

Системы обработки многopользовательских баз данных

Вопросы лекции:

- Технологии обработки данных.
- Сетевые технологии обработки данных.
- Технология клиент-сервер.
- Распределенные БД.
- Администрирование БД – защита данных с использованием паролей, шифрования, восстановление.

Технологии обработки данных

- БД могут быть **однопользовательскими** и **многопользовательскими**.
- В первом случае БД создается и обрабатывается на одном компьютере, во втором случае БД может храниться на сервере, а каждый пользователь сети со своего персонального компьютера или автоматизированного рабочего места (АРМ) получает доступ к общей информационной базе.

Технологии обработки данных

Создание БД в сети существенно

***увеличивает эффективность
информационных систем***

банков, бирж, инвестиционных
фондов и других структур рыночной
ЭКОНОМИКИ

Корпоративные (или intranet) сети являются сетями уровня предприятия, которые можно подключать к сети Internet. Они позволяют пользователю получить доступ к информации непосредственно со своего рабочего места.



Эти сети базируются на клиент-серверных технологиях

Как правило, компьютеры в сети не являются равноправными.
Те из них, которые владеют и управляют тем или иным ресурсом
называют СЕРВЕРОМ,
а компьютеры, которые используют этот ресурс, – КЛИЕНТОМ.

Конкретный сервер определяется видом ресурса, которым он
владеет

Так, если ресурсом являются
базы данных,
то речь идет о сервере баз данных,
назначение которого –
обслуживать запросы клиентов,
связанные обработкой данных.



Если ресурсом является
файловая система,
то говорят
о файловом сервере
или о файл-сервере.

Файл_1
Файл_2
Файл_3 ...

Этот же принцип распространяется и на взаимодействие программ.

Так, ядро реляционной SQL-ориентированной СУБД часто
называют сервером базы данных или SQL-сервером,
а программу, обращающуюся к нему за услугами
по обработке данных – SQL-клиентом.

Сетевые технологии обработки данных

- В зависимости от конфигурации технических и программных средств существуют **различные концепции** сетевой обработки данных.
- Задача большинства многопользовательских систем – позволить каждому пользователю работать с системой БД как с однопользовательской. Реализация такой возможности зависит от **используемых технологий при организации многопользовательской** обработки данных.

Сетевые технологии обработки данных

Технологии можно классифицировать следующим образом:

- **локальное** функционирование рабочих мест (с использованием локальной вычислительной сети (ЛВС) или без ЛВС, т.е. автономная работа компьютеров);
- **полностью централизованная обработка данных** (одна большая ЭВМ, обслуживающая несколько терминалов, которые сами не производят никаких вычислений, а только отражают данные и позволяют вводить информацию);
- **технология файл-сервер**;
- **технология клиент-сервер**.

Третья и четвертая технологии реализуются на базе локальных вычислительных сетей или глобальных вычислительных сетей.

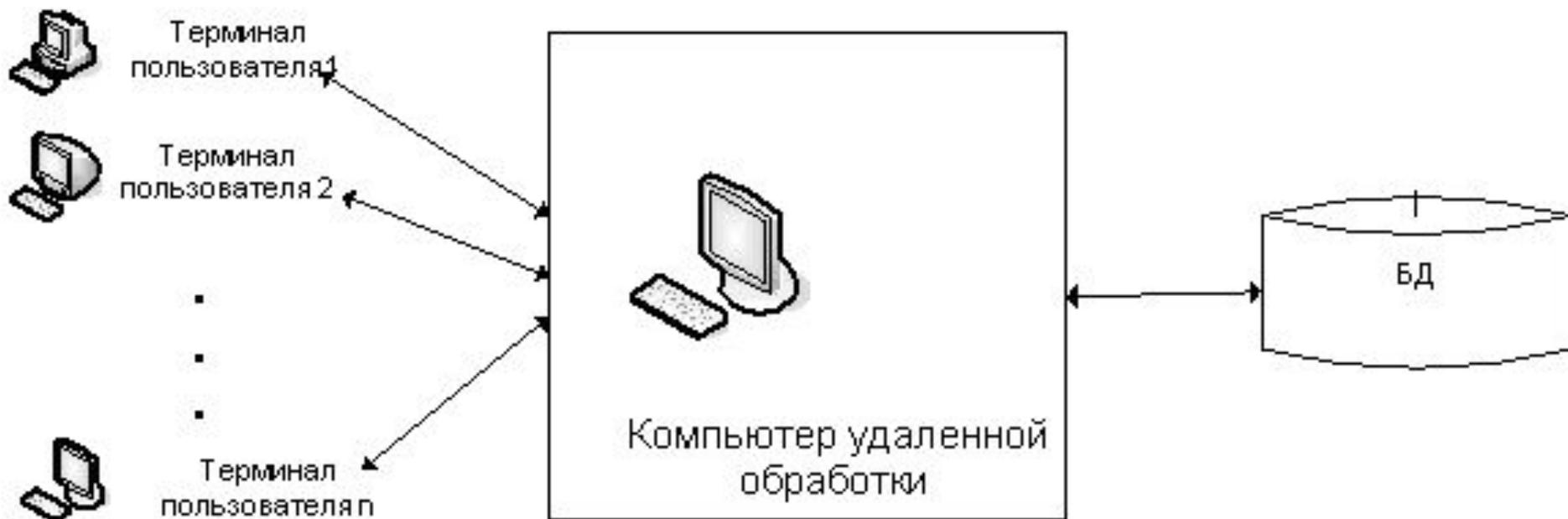
Рассматривая взаимодействие разных участников процесса обработки данных можно выделить несколько компонентов:

- СУБД
- База данных
- Прикладные программы (приложения)
- Сервер
- Клиент

В зависимости от того, как именно эти компоненты взаимодействуют друг с другом, выделяют различные модели обработки данных.

Сетевые технологии обработки данных

- **Централизованная обработка данных (удаленная обработка)**



Централизованная архитектура

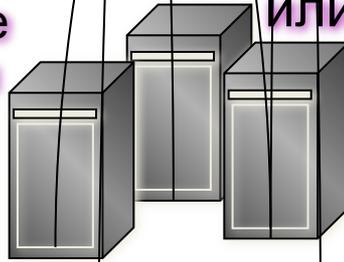
Первоначально СУБД имели централизованную архитектуру.

Здесь располагались:

СУБД,
прикладные
программы



Большая ЭВМ
или мини-ЭВМ



Согласно этому сама СУБД и прикладные программы, которые работали с базами данных, функционировали на центральном компьютере (большая ЭВМ, Mainframe, мини-ЭВМ и т.д.)

Там же располагались базы данных.

Сетевые технологии обработки данных

Централизованная обработка данных (удаленная обработка)

- Пользователи обрабатывают данные в пакетном режиме. Интерактивный режим доступа осуществляется с помощью **терминалов**, которые не обладают собственными вычислительными ресурсами. Программы управления коммуникациями (связью), прикладные программы, СУБД и ОС работают на **едином центральном компьютере**. Поскольку вся обработка производится единственным компьютером, то пользовательский интерфейс систем удаленной обработки обычно достаточно прост.

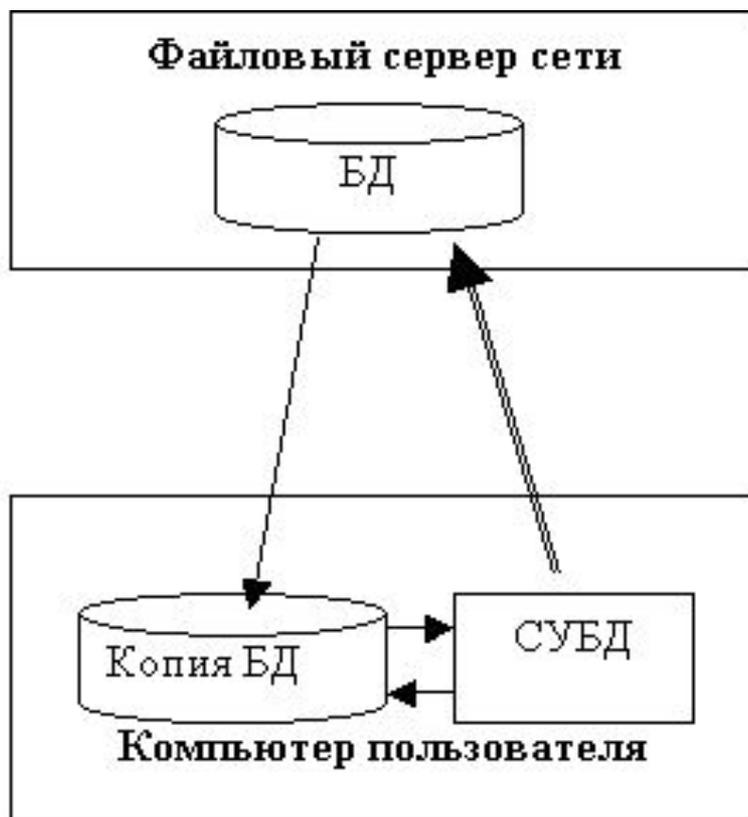
Сетевые технологии обработки данных

Централизованная обработка данных (удаленная обработка)

- Преимуществом такой обработки является возможность коллективного использования ресурсов и оборудования, централизованное хранение данных, а недостатком – отсутствие персонализации рабочей среды (все программное обеспечение хранится централизованно и используется коллективно). Исторически системы удаленной обработки были наиболее распространенной альтернативой многопользовательским системам баз данных.

Сетевые технологии обработки данных

- **Технология «файл-сервер»**



Сетевые технологии обработки данных

Технология «файл-сервер»

- При данной технологии информация сосредоточивается на различных рабочих станциях и сетевое программное обеспечение занято только передачей данных, не различая, нужна ли вся информация по запросу или только ее часть. На клиентском рабочем месте, с которого осуществляется запрос, происходит просмотр информации, выбор нужной и ее обработка. Далее измененная информация может передаваться в обратном направлении для сохранения в БД. Когда запросы происходят одновременно, каналы передачи данных перегружаются, и работа в локальной сети парализуется, что может повлечь за собой и искажения данных в БД.

Модель файлового сервера (File Server – FS)

Основные черты данной технологии следующие:



БД располагается на компьютере, который является файловым сервером.

СУБД, например ¹Access, может ²быть установлена или на файловом сервере или на ³каждой рабочей станции, но выполняется она всегда на рабочей станции пользователя.

Сеть обеспечивает аппаратную и программную поддержку обмена данными между компьютерами.

Сетевые технологии обработки данных

Технология «файл-сервер»

- Для этой архитектуры характерен коллективный доступ к общей БД на сервере, который является файловым сервером.
- Файловый сервер содержит файлы, необходимые для работы приложений и самой СУБД.
- Пользовательские приложения и сама сетевая СУБД размещены и функционируют на отдельных рабочих станциях и обращаются к файловому серверу по мере необходимости.

Достоинства:

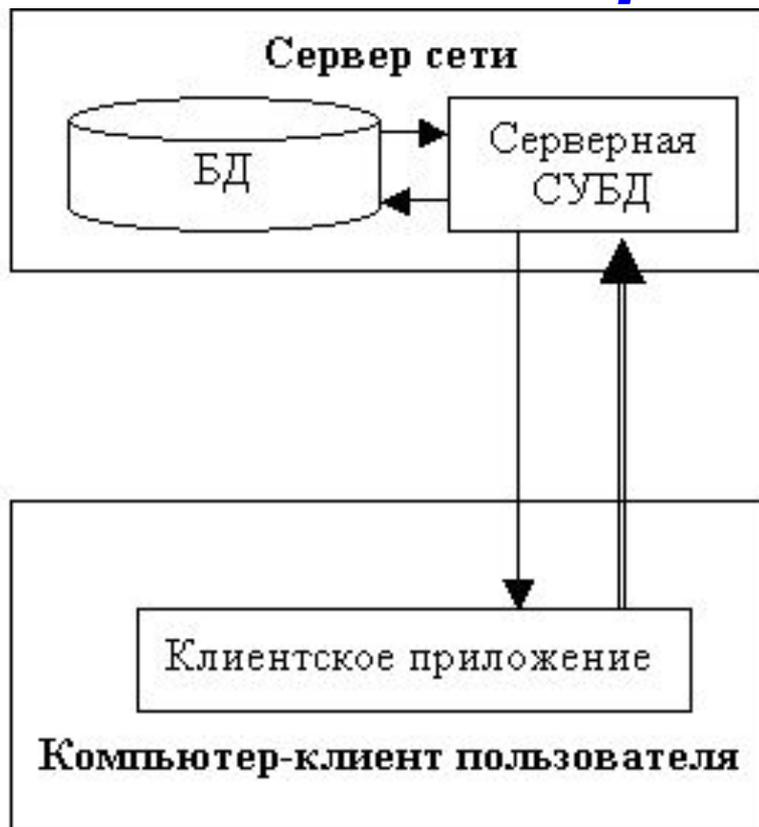
- низкая стоимость и высокая скорость разработки**
- невысокая стоимость обновления и изменения ПО**

Недостатки:

- большой сетевой трафик (полные копии БД перемещаются по сети с сервера на комп. клиента)**
 - снижение производительности при обработке больших объемов информации**
 - выполнение запроса к БД и управление целостностью осуществления на рабочей станции**
 - сложность поддержки целостности и восстановления БД на сервере**
 - на каждой рабочей станции должна находиться сама сетевая версия настольной СУБД, что требует наличия больших объемов операционной системы на комп. пользователя.**
- Файлово-серверная технология перемещается исключительно при работе с небольшим объемом данных, т.к. в противном случае могут наблюдаться большие задержки работы сети и пользователей компьютеров.**

Сетевые технологии обработки данных

- **Технология «клиент-сервер»**



ТЕХНОЛОГИЯ КЛИЕНТ-СЕРВЕР

На обеспечение эффективной работы большого числа пользователей с общей базой данных ориентирована *технология Клиент - Сервер*.

Согласно этой технологии пользователь-клиент передает со своего компьютера **запрос на машину** сервера, там СУБД обрабатывает запрос и обратно посылает только **результат выполнения запроса**.



Таким образом, значительно снижается объем данных, передаваемых по сети.

Приложение пользователя разрабатывается и выполняется под управлением СУБД, например Access, на машине клиента.

Общая БД размещается на мощном компьютере, где функционирует СУБД сервера баз данных.

СУБД сервера баз данных выполняет:

- Обработку данных, размещенных на сервере,*
- Отвечает за их целостность и сохранность.*

Для управления базой данных на сервере используется язык SQL – язык структурированных запросов.

Широко известны такие серверы баз данных, как SQL-Server фирмы Microsoft, Oracle Server фирмы Oracle и Netware SQL-Server фирмы Novell.

Сетевые технологии обработки данных

Технология «клиент-сервер»

- Является более прогрессивной технологией. При этой технологии СУБД делится на две части: клиентскую и серверную.
- Когда клиенту требуются данные, он посылает серверу запрос, в котором формулируется, какая именно информация необходима. Серверное программное обеспечение выбирает данные из БД и пересылает только то, что необходимо клиенту; таким образом, лишняя информация по каналам связи не передается. Благодаря этому увеличивается общая производительность системы.

Функции клиентского приложения разбиваются на следующие группы:

ввод-вывод данных (презентационная логика) – это часть кода клиентского приложения, которая определяет, что пользователь видит на экране, когда работает с приложением;

бизнес-логика – это часть кода клиентского приложения, которая определяет алгоритм решения конкретных задач приложения;

обработка данных внутри приложения (логика базы данных) – это часть кода клиентского приложения, которая связывает данные сервера с приложением. Для этой связи используется процедурный язык запросов SQL, с помощью которого осуществляется выборка и модификация данных в серверных СУБД.

Сервер баз данных в общем случае осуществляет целый комплекс действий по управлению данными. Основными среди них являются следующие:

- выполнение пользовательских запросов на выбор и модификацию данных и метаданных, получаемых от клиентских приложений, функционирующих на ПК локальной сети;
- хранение и резервное копирование данных;
- поддержка ссылочной целостности данных согласно определенным в БД правилам;
- обеспечение авторизованного доступа к данным на основе проверки прав и привилегий пользователя;
- протоколирование операций и ведение журнала транзакций.

Преимущества клиент/серверной обработки:

уменьшается сетевой трафик, так как через сеть передаются только результаты запросов.

груз файловых операций ложится в основном на сервер, который мощнее компьютеров-клиентов и поэтому способен быстрее обслуживать запросы. Как следствие этого, уменьшается потребность клиентских приложений в оперативной памяти. поскольку серверы способны хранить большое количество данных, то на компьютерах-клиентах освобождается значительный объем дискового пространства для других приложений.

повышается уровень непротиворечивости данных и существенно повышается степень безопасности БД, так как правила целостности данных определяются в серверной СУБД и являются едиными для всех приложений, использующих эту БД.

имеется возможность хранения бизнес-правил (например, правил ссылочной целостности или ограничений на значения данных) на сервере, что позволяет избежать дублирования кода в различных клиентских приложениях, использующих общую базу данных.

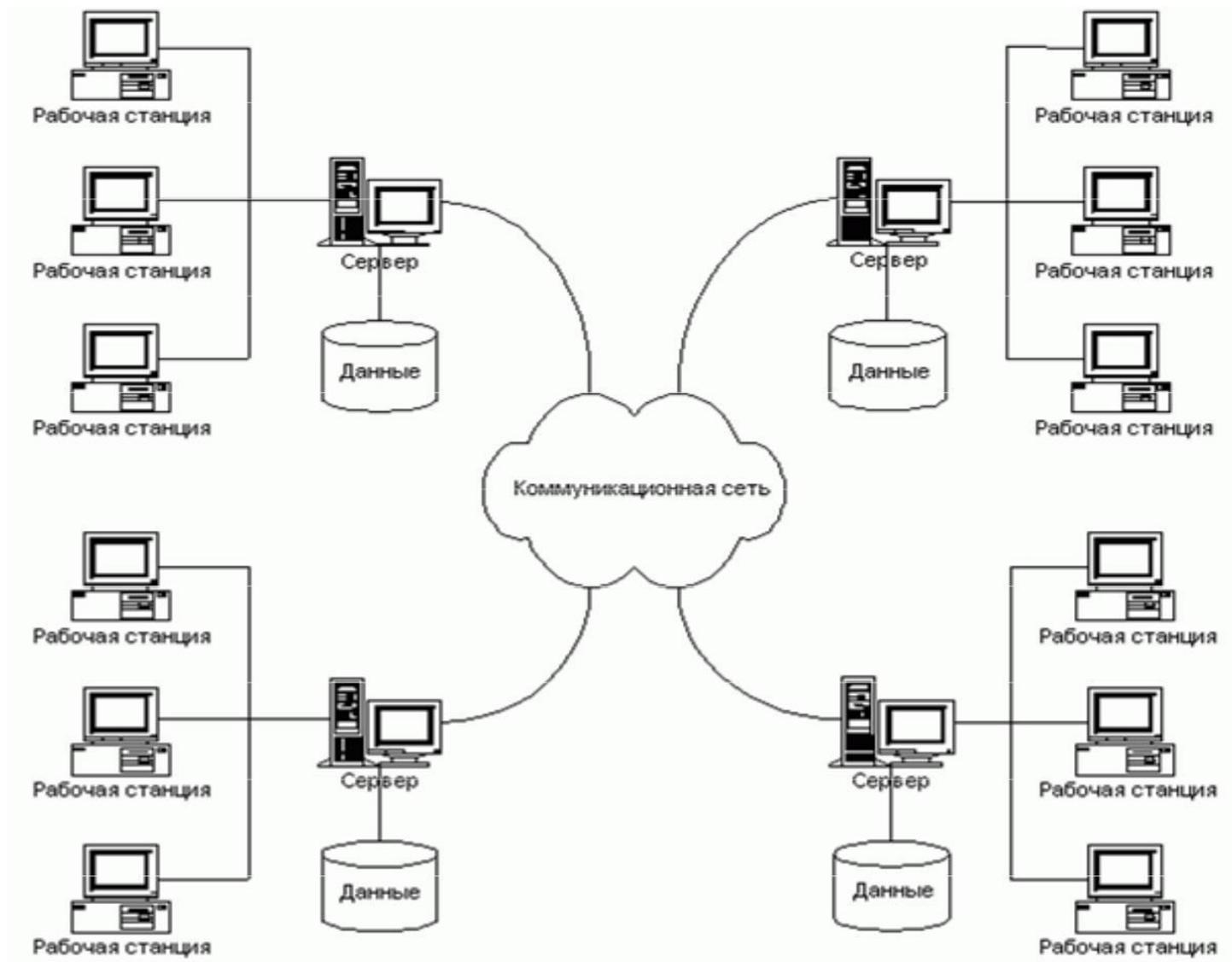
Сетевые технологии обработки данных

Технология «клиент-сервер» может быть реализована по следующим моделям: модель **«толстого клиента»** и модель **«тонкого клиента»**.

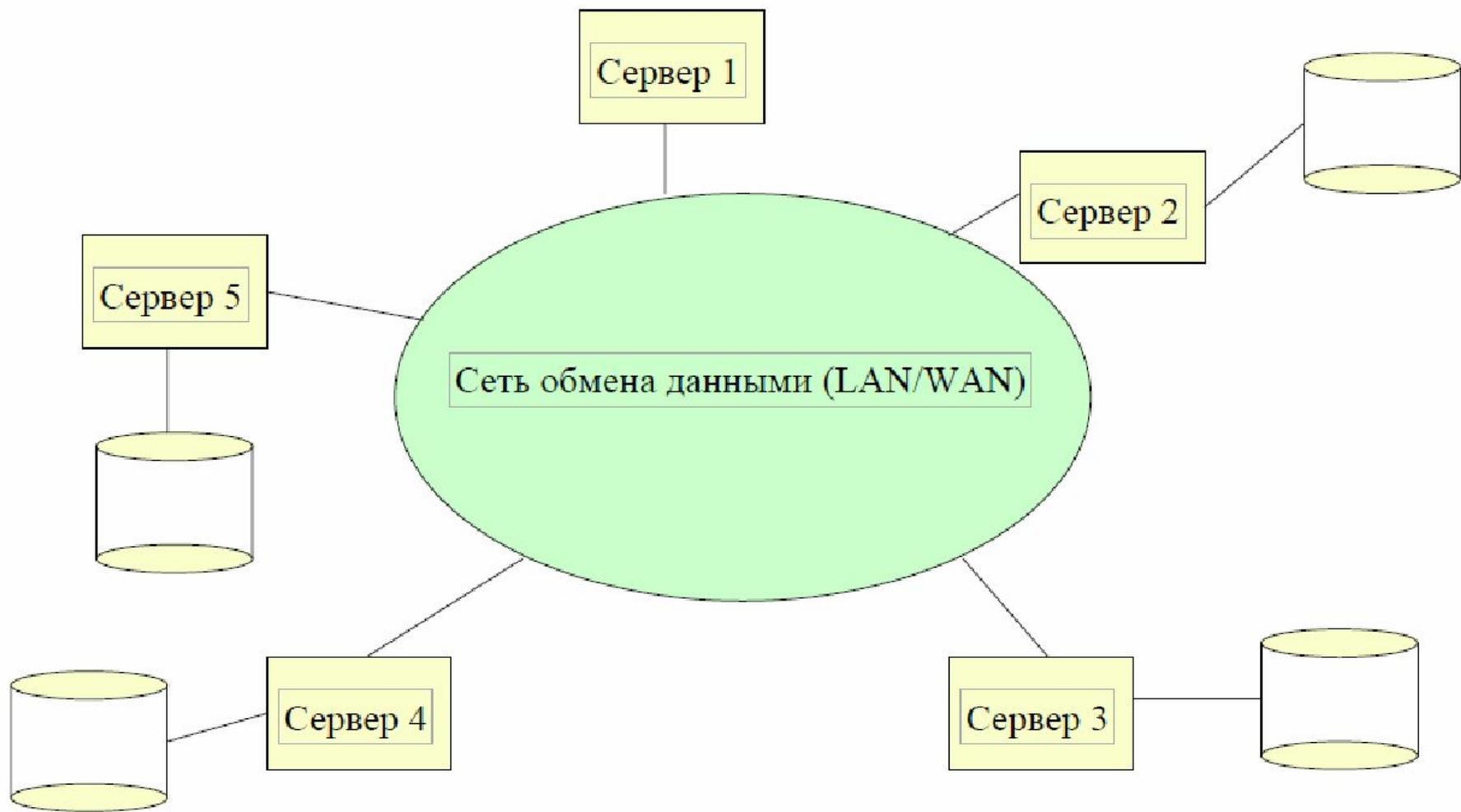
- Модель **«толстого клиента»**. Сервер осуществляет доступ к данным, а клиентский компьютер выполняет вычисления и пересылает данные назад для сохранения в общей БД. Недостаток данной модели заключается в том, что могут пересылаться большие объемы данных только для того, чтобы выполнить пару вычислений. Эта модель наиболее удобна для поисковых систем, и менее – для систем, связанных с расчетами.
- Модель **«тонкого клиента»** характеризуется тем, что значительная часть обработки данных выполняется сервером, в том числе и некоторые вычисления. Таким образом, устраняется непроизводительная пересылка данных, но к серверу предъявляются высокие требования по производительности.

Распределенные БД

- **Распределенная БД (РаБД)** – набор логически связанных между собой разделяемых данных и их описаний, которые физически распределены по нескольким компьютерам (узлам) в некоторой компьютерной сети.
- Каждая таблица в РаБД может быть разделена на некоторое количество частей, называемых **фрагментами**. Фрагменты могут быть **горизонтальными**, **вертикальными** и **смешанными**. Горизонтальные фрагменты представляют собой подмножества строк, а вертикальные – подмножества столбцов. Фрагменты распределяются на одном или нескольких узлах.



Распределённые базы в локальной сети



Распределенные БД

С целью улучшения доступности данных и повышения производительности системы для отдельных фрагментов может быть организована **репликация** – поддержка актуальной копии некоторого фрагмента на нескольких различных узлах. **Репликаты** – множество различных физических копий некоторого объекта БД, для которых в соответствии с определенными в БД правилами поддерживается синхронизация с некоторой «главной копией».

Распределенные БД

- Существуют несколько альтернативных стратегий размещения данных в системе: ***раздельное*** (фрагментированное) ***размещение***, ***размещение с полной репликацией*** и ***размещение с выборочной репликацией***.

Распределенные БД

- **Раздельное (фрагментированное) размещение.** В этом случае БД разбивается на непересекающиеся фрагменты, каждый из которых размещается на одном из узлов системы. При отсутствии репликации стоимость хранения данных будет минимальна, но при этом будет невысок также уровень надежности и доступности данных в системе. Отказ на любом из узлов вызовет утрату доступа только к той части данных, которая на нем хранилась.

Распределенные БД

- ***Размещение с полной репликацией.***

Эта стратегия предусматривает размещение полной копии всей БД на каждом из узлов системы.

Следовательно, надежность и доступность данных, а также уровень производительности системы будут максимальными. Однако стоимость хранения данных и уровень затрат на передачу данных в этом случае будут самыми высокими.

Распределенные БД

- **Размещение с выборочной репликацией.** Данная стратегия представляет собой комбинацию методов фрагментации, репликации и централизации. Одни массивы данных разделяются на фрагменты, тогда как другие подвергаются репликации. Все остальные данные хранятся централизованно. Целью применения данного метода является объединение всех преимуществ, существующих в остальных моделях, с одновременным исключением собственных им недостатков. Благодаря своей гибкости, именно эта стратегия используется чаще всего.

Распределенные СУБД

- Работу с РаБД обеспечивают распределенные СУБД.
- **Распределенная СУБД (РаСУБД)** – комплекс программ, предназначенный для управления распределенной БД и позволяющий сделать распределенность информации «прозрачной» для конечного пользователя.
- Основная задача РаСУБД состоит в обеспечении средств интеграции локальных баз данных, располагающихся в некоторых узлах компьютерной сети, с тем, чтобы пользователь, работающий в любом узле сети, имел доступ ко всем этим БД как к единой БД.

Распределенные СУБД

Гомогенные и гетерогенные распределенные БД

- РаБД можно классифицировать на **гомогенные** и **гетерогенные**.
- Гомогенной РаБД управляет один и тот же тип СУБД. Гетерогенной РаБД управляют различные типы СУБД, использующие разные модели данных – реляционные, сетевые, иерархические или объектно-ориентированные СУБД.
- Гомогенные РаБД значительно проще проектировать и сопровождать. Кроме того, подобный подход позволяет поэтапно наращивать размеры РаБД, последовательно добавляя новые узлы к уже существующей РаБД. Гетерогенные РаБД обычно возникают в тех случаях, когда независимые узлы, управляемые своей собственной СУБД, интегрируются во вновь создаваемую РаБД.

Определение идеальной ДДВ Криса

Дефта

1. Локальная автономия (local autonomy)
2. Независимость узлов (no reliance on central site)
3. Непрерывные операции (continuous operation)
4. Прозрачность расположения (location independence)
5. Прозрачная фрагментация (fragmentation independence)
6. Прозрачное тиражирование (replication independence)
7. Обработка распределенных запросов (distributed query processing)
8. Обработка распределенных транзакций (distributed transaction processing)
9. Независимость от оборудования (hardware independence)
0. Независимость от операционных систем (operating system independence)
1. Прозрачность сети (network independence)
2. Независимость от баз данных (database independence)

Распределенные СУБД

Преимущества и недостатки РаСУБД

Системы с распределенными БД имеют дополнительные преимущества перед традиционными централизованными системами баз данных.

Преимущества РаСУБД:

1. Отражение структуры организации.
2. Разделяемость и локальная автономность.
3. Повышение доступности данных.
4. Повышение надежности.
5. Повышение производительности.
6. Экономические выгоды.
7. Модульность системы.

Распределенные СУБД

Недостатки РаСУБД:

- **Повышение сложности.** РаСУБД являются более сложными программными комплексами, чем централизованные СУБД,
- **Увеличение стоимости.** Увеличение сложности означает и увеличение затрат на приобретение и сопровождение РаСУБД.
- **Проблемы защиты.** В централизованных системах доступ к данным легко контролируется. Однако в распределенных системах требуется организовать контроль доступа не только к данным, реплицируемым на несколько различных узлов, но и защиту сетевых соединений.
- **Усложнение контроля за целостностью данных.** В РаСУБД повышенная стоимость передачи и обработки данных может препятствовать организации эффективной защиты от нарушений целостности данных.
- **Отсутствие стандартов.** Отсутствуют стандарты на каналы связи и протоколы доступа к данным, а также отсутствуют инструментальные средства и методологии, способные помочь пользователям в преобразовании централизованных систем в распределенные.
- **Недостаток опыта.** Еще не накоплен необходимый опыт промышленной эксплуатации распределенных систем, сравнимый с опытом эксплуатации централизованных систем.

Администрирование БД

- По мере того как деятельность организаций всё больше зависит от компьютерных информационных технологий, проблемы защиты баз данных становятся всё более актуальными. Угрозы потери конфиденциальной информации стали обычным явлением в современном компьютерном мире. Если в системе защите есть недостатки, то данным может быть нанесен ущерб, который может быть выражен в: нарушении целостности данных, потере важной информации, попадании важных данных посторонним лицам и т.д.

Администратор - лицо, ответственное за целостность и непротиворечивость данных в системе, безопасность системы, эффективность функционирования системы и использования ею ресурсов. СУБД (система управления баз данных) "видит" администратора как пользователя, обладающего определенным набором привилегий. Привилегии администратора дают ему возможность использовать такие команды и утилиты СУБД и иметь доступ к таким системным таблицам, которые недоступны рядовым пользователям. Как правило, СУБД предоставляют в распоряжение администратора еще и специальный инструментарий, который обеспечивает удобный интерфейс для выполнения функций администратора.

функции администратора:

- установка СУБД;
- управление памятью;
- управление разделением данных между пользователями;
- копирование и восстановление БД;
- управление безопасностью в системе;
- передача данных между СУБД и другими системами;
- управление производительностью.

Защита данных с использованием паролей

- Чтобы обеспечить защиту данных в компьютерных системах необходимо определить перечень мер, обеспечивающих защиту. Основными методами защиты баз данных являются защита **паролем, шифрование, разграничение прав доступа**
- Защита паролем представляет собой простой и эффективный способ защиты БД от несанкционированного доступа. Пароли устанавливаются пользователями или администраторами БД. Учет и хранение паролей выполняется самой СУБД. Обычно, пароли хранятся в определенных системных файлах СУБД в зашифрованном виде. После ввода пароля пользователю СУБД предоставляются все возможности по работе с БД.

Защита данных с использованием шифрования

- Более мощным средством защиты данных от просмотра является их шифрование. **Шифрование** – это преобразование читаемого текста в нечитаемый текст, при помощи некоторого алгоритма; применяется для защиты уязвимых данных.

Разграничение прав доступа

- В целях контроля использования основных ресурсов СУБД во многих системах имеются средства **установления прав доступа** к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта (пользователь, создавший объект), а также администратор БД имеют все права. Остальные пользователи к разным объектам могут иметь различные уровни доступа. Разрешение на доступ к конкретным объектам базы данных сохраняется в файле рабочей группы.

Восстановление БД

- Поскольку данные, хранимые компьютерными средствами подвержены потерям и повреждениям, вызываемым разными событиями, важно обеспечить средства восстановления данных. Приведение базы данных точно в то состояние, которое существовало перед отказом не всегда возможно, но процедуры восстановления базы данных могут привести ее в состояние, существовавшее незадолго до отказа.
- Для обеспечения защиты БД, используют различные ограничения целостности данных, журнализацию, репликацию и резервное копирование

1. Резервное копирование базы данных

С целью обеспечения достоверности и постоянной работоспособности БД периодически вручную или автоматически осуществляется копирование базы данных. Основными причинами, побуждающими выполнение процедур копирования, являются различные структурные изменения базы данных:

- создание или удаление табличного пространства;
- добавление или переименование (перемещение) файла данных в существующем табличном пространстве;
- добавление, переименование (перемещение) или удаление журнала повторения и др.

2. Репликация

Репликацией БД называют создание специальных копий общей БД – *реплик*, с которыми пользователи могут одновременно работать на разных рабочих станциях. Пользователи могут вносить изменения в реплики, а потом обновлять общую БД, внося в неё изменения сделанные в репликах, то есть синхронизировать общую БД.

3. Журнализация изменений –

Это функция СУБД, которая сохраняет информацию, необходимую для восстановления базы данных в предыдущее состояние в случае логических или физических отказов.

Транзакция — это неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными (чтения, удаления, вставки, модификации).

В простейшем случае журнализация изменений заключается в последовательной записи во внешнюю память всех изменений, выполняемых в базе данных. Записывается следующая информация: порядковый номер, тип и время изменения; вид транзакции; объект, подвергшийся изменению (номер хранимого файла и номер блока данных в нём, номер строки внутри блока); предыдущее состояние объекта и новое состояние объекта.

Формируемая таким образом информация называется журнал изменений базы данных.

Характеристики серверов баз данных.

Современные серверные СУБД:

существуют в нескольких версиях для различных платформ, как правило, для различных коммерческих версий UNIX – Solaris, HP/UX. Многие производители также выпускают версии своих серверов баз данных для Windows NT Workstation Windows 95/98, а также версии для Linux;

в большинстве случаев поставляются с удобными административными утилитами; осуществляют резервное копирование и архивацию данных и журналов транзакций; поддерживают несколько сценариев репликаций; позволяют осуществлять параллельную обработку данных в многопроцессорных системах. Серверы, допускающие параллельную обработку, разрешают нескольким процессорам обращаться к одной БД, что обеспечивает высокую скорость обработки транзакций;

поддерживают создание хранилищ данных и OLAP. Хранилище данных – это совокупность данных, полученных прямо или косвенно их информационных систем, которые содержат текущую и деловую информацию, а также из некоторых внешних источников.

выполняют распределенные запросы и транзакции; дают возможность использовать различные средства проектирования схем данных – универсальные или ориентированные на конкретную СУБД; имеют средства разработки клиентских приложений и генераторы отчетов; поддерживают публикацию баз данных в Интернет; обладают широкими возможностями управления пользовательскими привилегиями и правами доступа к различным объектам БД.

Механизмы доступа к данным базы на сервере.

Все серверные СУБД имеют клиентскую часть, которая обращается к БД посредством СУБД. Между клиентским прилож. и СУБД не существует прямой связи и дополнительно встраиваются программные модули, позволяющие клиентскому приложению получать доступ к БД, создаваемым с помощью разных СУБД. Такие модули называются механизмами доступа к данным. Существует 2 основных способа доступа к данным из клиентских приложений: использование прикладного интерфейса и использование универсального программного интерфейса. Универсальный механизм доступа к данным обеспечивает возможность использования одного и того же интерфейса для доступа к разным типам СУБД. Обычно он реализован в виде специальных дополнительных модулей, называемых драйверами. Наиболее распространенным программным интерфейсом, обеспечивающим доступ к данным конкретной базы данных является ODBC фирмы Microsoft. Для доступа к данной конкретн. СУБД, кроме клиентск. части необходимо приложение ODBC и драйвер.

Прикладной программный интерфейс (API) предст. соб. набор функций, вызываемых из клиентского приложения. Он может работать только с СУБД данного производителя и при ее замене придется переписывать значительную часть кода клиентского приложения.

Универсальный механизм доступа к данным обеспечивает возможность использования одного и того же интерфейса для доступа к разным типам СУБД. Обычно он реализован в виде специальных дополнительных модулей, называемых драйверами. Наиболее распространенным программным интерфейсом, обеспечивающим доступ к данным конкретной базы данных является ODBC фирмы Microsoft. Для доступа к данной конкретн. СУБД, кроме клиентск. части необходимо приложение ODBC и драйвер.

ODBC – открытый стандарт совместимости БД, разработанный в 1990-х для предоставления независимого от СУБД способа обработки информации из реляц БД. ODBC – интерфейс, с помощью которого прикладн проги могу обращ-ся к БД и обраб-ть независ от СУБД способом. ODBC-драйвер выполняет все вызовы ODBC-функций и «переводит» их на язык источника данных. СУБД хранит и выводит данные в ответ на запросы со стороны ODBC-драйвера. Приложение ODBC – для определения доступа источника данных для конкрет. компа и описания источн. данных. Задание ODBC-источника данных является действием, которое осуществляется средствами операционной системы, управляющей компьютером. Источник данных – структура дан. ODBC, идентифицирующая БД и СУБД, которая ее обрабатывает.

С его помощью могут быть заданы: *1.пользовательский* – ист. дан., доступный только текущему пользователю на текущем компьютере; *2.файловый* – фал, кот. может совместно использоваться пользователем БД; *3.системный* – источник данных, доступный всем пользователям и службам текущего компьютера.

Преимущества: - простота разработки приложения; - позволяет создавать распределенные гетерогенные приложения без учета конкретной СУБД – приложения становятся независимыми от СУБД. Недостатки: - снижение скорости доступа к данным; - увеличение времени обработки запросов; - предварит инсталляция и настройка ODBC на каждом рабочем месте; - представляет доступ только к SQL-ориентированным БД.

OLEDB и ADO – осн. часть универс-го мех. доступа к данным фирмы Майкрософт, позволяющ. осуществить доступ к реляцион. и нереляцион. источн. данных. OLEDB – реализация разработанного Майкрософт объективного стандарта OLE. Для доступа к источнику дан. С пом. OLEDB треб-ся на клиентском компе установить провайдер для данной СУБД. Механизм доступа к данным ADO: – высокоуровнев. программн. интерфейс для доступа к дан. из приложения; - Содержит набор объектов, исп-емых для соединения с источником дан., чтения, добавления и модификации данных.