД.

Эксплуатацию учрежденческой или ведомственной АИС осуществляет администратор, в качестве которого выступает должностное лицо или группа лиц обслуживающего персонала. На администратора возлагаются задачи по разработке описания БД, формированию и настройке средств СУБД, поддержанию целостности БД, выбору алгоритмов обращения к данным, анализу качества работы АИС, реорганизации БД и СУБД при изменении условий или требований по эксплуатации, защите данных от несанкционированного доступа.

Пользователями АИС являются должностные лица органов управления. Они обращаются с помощью запросов на поиск данных или их корректировку. Обычно каждый пользователь имеет доступ к определенной совокупности данных для совершения ограниченного набора действий. К АИС обращаются также и программы функциональных задач.

6.1.3. Уровни представления данных

Для реализации независимости данных от их описания и от использующих их программ используют многоуровневое представление данных. Многоуровневое представление данных предложено исследовательской группой в области баз данных ANSI/SPARC (American National Standards Institute / System Planning and Requirements Committee – Комитет по системному планированию и выработке требований Американского национального института стандартов). Структурная основа этого представления включает **три уровня**, каждому из которых ставится в соответствие модель данных. Описания моделей данных средствами СУБД получили название *схем*. В число уровней входят внешний, концептуальный и внутренний, рис. 6.1.

Внешний, предназначенный для описания пользовательского представления базы данных. Используется при рассмотрении вопросов, связанных со смысловым содержанием информации независимо от способа ее представления в памяти ЭВМ. На этом уровне выделяют:

объекты предметной области, сведения о которых накапливаются в АИС;

основные характеристики объектов;

связи между ними.

Описание данных на внешнем уровне называется **инфологической моделью**.

Концептуальный, дающий логическое описание части реального мира, моделируемого базой данных. Это основа построения базы данных и является отображением инфологической модели на средства реализации базы с помощью СУБД;

Внутренний – служит для описания представления базы данных на машинных носителях.

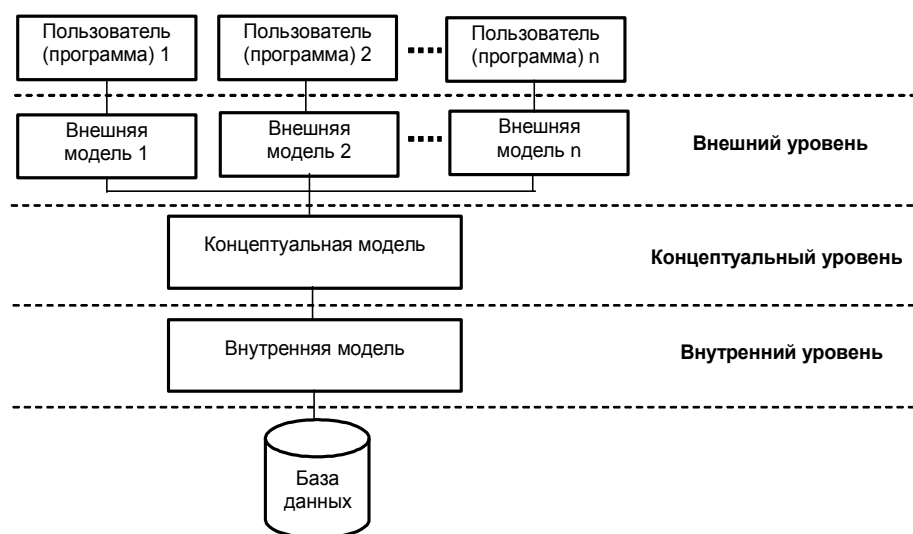


Рис. 6.1. Уровни представления данных

Обеспечение независимости структуры базы данных от хранимой в ней информации основано на том, что разнообразные пользовательские представления в виде неоднородных моделей описываются множеством *внешних схем*. *Концептуальная схема* дает обобщенное описание пользовательских представлений и не зависит от принципов конкретной реализации базы данных. В свою очередь, структура базы данных определяются собственной *внутренней схемой*.

Введение в рассмотрение трех уровней представления данных приводит к тому, что для обеспечения доступа к данным необходимо обеспечить *три уровня отображения*:

- 1) внешняя модель – концептуальная модель;
- 2) концептуальная модель – внутренняя модель;
- 3) внутренняя модель – физическая база данных.

Первые два типа обеспечиваются СУБД, последний – средствами операционной системы компьютера.

Механизм обеспечения независимости прикладных программ от данных состоит в следующем. Отображение *внешняя модель – концептуальная модель* обеспечивает независимость прикладных программ от логической структуры, определяемой концептуальной моделью. Изменение этой структуры не требует модификации прикладных программ, меняется лишь отображение *внешняя модель – концептуальная модель*.

Отображение *концептуальная модель – внутренняя модель* обеспечивает независимость логической и физической структур базы данных.

Отображение *внутренняя модель* – *физическая база данных* обеспечивает независимость операций хранения и обработки данных от используемых технических средств. Для этого используются либо стандартные методы доступа операционных систем к данным, либо разрабатываются собственные методы доступа СУБД.

Таким образом, изменение структуры данных на одном из уровней представления (в одной из моделей) приводит лишь к необходимости изменения отображения этого уровня на смежные. Это и обеспечивает независимость данных от прикладных программ пользователей.

6.1.4. Модели данных

Реализация многоуровневого подхода, обеспечивающего интеграцию и независимость различных способов (моделей) представления данных о предметной области, является основой функционирования базы данных.

Модели данных разделяются на сетевые, иерархические, реляционные и объектно-ориентированные. В настоящее время выделяют также постреляционные и многомерные.

В настоящее время наибольшее распространение получили реляционные базы данных, так как похожими картотеками пользовались до появления электронных баз данных. Сетевые и иерархические базы данных считаются устаревшими, объектно-ориентированные, постреляционные и многомерные пока не стандартизированы и не получили широкого распространения.

Реляционные базы данных – базы данных, основанные на реляционной модели. Слово «реляционный» происходит от английского «relation» (отношение). Для работы с реляционными БД применяют реляционные СУБД. Теория реляционных баз данных была разработана доктором Коддом из компании IBM в 1970 году. В реляционных базах данных все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные. Запросы к таким таблицам возвращают таблицы, которые сами могут становиться предметом дальнейших запросов. Каждая база данных может включать несколько таблиц, которые, как правило, связаны друг с другом, откуда и произошло название реляционные. Кратко особенности реляционной базы данных можно сформулировать следующим образом:

в одной таблице хранятся сведения об однотипных объектах, т.е. объектах обладающих одинаковым набором свойств. Объект – компонент предметной области, информацию о котором следует хранить. Объект может быть реальным или абстрактным;

каждый столбец таблицы соответствует одному простому свойству объекта. Набор значений одного столбца и совокупность правил, определяющих допустимость значений этого столбца, называется *доменом*;

каждая строка (кортеж) содержит сведения о конкретном объекте (экземпляре объекта);

в заполненной таблице не допускается наличие одинаковых по содержанию строк;

таблицы и имена столбцов в пределах каждой таблицы должны быть уникальными;

в каждой таблице следует назначить единственный ключ. Ключ может состоять из одного или нескольких столбцов. Ключ обеспечивает однозначную идентификацию любого объекта в таблице (значение ключа не повторяется в таблице);

в таблицах могут назначаться так называемые *индексы*. Индексы служат для ускорения поиска нужных сведений и для связывания таблиц. Индексы позволяют СУБД просматривать БД как бы упорядоченную по его значению. Например, пусть имеется БД абонентов телефонной сети, и информация в ней упорядочена по ключу - номеру телефона. Поиск по фамилии требует полного просмотра такой базы; но если создать индекс по столбцам "Фамилия", "Имя", "Отчество", то такой индекс ускорит поиск сведений об абоненте по его фамилии, имени и отчеству. Индексы можно создавать по любому столбцу или совокупности столбцов. Таблица может содержать несколько индексов. Значения индексов могут повторяться;

запросы к базе данных возвращают результат (выборку данных) в виде таблиц, которые тоже могут выступать как объект запросов.

Строки в реляционной базе данных неупорядочены – упорядочивание производится в момент формирования ответа на запрос.

Приведем теоретические основы реляционной модели данных.

В реляционной модели данные представляются в виде совокупности двумерных таблиц, называемых *отношениями*. Модель получила название от англ. *relation* – отношение.

Для описания реляционных структур данных используются следующие понятия.

Атрибут – элементарная единица данных, значения которой заносятся в одну из граф таблицы. Атрибуты, позволяющие однозначно выбирать отдельные кортежи, называются *ключами* отношения. Они содержат уникальные значения.

Домен – множество значений, которые может принимать атрибут.

Кортеж – упорядоченный набор значений атрибутов, число которых равно числу граф таблицы (по сути, это строка таблицы). Кортеж содержит совокупность значений всех атрибутов отношения, характеризующую один и тот же объект предметной области.

Отношение – двумерная таблица (рис.6.2), представляющая набор однотипных кортежей и удовлетворяющая определенным требованиям:

атрибутами отношений могут быть только элементарные данные, взятые из некоторого фиксированного домена;

в одном отношении все кортежи имеют одинаковую структуру, в то время как в различных отношениях могут быть разные кортежи;

в одном отношении не может быть двух одинаковых кортежей;

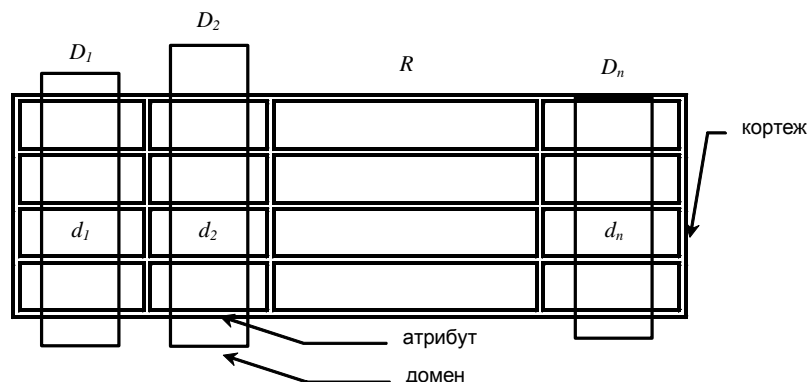


Рис. 6.2. Графическое представление отношения

просмотр кортежей в одном отношении может осуществляться в любой последовательности безотносительно к содержанию.

Число атрибутов – это *степень (порядок)* отношения, а число кортежей – его *мощность*.

Упорядоченная совокупность имен атрибутов, входящих в отношение, с выделением среди них ключей называется *логической структурой*, или *схемой отношения*. Совокупность всех схем отношений базы данных называется *логической структурой*, или *схемой БД*.

В памяти ЭВМ каждое отношение представляется в виде файла. Такой файл состоит из последовательности записей, по одной на каждый кортеж отношения. При этом одинаковые записи исключаются. Это следует из требования о необходимости отсутствия в одном отношении двух одинаковых кортежей. Все записи должны быть однотипны, т.е. у них должно быть одно и то же количество полей, поля разных записей должны следовать в строго определенном порядке и в соответствующих полях должна храниться информация одного и того же типа. Нетрудно провести соответствие между основными понятиями:

Объект	Таблица	Отношение	Файл
экземпляр	строка	кортеж	запись
атрибут	столбец	атрибут	поле

Объекты предметной области находятся по отношению друг к другу в определенных отношениях (функциональных, подчиненности, видовых и т.д.). Существенные отношения должны найти отражение в

БД. В БД не указываются содержательные аспекты этих отношений, а находят отражение только наличие и формальный вид этих отношений:

один к одному (1:1). Одному экземпляру объекта А соответствует один экземпляр объекта Б или не соответствует ни один экземпляр объекта Б. Это соотношение симметрично. Примером может служить связь таких объектов как «муж» и «жена»;

один ко многим (1:М или 1: ∞). Одному экземпляру Объекта А соответствует любое количество (0, 1, 2, ...) экземпляров объекта Б, а любому экземпляру объекта Б соответствует один экземпляр объекта А. Примером может служить отношение объектов «учебная группа» и «студент»;

многие к одному. По сути, этот тип связи эквивалентен предыдущему.

Формально существуют связи типа **многие ко многим**, например между такими объектами «учебные группы» и «учебные дисциплины». В реляционных БД этот вид связи обычно не допускается. Если необходимо отобразить такое отношение объектов, то следует его преобразовать к совокупности связей типа один ко многим. Для этого требуется три таблицы: по одной для каждого объекта и третья - для хранения связей между ними (промежуточная таблица). В этой третьей таблице для ключа первой таблицы указываются значения ключа второй таблицы.

Связывание таблиц осуществляется на основе ключей и индексов: в исходной таблице в качестве основы для связи используется ключ;

в подчиненную таблицу для обеспечения связи включают те же поля, что и ключи в исходной таблице, но только объявляют их как индексы (значения этих полей в подчиненной таблице могут повторяться).

Главными достоинствами реляционных БД являются:

простота представления данных (табличная форма часто применяется должностными лицами для хранения информации);

простота внесения изменений в базу данных;

упрощение процедур разграничения доступа к данным в разных таблицах;

простота физической реализации двумерных таблиц др.

6.2. Системы управления базами данных

Система управления базами данных (СУБД) – специализированная программа (чаще комплекс программ), предназначенная для манипулирования базой данных. Для создания и управления информационной системой СУБД необходима в той же степени, как для разработки программы на алгоритмическом языке необходим транслятор.

Таким образом, СУБД определяется как система программного обеспечения, которая позволяет:

на стадии создания АИС – формировать описание БД, настраивать типовые средства на конкретные условия применения;

на стадии эксплуатации – обрабатывать обращения к базе данных от прикладных программ и/или пользователей и поддерживать целостность базы (**целостность** – это состояние БД, при котором все значения данных правильны в том смысле, что отражают состояние реального мира и подчиняются правилам взаимной непротиворечивости). СУБД обеспечивает связь между прикладными программами или пользователями и базой данных. Любой доступ к данным осуществляется через СУБД.

Использование СУБД обеспечивает:

1) минимизацию избыточности данных – в предельном случае любые данные могут храниться в одном экземпляре;

2) совместное использование данных многими пользователями;

3) независимость данных от программ;

4) эффективность доступа к данным, как удовлетворение требований по своевременности, достоверности и др.;

5) простоту работы с базой и т.д.

СУБД работает под управлением операционной системы. Она служит инструментом для работы с базой данных и разработки прикладных программ. Программы СУБД для своей работы используют служебные данные, т.е. эту систему в функциональном отношении можно рассматривать как совокупность программ и служебных данных.

Обычно на СУБД возлагается выполнение следующих *функций*:

1) описание данных;

2) манипулирование данными;

3) заведение базы данных;

4) выполнение запросов;

5) выдача отчетов;

6) сервис (поддержание целостности, справочные функции, восстановление базы).

Существует широкий класс СУБД различного назначения и областей применения. В дальнейшем будем рассматривать СУБД для работы на персональных компьютерах.

Персональные компьютеры оказали большое влияние на развитие технологии баз данных и ее массовое распространение. Появление персональных компьютеров привело к значительному изменению уровня инструментального оснащения разработок информационных систем, основанных на концепции баз данных. В то время, как выпуск СУБД для «больших» или мини-ЭВМ являлся исключительным событием, сейчас сформировалась мощная индустрия, производящая средства программного обеспечения для создания баз данных на ПЭВМ.

Созданное к настоящему времени программное обеспечение для персональных компьютеров представлено десятками продуктов, позволяющее создавать и эксплуатировать базы данных практически на всех моделях компьютеров в обстановке различных операционных систем и в интересах широкого круга пользователей.

К средствам, предназначенным для разработки и ведения баз данных, относятся не только СУБД, но и разнообразные средства их окружения:

- 1) компиляторы языков программирования СУБД;
- 2) отладчики;
- 3) средства разработки меню и экранных форм ввода-вывода данных;
- 4) средства графического представления данных;
- 5) интерфейсные средства для доступа к базе в рамках традиционных языков программирования и т.д.

Важнейшим свойством, присущим инструментальным средствам для построения баз данных на ПК, является развитый пользовательский интерфейс.

подавляющее большинство СУБД для персональных компьютеров поддерживает реляционную модель данных. Набор систем, поддерживающих иерархическую, сетевую и другие модели, довольно ограничен.

Многообразие созданных к настоящему времени СУБД требует проведение их классификации с целью всесторонней характеристики. В основу этой классификации могут быть положены разнообразные признаки. Представляется целесообразным указание признаков классификации с последующей краткой характеристикой выделяемых классов.

Тип поддерживаемых моделей данных. По этому признаку различают СУБД, поддерживающие иерархические, сетевые, реляционные и другие модели. Примерами иерархических СУБД являются БИСОД (Базовая информационная система обработки данных), МИРИС (Малая информационно-расчетная измерительная система). Яркий образец СУБД сетевого типа – ДИСОД (Диалоговая информационная система обработки данных). Эти СУБД широко применялись на больших и мини-ЭВМ в 80-90-х годах прошлого века.

Современные СУБД в подавляющем большинстве поддерживают реляционную модель данных.

Класс используемых аппаратных платформ. Здесь выделяют СУБД, ориентированные на работу в среде больших машин (mainframe), с одной стороны, и персональных компьютеров различной архитектуры (IBM PC, Apple Macintosh, Sun и т.д.), с другой.

Тип вычислительных систем. В соответствии с таким делением можно указать СУБД для автономно используемых компьютеров, а также СУБД для работы в глобальных и локальных сетях, так называемые системы управления распределенными базами данных (СУРБД). Примеры систем управления локальными базами данных – это ранние СУБД для персональных компьютеров: DBase, FoxBase, Ребус и т.п. Современные СУБД, как правило, способны функционировать в распределенных средах. К наиболее известным системам можно отнести MSt Access, Paradox, MSt SQL Server, Oracle, Informix и др.

Степень промышленного освоения. Деление всего множества СУБД по данному признаку позволяет обнаружить стандартные, промышленно эксплуатируемые типовые СУБД, и системы, в основе которых лежат уникальные разработки.

Характер создаваемых приложений. По этому признаку можно выделить СУБД, используемые для разработки баз данных средней степени сложности и объема. Такие базы применяются при построении персональных информационных систем и информационных систем предприятий малого и среднего бизнеса. Несомненным лидером здесь выступает СУБД MSt Access, позволяющая разрабатывать гибкие и удобные в работе с точки зрения конечных пользователей системы. Альтернативными являются СУБД, предназначенные для построения крупных корпоративных баз данных высокой степени сложности. В качестве примера можно выделить СУБД MS SQL Server, которая принципиально позволяет поддерживать несколько сотен баз данных, каждая из которых может удовлетворять информационные потребности десятков и сотен пользователей.

В проектировании баз данных выделяются следующие этапы:
анализ информационных потребностей;
инфологическое моделирование;
логическое проектирование;
физическая реализация.

Анализ информационных потребностей. На этом этапе разработчик базы данных анализирует информацию, циркулирующую в органе управления, для которого разрабатывается база данных, и определяет, какие задачи будут решаться основными пользователями – должностными лицами – с помощью базы данных. В обязательном порядке на этом этапе определяется, с какими запросами пользователи будут обращаться к базе и какие выходные документы (отчеты) они будут получать. При этом для запросов определяются

перечень выдаваемой информации, условия поиска, частота выполнения запроса, кто из должностных лиц его будет выполнять. Для отчетов дополнительно определяется форма документа, выводимого на печать. Полученные на этом этапе решения оформляются в произвольном виде (обычно табличном).

Инфологическое моделирование. Основной задачей этого этапа является разработка ИЛМ предметной области. Исходными данными для этого являются результаты, полученные на предыдущем этапе, а также знания о специфике предметной области. ИЛМ должна отображать объекты (сущности) предметной области, их учитываемые характеристики – атрибуты, а также взаимные связи между объектами и атрибутами. ИЛМ представляется чаще всего в виде графической диаграммы.

Кроме формирования ИЛМ, на данном этапе дается характеристика всех атрибутов, учитываемых в базе данных. Она предполагает указание принадлежности атрибута (к объекту или к связи), типа данных, к которому относятся значения атрибутов, их длины, области допустимых значений, ограничений целостности, выводимости из значений других атрибутов и ряд других характеристик. Результаты оформляются в табличном виде.

Логическое проектирование. Исходными данными являются результаты, полученные на этапе инфологического моделирования. Основным результатом этого этапа является логическая структура базы, называемая реляционной схемой базы данных.

Проектирование реляционных схем является одним из самых сложных и ответственных этапов всего процесса проектирования. Одной из ключевых задач здесь является нормализация отношений, т.е. приведение схем отношений к требуемой нормальной форме. Для нормализации баз данных разработаны специальные методы.

Кроме проектирования реляционной схемы, на этапе логического проектирования осуществляется оценка качества будущей базы данных по таким показателям, как ее предполагаемый объем и оперативность выполнения запросов.

Физическая реализация. На этом этапе в среде выбранной СУБД формируются структуры файлов-таблиц, осуществляется их первоначальное заведение, разрабатываются файлы-запросы и файлы-отчеты в соответствии с решениями, полученными на первом этапе. Таким образом, промежуточным результатом, полученным на этом этапе, является демонстрационный прототип (макет) базы данных, показывающий возможность (или невозможность) реализации всех предъявляемых к ней требований.

В случае, если макет удовлетворяет предъявляемым требованиям, база данных заполняется до полного объема и передается в опытную эксплуатацию. В противном случае осуществляется возврат на один из

предыдущих этапов с целью уточнения полученных на нем результатов.

6.3. Системы распределенных вычислений

Некоторые задачи являются настолько требовательными к вычислительным мощностям, что даже мощнейшие из современных суперкомпьютеров не справляются. Немаловажным фактором является и стоимость оборудования (если требуется создать новую суперсистему) или стоимость машинного времени (в случае использования какого-либо суперкомпьютера). В первом случае стоимость исчисляется сотнями, во втором – десятками... миллионов долларов (евро и.т.п.), разумеется. При таком ценовом раскладе проект имеет все шансы стать нерентабельным и лишиться всякого финансирования.

Изящное решение не заставило себя ждать. **Системы распределенных вычислений.** В их основу положен принцип разбиения одной задачи на множество подзадач, с решением которых легко справится среднестатистическая система. Данные, подлежащие обработке, рассылаются по Сети, обрабатываются и затем отсылаются на главный сервер, где происходит “сборка” результатов обработки. Преимущества такого подхода очевидны: легкая масштабируемость (читайте - расширяемость) Сети, производительность, соизмеримая с производительностью суперкомпьютеров, небольшая сумма вложенных средств.

Такие системы применяются не только в интернет-проектах, но и на уровне локальных сетей, например при сетевом рендеринге. Если требуется отрендерить много больших изображений, то данные рассылаются по сети и рендерингом занимаются сетевые компьютеры, а финальная картинка собирается на сервере.

Идея создания систем распределенных вычислений родилась в далеком 1970 году, когда компьютеры занимали комнаты, гудели многоваттными блоками питания, лениво поедали тонны перфокарт и неторопливо подмигивали системщикам лампочками на панелях. Первые эксперименты с сетевыми программами вылились в создание первого вируса, распространяющегося по сети под именем Creeper (“Вьюнок”), и последовавшим за ним его убийцы Reaper (“Жнец” или “Потрошитель”).

Распространяясь по прародителю современного Интернета – сети ARPAnet, обе программки эффективно загружали память сетевых машин и отнимали драгоценное процессорное время. “Вьюнок” делал это из вредности, выдавая текстовые сообщения, а “Жнец” сканировал память машины на предмет наличия паразита. Под покровом тайны остался факт, какая же программка больше загружала машину.

В 1973 году детище компании PARC (Xerox Palo Alto Research Center), являвшееся по своей сути первым “червем”, последовательно и обстоятельно загрузило 100 компьютеров в Ethernet-сети компании таким образом, что все свободное (!) процессорное время было отдано под деятельность червя: создание и рассылку себе подобных. Такая на первый взгляд бесполезная вещь, как вирус, дала идею для создания систем сетевого рендеринга на базе компьютеров Apple.

Затем последовало затишье... Новый прорыв в области систем распределенных вычислений пришелся на период экспансии сети Интернет в начале 90-х. В первом проекте, получившем широкую огласку, были задействованы несколько тысяч компьютеров по всей глобальной Сети. Целью проекта был взлом алгоритма шифрования методом прямого перебора. Но вторым и значительно более популярным проектом стал SETI@home.

В дальнейшем возникла идея мета-компьютинга. Термин возник вместе с развитием высокоскоростной сетевой инфраструктуры в начале 90-х годов и относился к объединению нескольких разнородных вычислительных ресурсов в локальной сети организации для решения одной задачи. Основная цель построения мета-компьютера в то время заключалась в оптимальном распределении частей работы по вычислительным системам различной архитектуры и различной мощности. Например, предварительная обработка данных и генерация сеток для счета могли производиться на пользовательской рабочей станции, основное моделирование на векторно-конвейерном супер-компьютере, решение больших систем линейных уравнений – на массивно-параллельной системе, а визуализация результатов – на специальной графической станции.

В дальнейшем, исследования в области технологий мета-компьютинга были развиты в сторону однородного доступа к вычислительным ресурсам большого числа (вплоть до нескольких тысяч) компьютеров в локальной или глобальной сети. Компонентами мета-компьютера могут быть как простейшие ПК, так и мощные массивно-параллельные системы. Что важно, мета-компьютер может не иметь постоянной конфигурации - отдельные компоненты могут включаться в его конфигурацию или отключаться от нее; при этом технологии мета-компьютинга обеспечивают непрерывное функционирование системы в целом. Современные исследовательские проекты в этой области направлены на обеспечение прозрачного доступа пользователей через Интернет к необходимым распределенным вычислительным ресурсам, а также прозрачного подключения простаивающих вычислительных систем к мета-компьютерам.

Очевидно, что наилучшим образом для решения на мета-компьютерах подходят задачи переборного и поискового типа, где вычислительные узлы практически не взаимодействуют друг с другом и основную часть работы производят в автономном режиме. Основная

схема работы в этом случае примерно такая: специальный агент, расположенный на вычислительном узле (компьютере пользователя), определяет факт простоя этого компьютера, соединяется с управляющим узлом мета-компьютера и получает от него очередную порцию работы (область в пространстве перебора). По окончании счета по данной порции вычислительный узел передает обратно отчет о фактически проделанном переборе или сигнал о достижении цели поиска.

Далее будут описаны и приведены ссылки на основные исследовательские проекты в области мета-компьютинга, разработанные программные технологии, конкретные примеры мета-компьютеров

"Distributed.net". <http://www.distributed.net/>.

Одно из самых больших объединений пользователей Интернет, предоставляющих свои компьютеры для решения крупных переборных задач. Основные проекты связаны с задачами взлома шифров (RSA Challenges). В частности, 19 января 1999 года была решена предложенная RSA Data Security задача расшифровки фразы, закодированной с помощью шифра DES-III. В настоящее время в distributed.net идет работа по расшифровке фразы, закодированной с 64-битным ключом (RC5-64). С момента начала проекта в нем зарегистрировались 191 тыс. человек. Достигнута скорость перебора, равная 75 млрд. ключей в секунду (всего требуется проверить 264 ключей). За решение этой задачи RSA предлагает приз в \$10 тыс.

GIMPS – Great Internet Mersenne Prime Search.

<http://www.mersenne.org/> .

Поиск простых чисел Мерсенна (т.е. простых чисел вида 2^p-1). С начала проекта было найдено 4 таких простых числа. Организация Electronic Frontier Foundation предлагает приз в \$100 тыс. за нахождение простого числа Мерсенна с числом цифр 10 миллионов.

Проект SETI@home (Search for Extraterrestrial Intelligence) – поиск внеземных цивилизаций с помощью распределенной обработки данных, поступающих с радиотелескопа. Присоединиться может любой желающий. Доступны клиентские программы для Windows, Mac, UNIX, OS/2 (клиент Windows срабатывает в качестве screen-saver'a). Для участия в проекте зарегистрировались около 920 тыс. человек.

Globus. <http://www.globus.org>.

Разработка ПО для организации распределенных вычислений в Интернет. Проект реализуется в Argonne National Lab. Цель The Globus Project – построение т.н. "computational grids", включающих в себя вычислительные системы, системы визуализации, экспериментальные установки. В рамках проекта проводятся исследования по построению распределенных алгоритмов, обеспечению безопасности и отказоустойчивости мета-компьютеров.

В рамках проекта Globus разработан ряд программных средств:

- Globus Resource Allocation Manager – единообразный интерфейс к различным "локальным" системам распределения нагрузки (LSF, NQE, LoadLeveler) Для описания требований приложения к ресурсам разработан специальный язык RSL (Resource Specification Language)
- Globus Security Infrastructure – система аутентификации на базе открытого ключа и X.09-сертификатов
- Metacomputing Directory Service (MDS) – репозиторий информации о вычислительных ресурсах, входящих в метакомпьютер
- Nexus - коммуникационная библиотека
- Heartbeat Monitor (HBM) - средство мониторинга, позволяющее определить сбой некоторых машин и процессов, входящих в метакомпьютер
- Globus Access to Secondary Storage (GASS) – средство доступа к удаленным данным через URL

Программное обеспечение Globus 1.0.0 доступно бесплатно. Доступна также реализация MPI (MPICH-G) поверх Globus.

Для тестирования Globus был создан реальный метакомпьютер GUSTO (testbed environment), который включает около 40 компонент с суммарной пиковой производительностью 2.5 TFLOPS

Проект *Legion: A Worldwide Virtual Computer* университета Вирджинии. Цель – разработка объектно-ориентированного ПО для построения виртуальных мета-компьютеров, включающих до нескольких миллионов индивидуальных хостов, объединенных высокоскоростными сетями. Пользователь, работающий на своем домашнем компьютере, должен иметь абсолютно прозрачный доступ ко всем ресурсам мета-компьютера.

В рамках Legion возможно исполнение параллельных приложений – поддерживаются библиотеки MPI и PVM, а также язык Mentat. Программное обеспечение проекта доступно бесплатно (поддерживаются платформы SGI IRIX, Linux, Alpha/OSF1, RS/6000).

PACX-MPI. <http://www.hlr.de/structure/organisation/par/projects/pacx-mpi/>.

Расширение MPI для поддержки распределенных вычислений. Поддерживается объединение в единый мета-компьютер нескольких MPP-систем, возможно с различными реализациями MPI. Передача данных между MPP производится через Интернет с помощью TCP/IP. В настоящее время в рамках PACX-MPI реализовано подмножество стандарта MPI 1.2.

На конференции SC'98 было продемонстрировано совместное использование двух 512-процессорных суперкомпьютеров Cray T3E, находящихся в университете Штутгарта (Германия) и в PSC (Питтсбург, США). Для объединения компьютеров использовалась библиотека PACX-MPI.

Condor. <http://www.cs.wisc.edu/condor/> .

Система Condor разрабатывается в университете шт. Висконсин (Madison). Condor распределяет независимые подзадачи по существующей в организации сети рабочих станций, заставляя компьютеры работать в свободное время (то есть в то время, когда они простаивали бы без своих пользователей). Программное обеспечение системы Condor доступно бесплатно. В настоящее время поддерживаются платформы SGI, Solaris, Linux, HP-UX, и Digital Unix, однако планируется также поддержка Windows NT.

Недавно появился, но уже успел стать популярным английский термин Grid, который означает среду, в которой объединены находящиеся в разных местах глобальной телекоммуникационной сети вычислительные установки. Такая среда предназначена для выполнения распределенных приложений, использующих ресурсы этих установок.

История Грид начиналась с научных приложений, и до недавнего времени проект назывался «Мета-компьютинг». В начале 90-х годов возникла идея создать из многочисленных суперкомпьютерных центров США очень большой Мета-компьютер, так чтобы пользователи могли получать практически неограниченные ресурсы для вычислений и хранения данных. Реализация этой идеи продолжается до сих пор, однако, те методы, которые уже разработаны, оказались применимы не только для высокопроизводительных вычислений, но и для других областей жизни.

Сейчас интерес к Grid очень высок практически во всех странах мира, что выражается в большом количестве национальных и интернациональных проектов, исследовательских работ и публикаций по этой тематике. Объясняется это тем, что институты современного общества, такие как банки, службы управления и мониторинга, торговые и производственные предприятия сами по себе имеют распределенную природу и нуждаются в инфраструктуре, позволяющей организовать корпоративное и межкорпоративное взаимодействие на основе распределенных программных приложений.

Grid опирается и развивает традиционные технологии Интернет, однако впервые серьезно ставится вопрос о гарантированном качестве обслуживания. В этом плане можно говорить о Grid, как об Интернете следующего поколения. Так ли это будет, неизвестно, тем не менее, определенно можно сказать, что появляется новая сфера деятельности для специалистов в области программирования.

6.4. Архитектура центра обработки данных

Конкретный вариант построения АИС определяется её архитектурой. В связи с тем, что АИС является достаточно сложным образованием, при описании ее построения пользуются рядом

структур, отличающихся типами элементов и связей между ними (табл. 6.1).

Таблица 6.1

Используемые при описании АИС виды структур

Вид структуры	Тип элементов	Тип связей
Функциональная	Функции и задачи	Информационные
Техническая	Технические средства, устройства и узлы	Электрические
Информационная	Единицы информации	Процедуры и операции преобразования информации
Программная	Программные компоненты, программные компоненты и программы	Управляющие и информационные
Организационная	Организационно-штатные единицы должностных лиц	Отношения подчинения и взаимодействия

Ограничимся рассмотрением технической структуры АИС как наиболее употребительной при ее анализе и синтезе.

Основными элементами АИС являются центры обработки данных (ЦОД) и телекоммуникационная сеть передачи данных (ТКСПД).

Под ЦОД понимается совокупность функционально и конструктивно связанных технических, программных и информационно-лингвистических компонентов, принадлежащая одному объекту АИС и предназначенная для автоматизированного выполнения заданных функций управления на объекте. ЦОД обычно представляет собой промышленное изделие.

Объект АИС определяется как совокупность помещений или транспортных средств на участке местности с размещенным в них ЦОД.

В системах с иерархическим управлением различают ЦОД головных, промежуточных и низовых объектов (рис. 6.3).

Большая часть технических средств автоматизации, из которых строятся ЦОД (рис. 6.4), конструктивно объединяется в *технологические комплексы*:

- электронно-вычислительные комплексы (серверы, машины) (ВК, ЭВМ),
- комплексы средств общения пользователей с ВК (автоматизированные рабочие места – АРМ, выполненные на базе персональных ЭВМ или терминалов),
- комплексы средств передачи данных (КСПД),
- комплексы средств фиксации и сбора первичной информации (АДИ),
- комплексы средств ведения единого времени (КСВЕВ),
- комплекс средств технического обслуживания и ремонта (КСТОР),
- комплекс средств обучения и тренажа должностных лиц обслуживающего и оперативного состава (КСОТ),

– комплексы средств жизнеобеспечения (КСЖО).

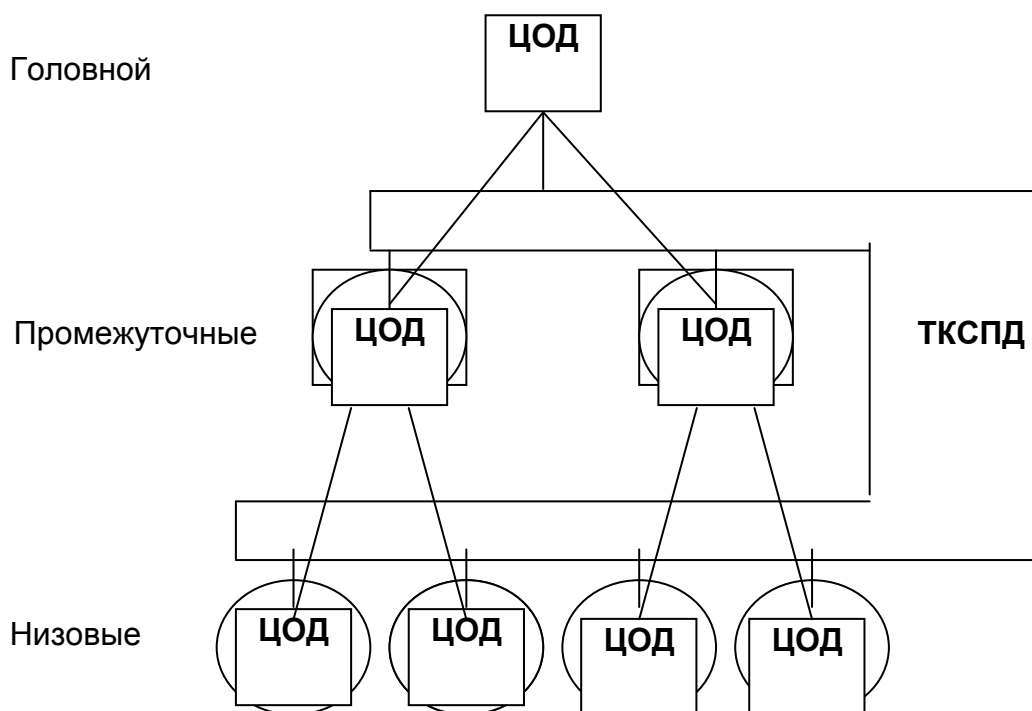


Рис. 6.3. Техническая структура информационной системы

Вычислительные комплексы, машины, серверы обеспечивают:

- создание и ведение информационной базы,
- решение информационных и расчетных задач,
- контроль и восстановление процессов преобразования данных.

В состав ВК включается две-три ЭВМ или сервера с периферийными устройствами. Могут использоваться и отдельные ЭВМ, серверы.

Комплексы средств общения предоставляют пользователям как источникам и потребителям информации следующие возможности:

- ввод данных в ВК,
- прием данных из ВК,
- хранение некоторой части данных,
- решение несложных задач,
- документирование данных.

Оборудование АРМ может включать полный или сокращенный набор средств согласно перечисленным возможностям.

Комплексы средств передачи данных (КСПД) обеспечивают передачу данных от источника (отправителя) к потребителю (получателю). КСПД отправителей и получателей данных соединяются друг с другом с использованием выделенных и/или коммутируемых каналов связи непосредственно или через центры коммутации, образуя телекоммуникационную сеть передачи данных (ТКСПД).

Комплексы средств фиксации и сбора первичной информации, на основе автоматических датчиков информации (АДИ), фиксируют и

собирают информацию в местах ее возникновения, а также представляют ее в виде, допускающем передачу. Примеры АДИ: датчики заражения местности веществами, датчики доступа в помещения, датчики фиксации изготовленных изделий и др.

Комплексы средств ведения единого времени (КСВЕВ) осуществляют:

- формирование сигналов единого времени заданной периодичности и принудительную выдачу их абонентам,
- отсчет абсолютного астрономического времени и выдачу его абонентам принудительно или по требованию,
- индикацию текущего астрономического времени.

В состав комплекса средств ведения единого времени обычно включается аппаратура формирования временных сигналов, выносные индикаторы времени и радиоприемник.

Комплексы средств технического обслуживания и ремонта (КСТОР) призваны поддерживать работоспособное состояние остальных комплексов АИС.

Комплексы средств обучения и тренажа (КСОТ) используются при обучении пользователей и персонала службы эксплуатации и проведении с ними тренировок.

Комплексы средств жизнеобеспечения (КСЖО) создают необходимые для работы пользователей и персонала условия. Элементами КСЖО выступают светильники, кондиционеры, отопительные устройства, фильтровентиляционные установки и некоторое другое оборудование.

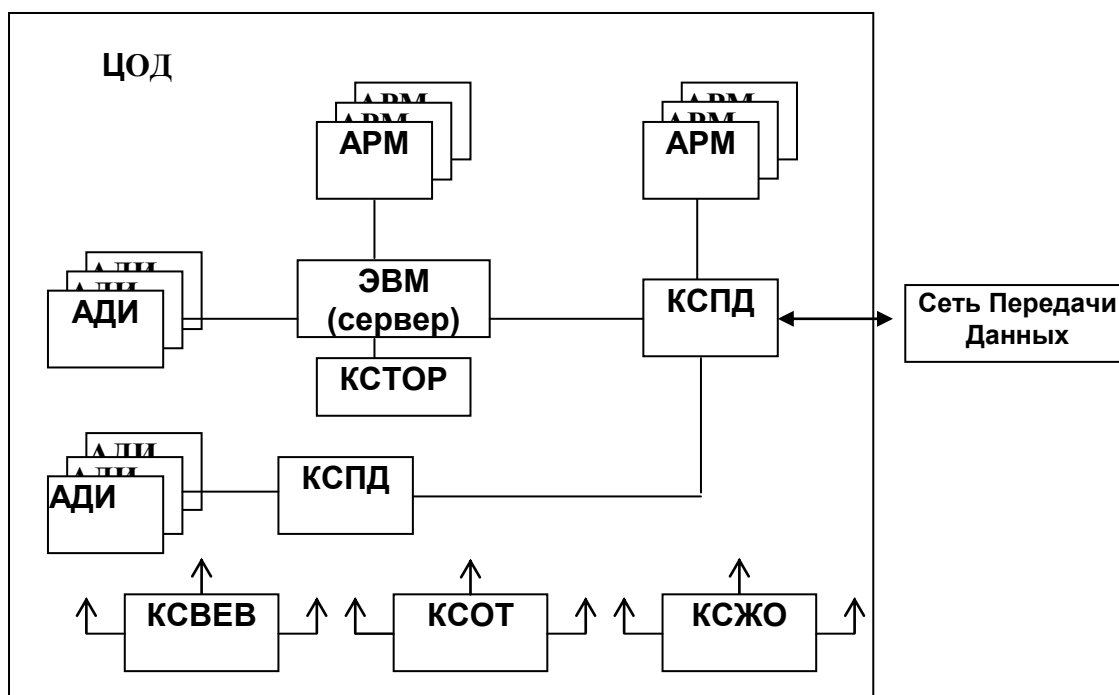


Рис. 6.4. Обобщенная архитектура центра обработки данных

Выводы

1. Распределенная информационная система является средой, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технические и программные средства связи и т.д. Основная цель информационной системы – организация хранения и передачи информации.
2. Основными элементами распределенных информационных систем являются базы данных (БД). По характеру своего размещения БД могут быть локальными или распределенными.
3. В настоящее время наибольшее распространение получили реляционные базы данных. Главными достоинствами реляционных БД являются:
 - простота представления данных;
 - простота внесения изменений в базу данных;
 - упрощение процедур разграничения доступа к данным в разных таблицах;
 - простота физической реализации двумерных таблиц и др.
4. Управление элементами распределенных информационных систем осуществляется с помощью специальных систем управления. Система управления базами данных (СУБД) – специализированная программа (или комплекс программ), предназначенная для манипулирования базой данных. Подавляющее большинство СУБД для персональных компьютеров поддерживает реляционную модель данных.
5. Основными элементами автоматизированной информационной системы (АИС) являются центры обработки данных (ЦОД) и телекоммуникационная сеть передачи данных (ТКСПД). Под ЦОД понимается совокупность функционально и конструктивно связанных технических, программных и информационно-лингвистических компонентов, принадлежащая одному объекту АИС и предназначенная для автоматизированного выполнения заданных функций управления на объекте.
6. Для повышения производительности элементов распределенной информационной системы используют распределенные вычисления.

Литература

1. Новые информационные технологии в науке и образовании: Уч. пособие / Под ред. И.Б. Саенко – СПб.: ВАС, 2007.