САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Институт прикладной математики и механики

Кафедра Телематики (при ЦНИИ РТК)

Сергей Геннадьевич Попов

Архитектура систем управления реляционными базами данных

Учебное пособие

Санкт-Петербург - 2015

Содержание курса

ЛЕКЦИЯ 1

Файловые системы

Содержание лекции 1

- Понятие файловой системы
- Способы организации файловых систем
- Классификация файловых ситем
- Записеориентированные файловые системы
- Структура файловых систем
- Файлы и каталоги
- Файлы последовательного доступа
- Недостатки файловых систем
- Преимущества СУБД

Понятие файловой системы

Файловая система (англ. file system) — порядок, определяющий способ организации, хранения и именования данных на носителях информации.

Функции файловой системы:

- именование файлов;
- программный интерфейс работы с файлами для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;
- Обеспечение устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;
- содержание параметров файла, необходимых для правильного его взаимодействия с другими объектами.

Способы организации файловых систем

Классификация файловых систем:

- для носителей с произвольным доступом: FAT32, HPFS, ext2;
- для носителей с последовательным доступом: QIC;
- для оптических носителей: ISO9660, HFS, UDF;
- виртуальные файловые системы: AEFS;
- сетевые файловые системы: : NFS, CIFS, SSHFS, GmailFS;
- для флэш-памяти: YAFFS, ExtremeFFS, exFAT;
- специализированные: ZFS, VMFS.

Файловая система содержит файлы, каталоги и ссылки.

Файл – поименованная область диска.
Каталог – файл со ссылками на другие файлы.
Ссылка – поименованная область, содержащая ссылку на файл.

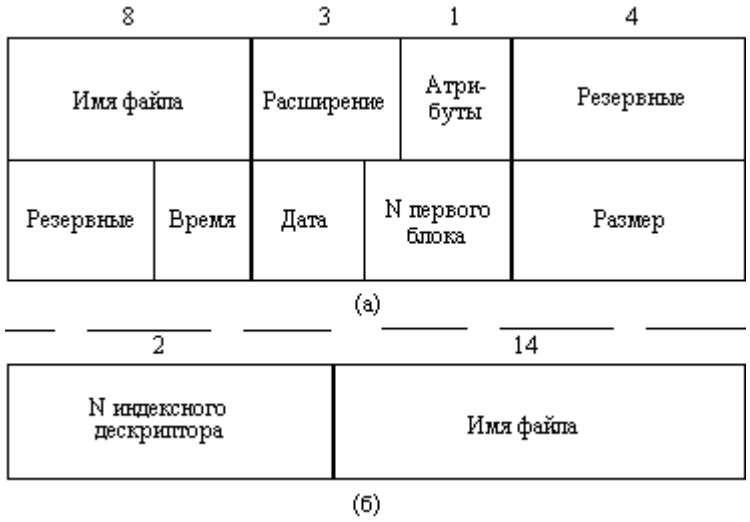
Атрибуты файла

- информация о разрешенном доступе,
- пароль для доступа к файлу,
- владелец файла,
- создатель файла,
- признак "только для чтения",
- признак "скрытый файл",
- признак "системный файл",
- признак "архивный файл",
- признак "двоичный/символьный",
- признак "временный" (удалить после завершения процесса),
- признак блокировки,
- длина записи,
- указатель на ключевое поле в записи,
- длина ключа,
- времена создания, последнего доступа и последнего изменения,
- текущий размер файла,
- максимальный размер файла.

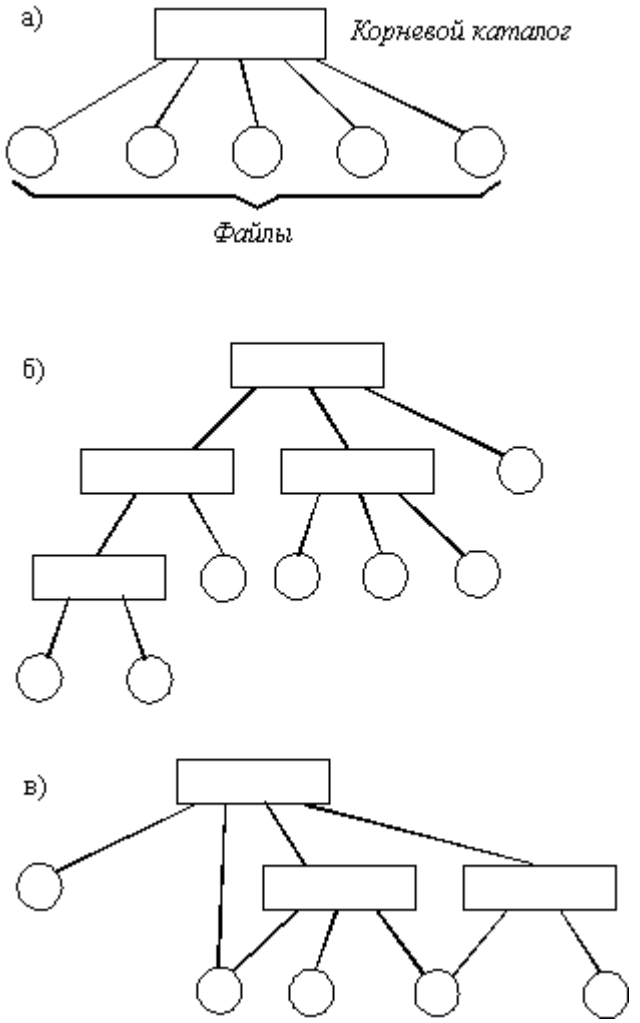
Модель файловой системы

- Каждый уровень модели содержит интерфейсы к следующему уровню

Структура каталогов



Структура записи каталога MS-DOS и Unix



Логическая организация файловой системы

Логическая организация файлов

Индекс	1	2	3	4	5	6
Адрес	21	201	315	661	670	715

Индекс ≡ ключ

Логическая структура хранения файлов

Операции над файлами:

- создание файла,
- уничтожение файла,
- открытие файла,
- закрытие файла,
- чтение файла,
- запись в файл,
- дополнение файла,
- поиск в файле,
- получение атрибутов файла,
- установление новых значений атрибутов,
- переименование,
- выполнение файла,
- чтение каталога.

Недостатки файловых систем

- зависимость данных;
- жесткость;
- статичность;
- отсутствие интеграции;
- дублирование данных;
- противоречивость;
- евозможность совместного использования;
- неэффективность;
- невозможность обработки нетипичных запросов.

Достоинства СУБД

Принципы, соблюдаемые при разработке СУБД:

- Независимость данных.
- Универсальность
- Совместимость
- Неизбыточность данных.
- Защита данных.
- Целостность данных.
- Управление одновременной работой.

Дополнительные принципы:

- СУБД должна быть универсальной
- СУБД должна поддерживать как централизованные, так и распределенные базы данных.

Спасибо за внимание

следующая лекция

Журналирование в файловых системах

ЛЕКЦИЯ 2

Журналирование в файловых системах

Содержание лекции 2

- Потокоориентированные файловые системы
- Тегирование, теги файлов, теговая файловая система
- Состав журналируемой файловой системы
- Типы журналируемых файловых систем
- Технологии журналирования
- Структура журнала файловой системы
- Пример журналируемой файловой системы ext3

Потокоориентированные файловые системы

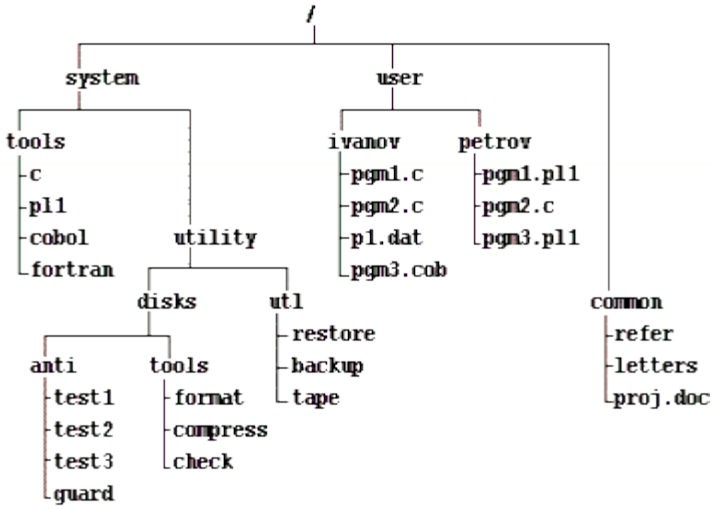
- Цикл работы: открытие файла, чтение и запись данных с носителя, закрытие файла.
- Проблема: при выключении питания в процессе нарушается логическая целостность данных на носителе.
- Решение: реализовать транзакцию – и сохранять данные порциями.
- В этом случае при выключении питания диска логическая целостность файловой системы созраняется.

Транзакция – последовательность операций видимая как выполнение одной.

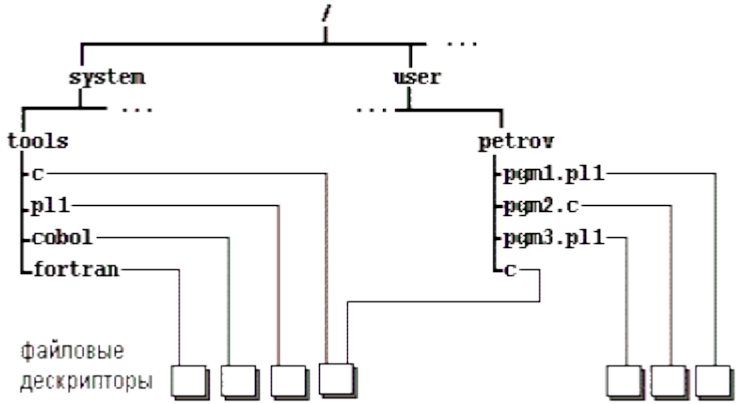
Иерархические файловые системы

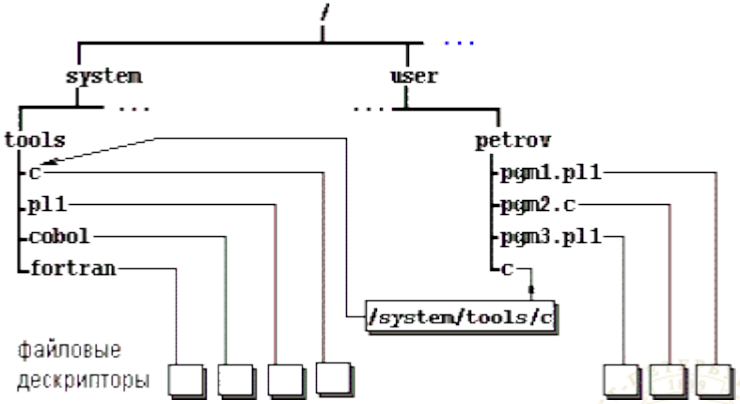
Иерархические файловые системы – представляют логическую структуру в виде дерева

Пример иерархической файловой системы
на основе структуры каталогов fat32.



Определение новых точек подключения для файловых дескрипторов





Применение файловых дескрипторов для задания альтернативных имён файлов

Теговая файловая ситема

- Тег – описатель, классифицирующий файл.
- Классификация осуществляется семантически. Каждому файлу приписывается произвольное число тегов.
- Теги составляют метаданные.
- По метаданным выполняется поиск.

Реализация в форме:
1) файловой системы;
2) фильтра файловой системы.

Примеры:
- ID3v1.x.
- taggedfrog

Состав журналируемой файловой системы

- **Журнал** - это файл, представляющий собой кольцевой буфер, в который заносятся все действия, связанные с изменением файловой системы.
- Журналируемая файловая система – ФС использующая журнал.

- Виды журналирования:
  - обратная запись,
  - упорядочивание,
  - режим данных.

Технологии ведения журнала

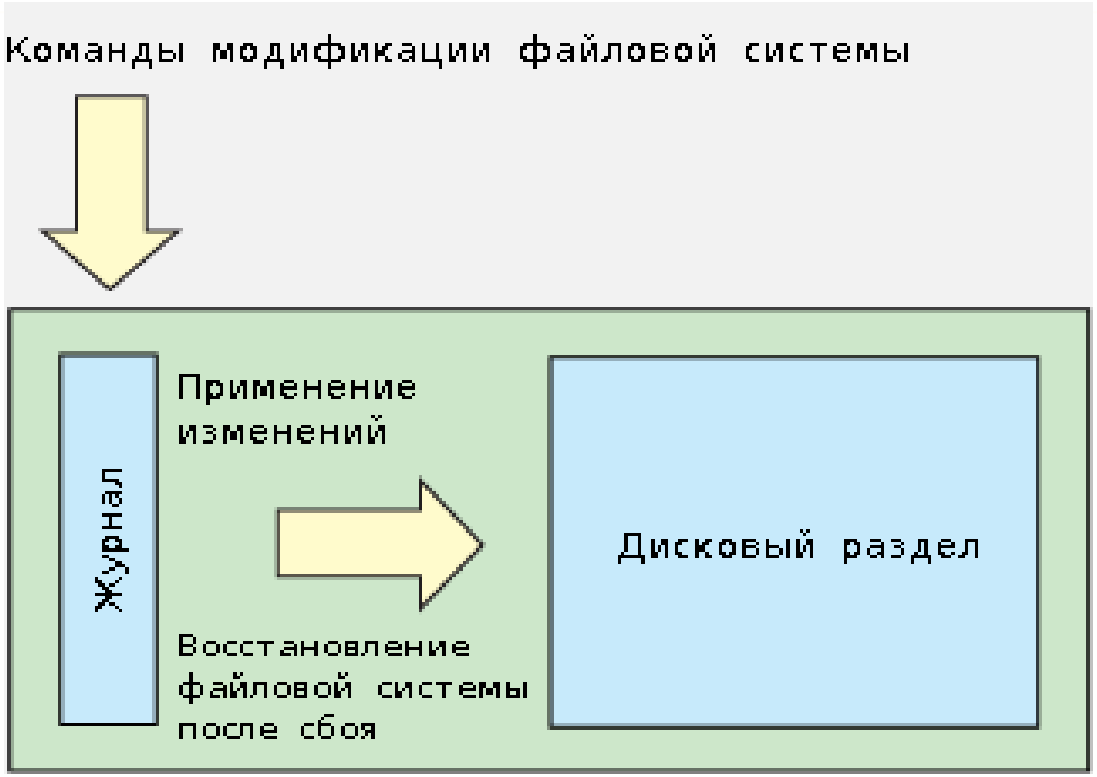
- **Метаданные файла** – данные о расположении, размере, правах и текущем режиме доступа к файлу.

- *Обратная запись* – журналируются только метаданные. Исключаются ошибки структуры хранения, сбои возможны сбои данных файла.
- *Режим упорядочивания* – сначала записываются данные, а затем вносятся изменения в журнал.
- *Режим данных* – в журнале хранятся и метаданные и данные.

- Запись данных на диск производится по событию: полнота журнала или по времени.

Технология функционирования журналируемой файловой системы

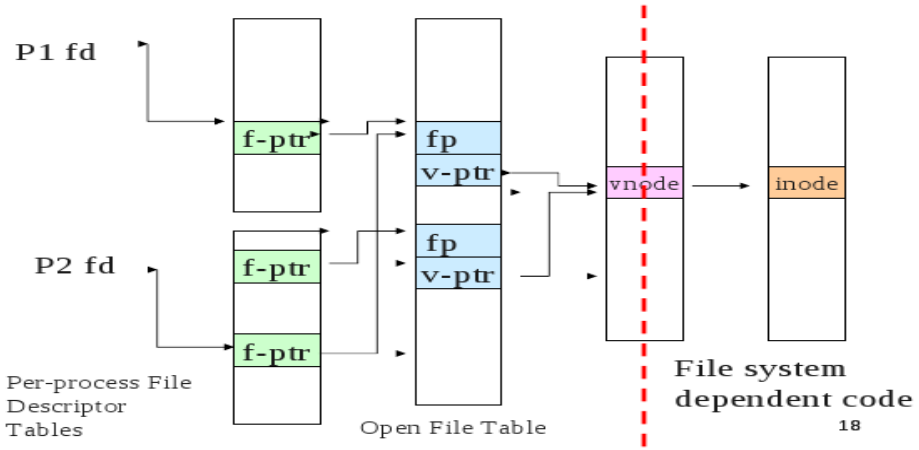
- Вход: высокоуровневые команды модификации содержимого файла
- Выход: запись в журнал и на диск.



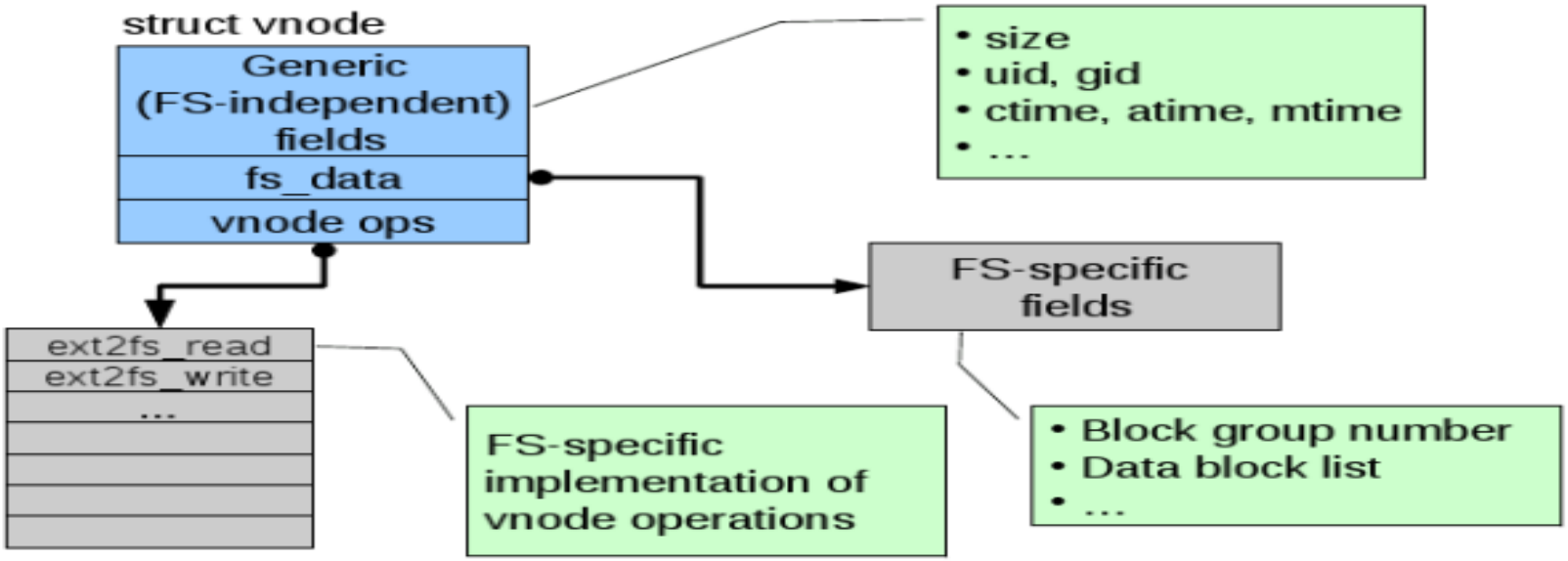
-Замедление работы

+ Восстановление данных после сбоя.

Структура журнала файловой системы VFS



Распределение функций журналирования
между файловой и операционной системами



Структура узла в описании журнала

Реализация журналирования в ext3

- Режимы журналирования:
- *data=writeback, data=ordered и data=journal*

Writeback – только метаданные. Высокая производительность.
Ordered – метаданные совместно с данными транзакциями.
Journal – журналирование данных и метаданных. Очень медленно!

Определение режима:
mount /dev/hda6 /mnt/disc -t ext3 -o data=<режим>

Тонкая настройка ext3:

- Применить инструмент tune2fs.
- Изменить точки монтирования в файле /etc/fstab.
- Изменить параметры ядра.

Спасибо за внимание

следующая лекция

Аппаратное обеспечение хранения данных

ЛЕКЦИЯ 3

Аппаратное обеспечение хранения данных

Содержание лекции 3

- Системы хранения данных.
- Протоколы передачи данных.
- Оборудование хранения данных устройства прямого доступа.
- Технологии сохранения данных.
- Иерархия устройств хранения данных.
- Наборы данных.

Оборудование хранения данных

Система Хранения Данных – это программно-аппаратное решение по организации надёжного хранения информационных ресурсов и предоставления гарантированного доступа к ним.

Типы систем хранения данных:

- Direct-attached storage (DAS)
- Network-attached storage (NAS)
- Storage area network (SAN)

Способы сохранения данных:

- Резервное копирование (Back-up)
- Репликация
- Иерархическое хранение (HSM)

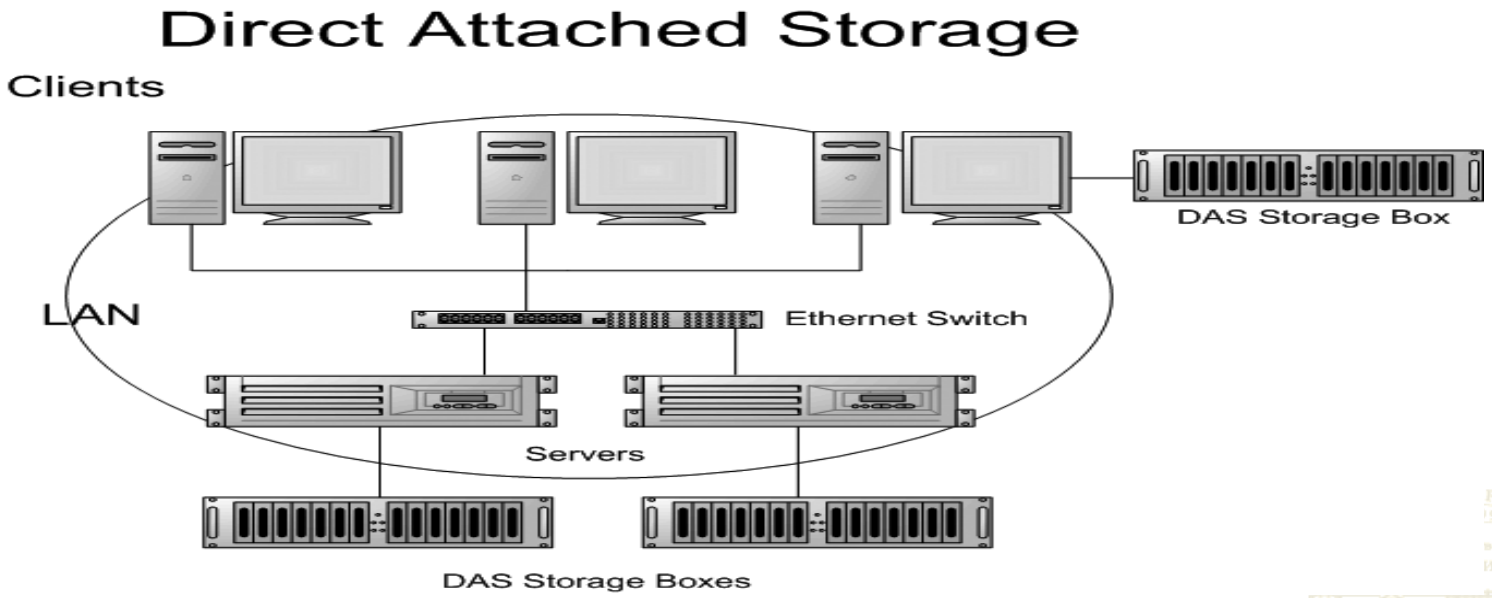
Протоколы доступа к данным

- SCSI - Small Computer System Interface, стандартов для физического подключения и передачи данных между компьютерами и периферийными устройствами.
- iSCSI (IP-SAN) – SCSI поверх Ethernet. На скоростях 1 GB обеспечиваются все требования стандарта.

- Fibre Channel —— семейство протоколов для высокоскоростной передачи данных.
- Fibre Channel Protocol – FC поверх Ethernet.

Direct-attached storage (DAS)

- DAS - непосредственно подключенные к вычислительной системе диски.
- Обычно как DAS квалифицируются варианты только непосредственного прямого подключения.
- Технологии подключения: ATA, SATA, SCSI.

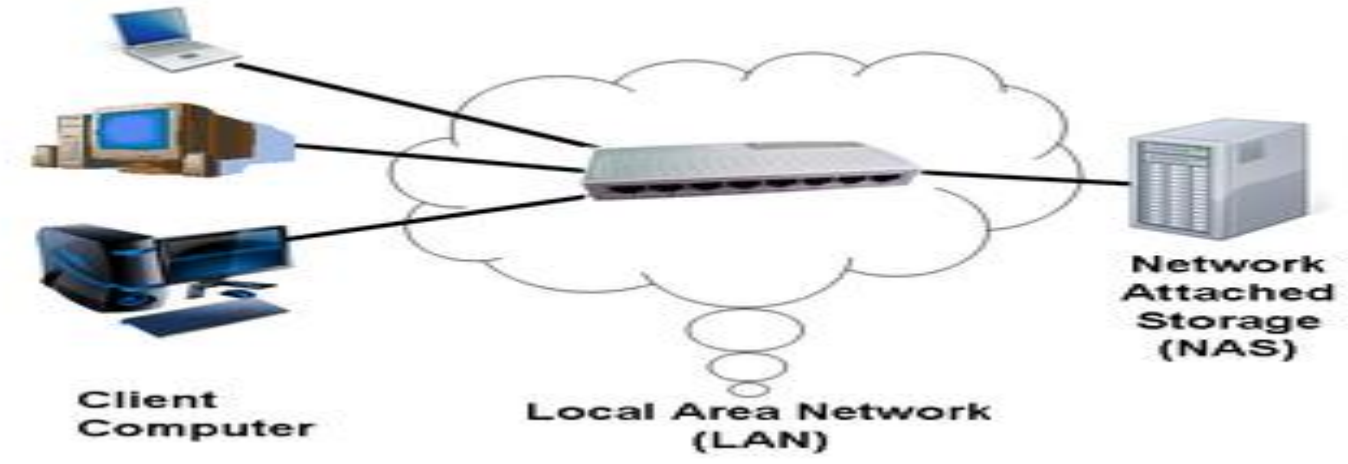


Network-attached storage (NAS)

NAS - устройство, подключенное в локальную сеть и предоставляющее доступ к своим дискам по одному из протоколов «сетевых файловых систем».

Пример - файловый сервер.

Технологии доступа CIFS, SMB NFS.



Storage area network (SAN)

- SAN-устройство, отображённое на локальный диск пользовательской машины.
- Подключается через выделенный адаптер.
- Протокол обращения – FibreChannel или iSCSI.

Технологии сохранения данных

- **Резервное копирование**

- Полное. затрагивает все файлы системы.

- Инкрементальное. Копируются только те файлы, которые были изменены с тех пор, как в последний раз выполнялось полное или добавочное резервное копирование.

- Дифференциальное. при разностном (дифференциальном) резервировании каждый файл, который был изменен с момента последнего полного резервирования, копируется каждый раз заново.

Технологии сохранения данных (продолжение)

- **Репликация.**
- *Синхронная репликация — это зеркалирование данных на две системы хранения* или два дисковых раздела внутри одной системы.

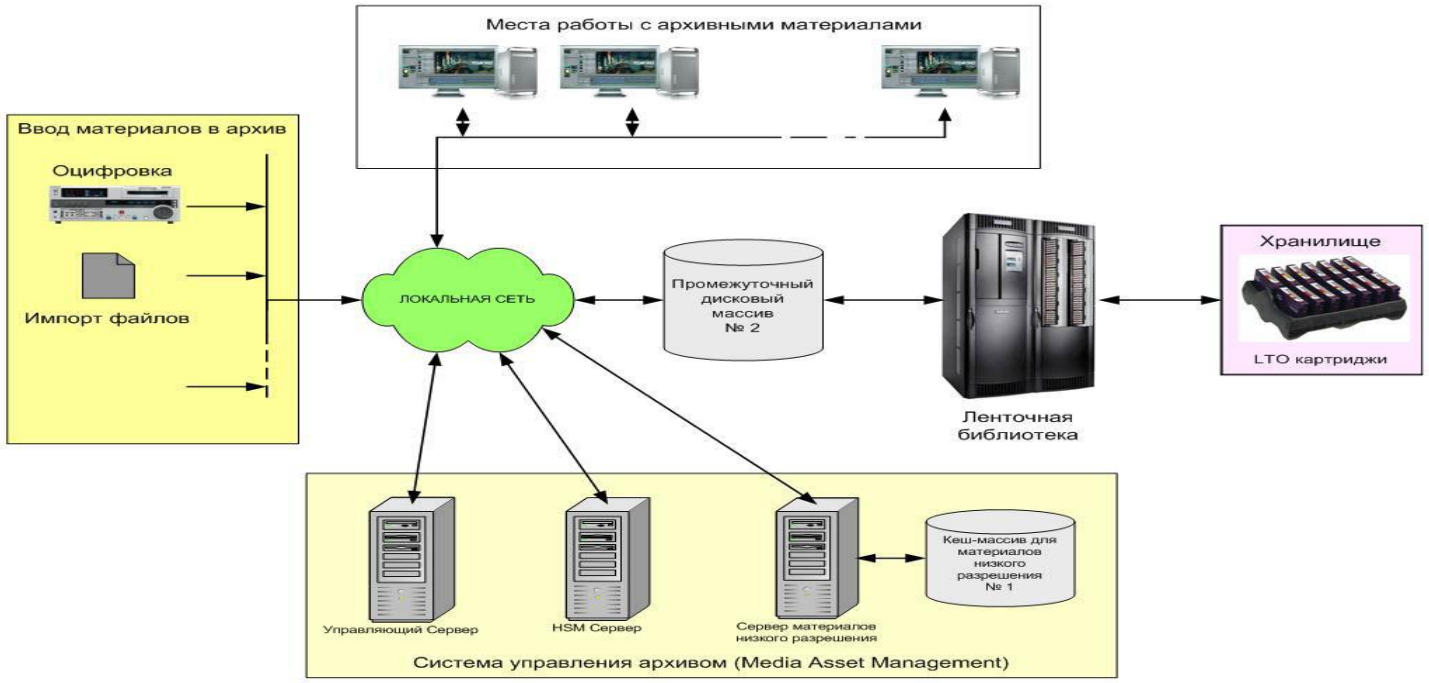
- *Асинхронной* называют репликацию, которая осуществляется не в тот же момент, когда осуществляется запись оригинального блока данных, а в «удобное время».

- *Полусинхронная*. В этом случае репликация проводится синхронной до тех пор, пока это позволяет быстродействие системы или канала связи, а затем на некоторое время переключается в асинхронный режим.

Репликация замедляет работу системы

Иерархическое хранение (HSM)

- Перемещение, автоматическое или в фоновом режиме пользовательских данных между дисками и/или СХД различных классов стоимости и производительности.
- Примеры реализации: EMC FAST, Hitachi Tiered Storage Manager (HTSM).



Спасибо за внимание

следующая лекция

Аппаратная и семантическая целостность данных

ЛЕКЦИЯ 4

Аппаратная и семантическая целостность данных

Содержание лекции 4

- Понятия целостности данных.
- Семантическая целостность.
- Проблема «утраченного обновления»
- Проблема «грязного чтения данных».
- Понятие транзакции.

Целостность данных

- Це́лостность ба́зы да́нных (database integrity) — соответствие имеющейся в БД информации её внутренней логике, структуре и всем явно заданным правилам. Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных.

- поддержка *структурной целостности*
- поддержка *языковой целостности*
- поддержка *ссылочной целостности*
- поддержка *семантической целостности.*

Структурная целостность

- реляционная СУБД должна допускать работу только с однородными структурами данных типа «реляционное отношение» т.е.
- отсутствие дубликатов кортежей,
- соответственно обязательное наличие первичного ключа,
- отсутствие понятия упорядоченности кортежей.

Языковая целостность

Реляционная СУБД должна обеспечивать языки описания и манипулирования данными не ниже стандарта SQL.

Не должны быть доступны иные низкоуровневые средства манипулирования данными, не соответствующие стандарту.

Ограничения целостности атрибута:

- значение по умолчанию,
- задание обязательности или необязательности значений (Null),
- задание условий на значения атрибутов.
- Ограничения целостности, задаваемые на уровне отношения.
- Некоторые семантические правила невозможно преобразовать в выражения, которые будут применимы только к одному столбцу.

Ссылочная целостность

- ОЦ на значения отдельного атрибута сущности.
- ОЦ на атрибуты отдельной сущности.
- ОЦ на допустимые связи между сущностями.
- ОЦ на допустимые множества связей между.
- сущностями, осложнённые условиями на допустимые связи входящих в них сущностей.

Типы целостности

- **Сущностная целостность**
- Сущностная целостность определяет строку как уникальную сущность в конкретной таблице. Она обеспечивает целостность столбцов идентификаторов или первичного ключа таблицы с помощью индексов и ограничений UNIQUE или PRIMARY KEY.

- **Доменная целостность**
- Доменная целостность — это достоверность записей в конкретном столбце. Она включает ограничения типа данных, ограничения формата при помощи ограничений CHECK и правил, а также ограничения диапазона возможных значений при помощи ограничений FOREIGN KEY, CHECK, DEFAULT, определений NOT NULL и правил.

Типы целостности (продолжение)

- **Реализация ссылочной целостности**
- Ссылочная целостность сохраняет определенные связи между таблицами при добавлении или удалении строк. В SQL Server ссылочная целостность основана на связи первичных и внешних ключей (либо внешних и уникальных ключей) и обеспечивается с помощью ограничений FOREIGN KEY и CHECK. Ссылочная целостность гарантирует согласованность значений ключей во всех таблицах. Этот вид целостности требует отсутствия ссылок на несуществующие значения, а также обеспечивает согласованное изменение ссылок во всей базе данных при изменении значения ключа.
- При обеспечении ссылочной целостности SQL Server не допускает следующих действий пользователей.
- Добавления или изменения строк в связанной таблице, если в первичной таблице нет соответствующей строки.
- Изменения значений в первичной таблице, которое приводит к появлению потерянных строк в связанной таблице.
- Удаления строк из первичной таблицы, если имеются соответствующие ей строки в связанных таблицах.

Понятие транзакции

- **Транзакция** – неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными (чтения, удаления, вставки, модификации), такая, что:
- 1) либо результаты всех операторов, входящих в транзакцию, отображаются в БД;
  2) либо воздействие всех операторов полностью отсутствует.

- СУБД отличаются:
- Моделями транзакций
- Автоматическим выполнением транзакций.
- Управляемым выполнением транзакций.

Пример реализации транзакции

Время	<8	8	15
Мои друзья	Друг X	нет	
Мои посты			Друг Y
Друзья X	Я	нет	

Спасибо за внимание

следующая лекция

Реализация транзакции

ЛЕКЦИЯ 5

Реализация транзакций

Содержание лекции 5

- Способы организации транзакции.
- Принципы блокировки доступа к данным.
- Предложения SQL COMMIT.
- Предложения SQL ROLLBACK.
- Реализация транзакций в SQL.
- Блокировки
- Расширенные блокировки.

Выпонение транзакции

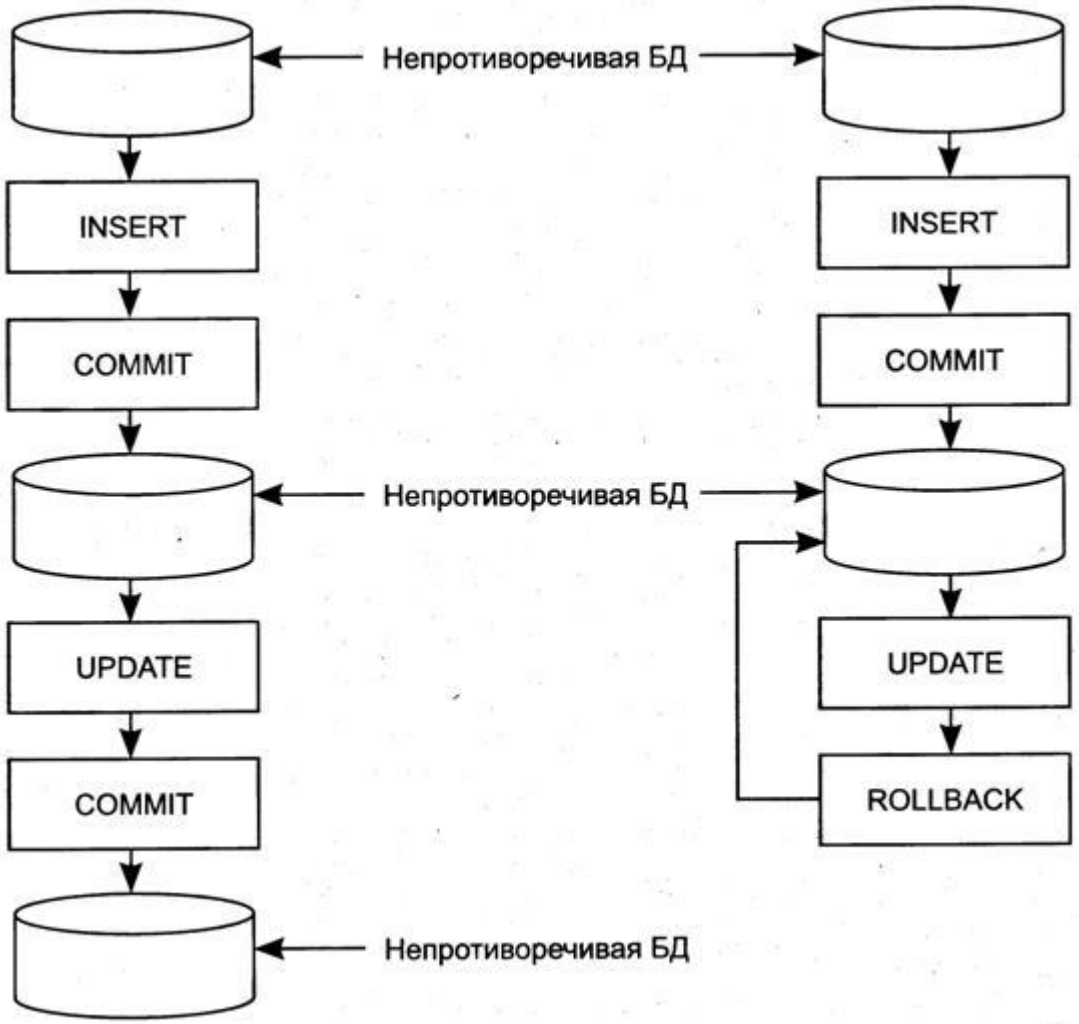
- *Транзакция* – это законченная совокупность действий над БД, которая переводит ее из одного целостного состояния в другое. Если операторы транзакции выполняются, происходит ее нормальное завершение и БД переходит в обновленное (целостное) состояние.

Требования к транзакции

- К транзакциям предъявляется набор требований, известный под названием ACID (Atomicity, Consistency, Isolation, Durability).

- *Атомарность* (atomicity). Транзакция представляет собой набор законченных действий.
- *Согласованность* (consistency). Предполагается, что в результате выполнения транзакции система переходит из одного корректного состояния в другое.
- *Изолированность* (isolation). При выполнении транзакции данные могут временно находиться в несогласованном состоянии.
- *Долговечность* (durability)

Автоматическое выполнение транзакции



Начальное состояние

Выполнение действия

Фиксация действия

Новое стабильное состояние

Выполнение действия

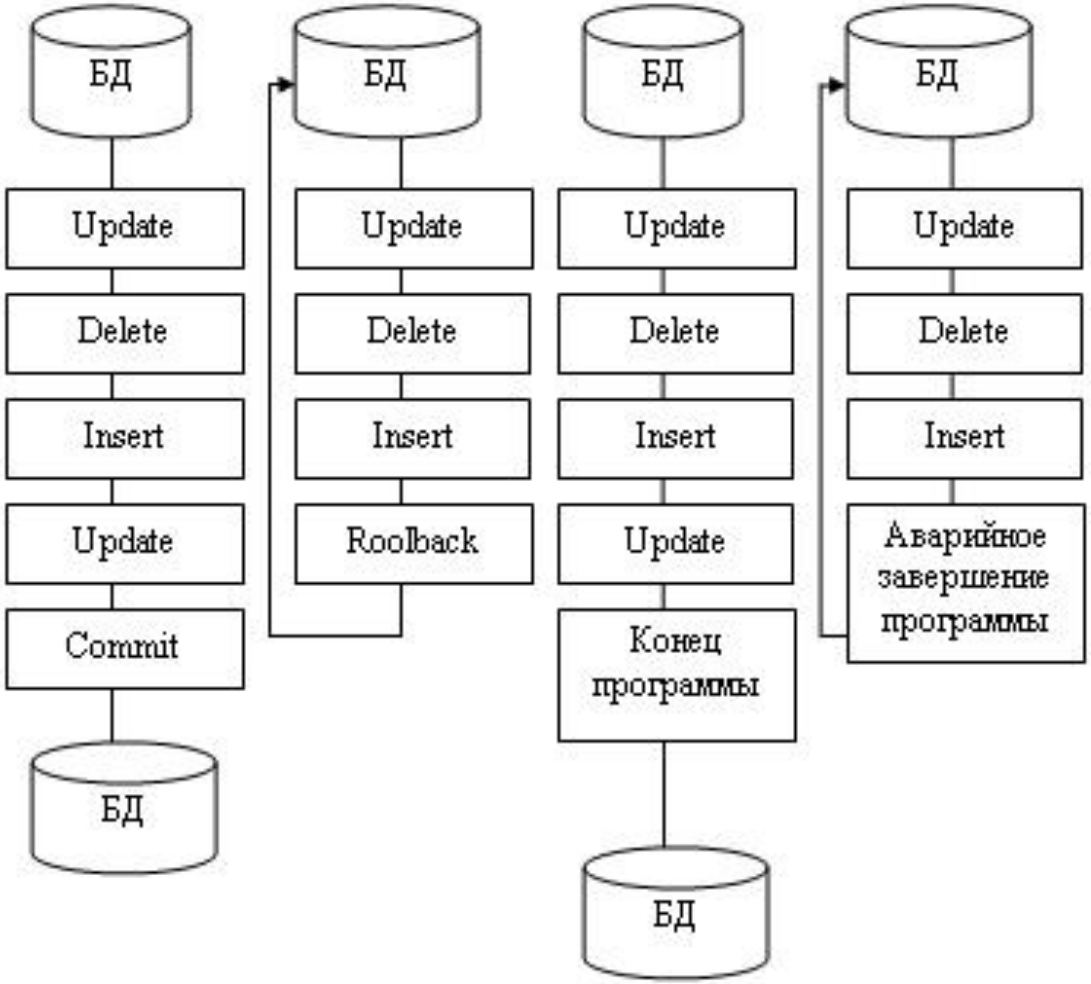
Фиксация т откат действия

Новое стабильное состояние

Управляемое выполнение транзакций

- инструкция BEGIN TRANSACTION сообщает о начале транзакции;
- инструкция COMMIT TRANSACTION сообщает об успешном выполнении транзакции;
- инструкция SAVE TRANSACTION позволяет создать внутри транзакции точку сохранения и присвоить сохраненному состоянию имя точки сохранения, указанное в инструкции;
- инструкция ROLLBACK отменяет выполнение текущей транзакции и возвращает БД к состоянию, где была выполнена инструкция SAVE TRANSACTION или к состоянию начала транзакции.

Модель транзакции *ANSI/ISO*



Начальное состояние

Операция 1

Операция 2

Операция 3

Завершение транзакции

Конечное состояние

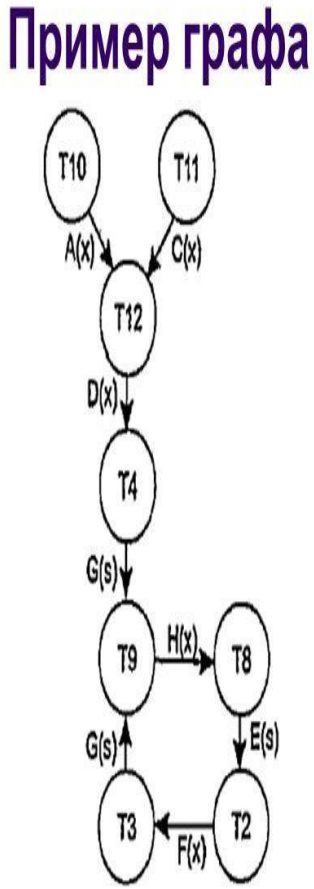
Блокировки

Для обеспечения атомарности транзакций Применяется упреждающее протоколирование.

СУБД обеспечивает блокировки двух типов:

S-блокировка - разделяемая блокировка или блокировка чтения, допускающая совместный доступ к строке таблицы;

X-блокировка - эксклюзивная блокировка или блокировка записи, не допускающая совместного доступа к строке.



- T1...T12 – транзакции
- A..G – объекты
- X- жеская блокировка
- S- мягкая

Расширенные блокировки СУБД

- IS - блокировка намерения чтения. Накладывается на некоторую таблицу T и означает намерение блокировать некоторую входящую в T строку в режиме S-блокировки.
- IX - блокировка намерения записи. Накладывается на некоторую таблицу T и означает намерение блокировать некоторую входящую в T строку в режиме X-блокировки.
- SIX - блокировка чтения с намерением записи. Накладывается на некоторую таблицу T и означает разделяемую блокировку всей этой таблицы с намерением впоследствии блокировать какие-либо входящие в нее строки в режиме X-блокировок.

Пример реализации блокировок

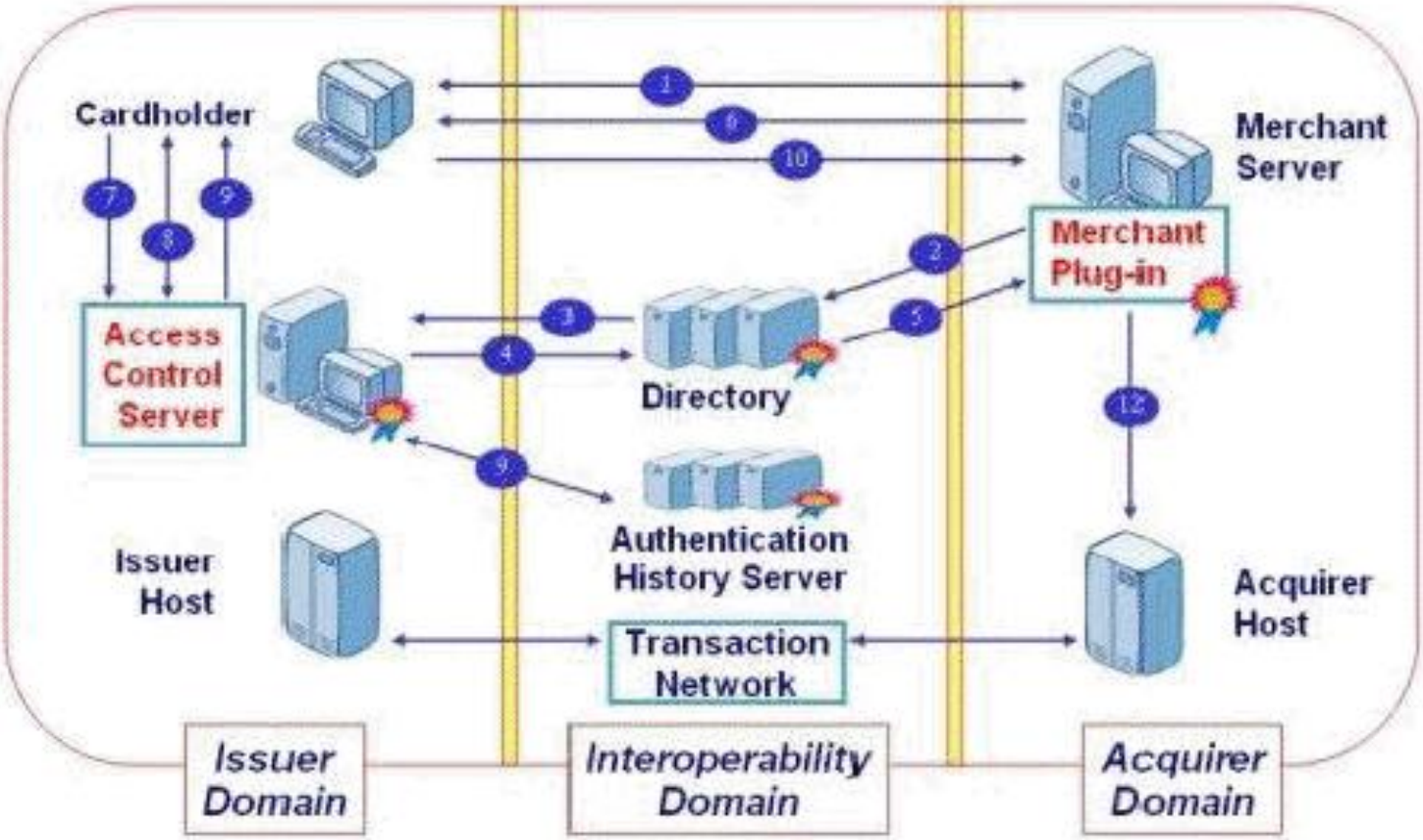


Рис. 2: Проведение онлайновой покупки

Спасибо за внимание

следующая лекция

Разрешение коллизий в транзакциях

ЛЕКЦИЯ 6

Разрешение коллизий в транзакциях

Содержание лекции 6

- Блокировки таблиц.
- Предложение SQL LOCK TABLE.
- Проблемы, связанные с блокировками.
- Понятие тупика.
- Бесконечное откладывание.
- Способы разрешения проблем.

Технологии блокировки таблиц

- Строковые блокировки

+ Меньше конфликтов блокировок при обращении к различным строкам из множества потоков.
+ Меньше изменений при откатах.
+ Возможна блокировка одной строки на длительное время.
- Требуется больше памяти.
- Строковая блокировка работает медленнее в сравнении с блокировкой на уровне страниц или таблиц.

- Блокировка уровня страницы и таблицы

+ Когда производится главным образом чтение.
+ При чтении и обновлении для строго заданных ключей;
+ Выполняется много операций просмотра и группировки.

Технологии блокировки таблиц

- LOCK TABLES tbl_name [AS alias]
- {READ | [READ LOCAL] | [LOW_PRIORITY] WRITE} [, tbl_name {READ | [LOW_PRIORITY] WRITE} ...]
- ...
- UNLOCK TABLES

- Команда LOCK TABLES блокирует указанные в ней таблицы для данного потока. Команда UNLOCK TABLES снимает любые блокировки
- Блокировка предотвращает захват таблицы другим запросом
- Блокировка WRITE обычно имеет более высокий приоритет, чем блокировка READ.

Примеры типов блокировок в MS SQL Server

Блокировки данных

- Совмещаемая блокировка (S)
- Блокировка обновления (U)
- Монопольная блокировка (X)
- Блокировка с намерением
- Блокировка схемы
- Блокировка массового обновления (BU)
- Блокировка обновления с намерением монопольного доступа (UIX)

Блокировки схемы

- блокировка изменения схемы (Sch-M)
- Блокировка стабильности схемы (Sch-S)

Проблемы блокировок

- Длительная временная блокировка.
- Взаимная блокировка.
- Постоянная взаимная блокировка.

- Блокировка самой базы данных.
- Блокировка файлов базы данных.
- Блокировка таблиц базы данных.
- Блокировка страниц (Единиц обмена с диском, обычно 2-16 Кб. На одной странице содержится несколько строк одной или нескольких таблиц).
- Блокировка отдельных строк таблиц.
- Блокировка отдельных полей.

Возникает тупик

Понятие тупика

- Проблема потери результатов обновления - *возник тупик*.
- Проблема незафиксированной зависимости (чтение "грязных" данных, неаккуратное считывание) - *проблема разрешилась*.
- Неповторяемое считывание - *проблема разрешилась*.
- Появление фиктивных элементов - *проблема не разрешилась*.
- Проблема несовместимого анализа - *возник тупик*.

- *Тупик* – взаимный монопольный захват двух объектов в пересечении двух транзакций.

- ***Тупик*** – состояние процесса, в котором он ожидает наступления события, которое никогда не произойдет.
- ***Зависание*** – ситуация, когда один или более процессов оказываются в состоянии тупика.

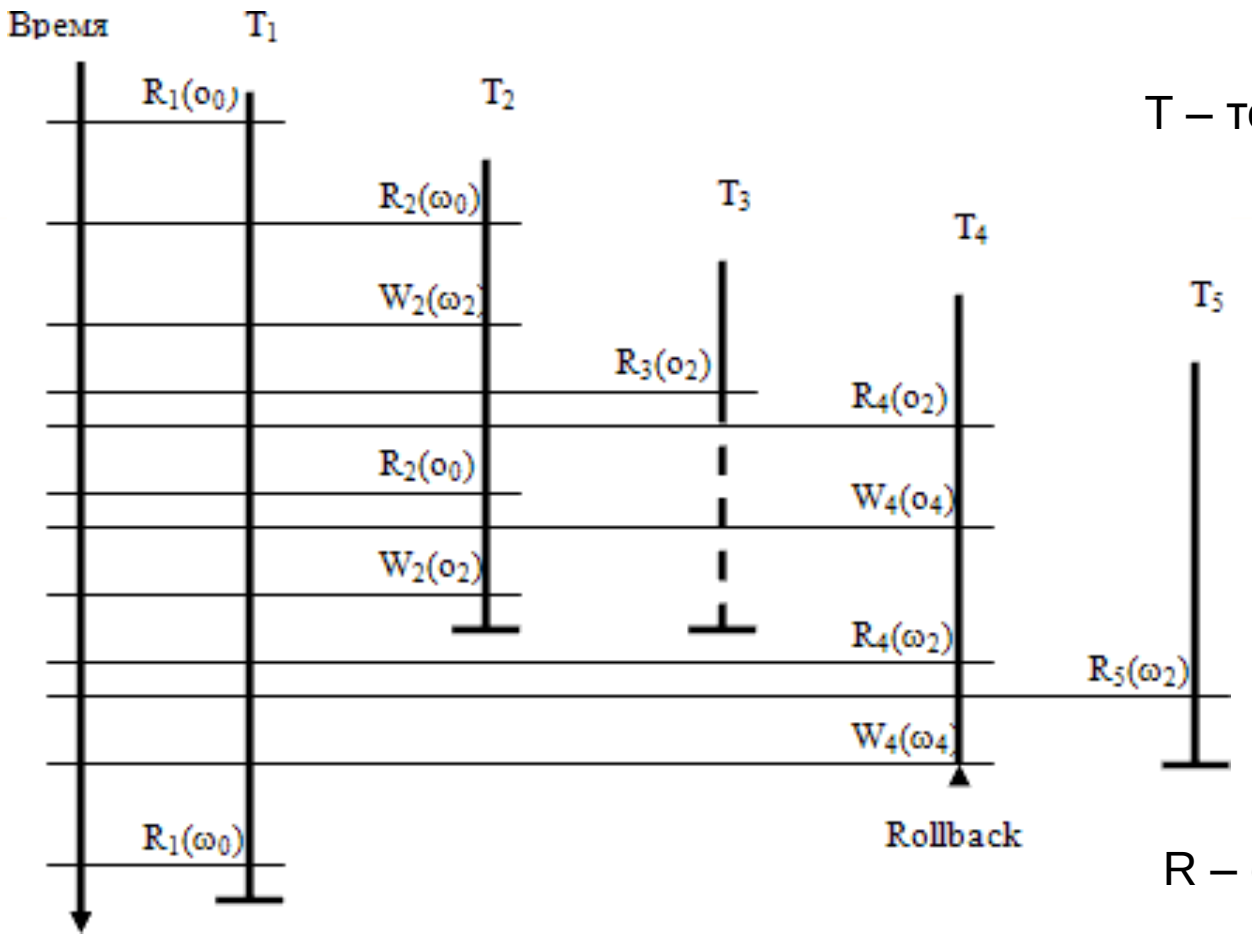
Тупики могут быть вызваны:

- ожидание несуществующего события
- ожидание ненаступаемого события
- бесконечное откладывание (ситуация, при которой предоставление процессора некоторому процессу будет откладываться на неопределенно долгий срок, в то время как система будет уделять внимание другим процессам.)

Разрешение тупиковых ситуаций

- Тупики могут быть предотвращены на стадии написания программ, то есть про¬граммы должны быть написаны таким образом, чтобы тупик не мог возникнуть при любом соотношении взаимных скоростей потоков.
- В тех же случаях, когда тупиковую ситуацию не удалось предотвратить, важно быстро и точно ее распознать, поскольку блокированные потоки не выполняют никакой полезной работы.
- Если же тупиковая ситуация возникла, то не обязательно снимать с выполнения все заблокированные потоки. Можно снять только часть из них, освободив ресурсы, ожидаемые остальными потоками.

Пример и разрешение тупика



T – точки журналирования

ω – состояние при завершении

R – состояние отношения при завершении

Спасибо за внимание

следующая лекция

Журналирование в базах данных

ЛЕКЦИЯ 7

Журналирование в базах данных

Содержание лекции 7

- Журналирование изменений БД.
- Индивидуальные откаты транзакций.
- Восстановление после «мягкого» сбоя. Тёплый пуск.
- Восстановление после «жесткого» сбоя. Холодный пуск.
- Мониторы транзакций.
- Пример реализации в IBM CICS и TPF.

Журнализация изменений

- Журнализация изменений — функция СУБД, которая сохраняет информацию, необходимую для восстановления базы данных в предыдущее согласованное состояние в случае логических или физических отказов.

- В простейшем случае журнализация изменений заключается в последовательной записи во внешнюю память всех изменений, выполняемых в базе данных.

- Объём данных для журналирования:
- порядковый номер, тип и время изменения;
- дентификатор транзакции;
- объект, подвергшийся изменению;
- предыдущее состояние объекта и новое состояние объекта.

Виды восстановления данных

- *Индивидуальный откат транзакции*. Откат индивидуальной транзакции может быть инициирован либо самой транзакцией путем подачи команды ROLLBACK, либо системой.

- *Мягкий сбой системы* (*аварийный отказ программного обеспечения*). Мягкий сбой характеризуется утратой *оперативной* памяти системы.

- *Жесткий сбой системы* (*аварийный отказ аппаратуры*). Жесткий сбой характеризуется повреждением *внешних носителей* памяти.

Индивидуальный откат транзакции

Откат требуется в случае: штатного использования ПО СУБД.

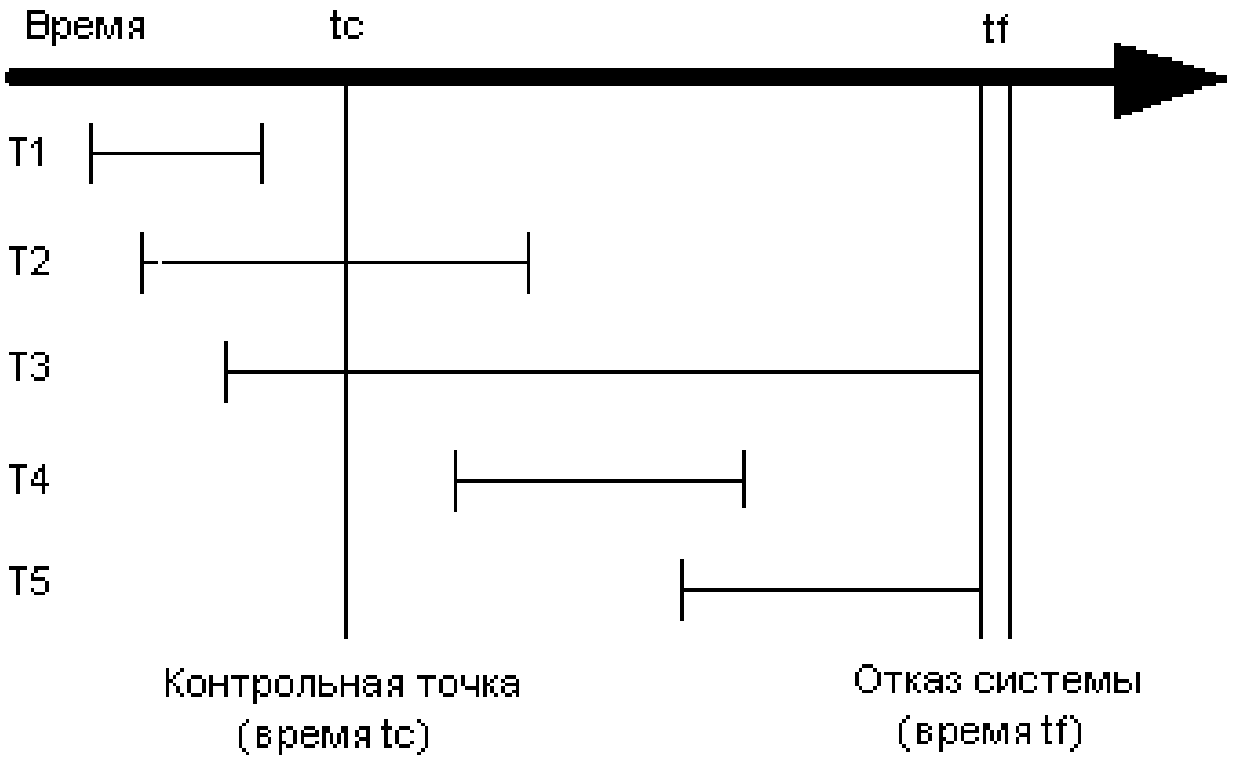
Для транзакции, требующей отката:

- Выбирается очередная запись из списка данной транзакции.
- Выполняется противоположная по смыслу операция.
- Любая из этих обратных операций также журналируются.
- При успешном завершении отката в журнал заносится запись о конце транзакции.

С точки зрения СУБД такая транзакция считается зафиксированной.

Восстановление после «мягкого» сбоя

- Поражаются все выполняющиеся в момент сбоя транзакции, теряется содержимое всех буферов базы данных. Данные, хранящиеся на диске, остаются неповрежденными.



Восстановление после «мягкого» сбоя (продолжение)

- ***T1*** - транзакция успешно завершена до принятия контрольной точки.
- ***T2*** - транзакция начата до принятия контрольной точки и успешно завершена после контрольной точки, но до наступления сбоя.
-
- ***T3*** - транзакция начата до принятия контрольной точки и не завершена в результате сбоя.
- ***T4*** - транзакция начата после принятия контрольной точки и успешно завершена до сбоя системы.
- ***T5*** - транзакция начата после принятия контрольной точки и не завершена в результате сбоя. Никаких следов этой транзакции нет ни во внешней памяти журнала транзакций, ни во внешней памяти базы данных. Для такой транзакции никаких действий предпринимать не нужно, ее как бы и не было вовсе.

Восстановление после жесткого сбоя

При жестком сбое база данных на диске нарушается физически.

Восстановление начинается с обратного копирования базы данных из архивной копии.

Затем выполняется просмотр журнала транзакций для выявления всех транзакций, которые закончились *успешно* до наступления сбоя.

Наиболее плохим случаем является ситуация, когда разрушены физически и база данных, и журнал транзакций. В этом случае единственное, что можно сделать - это восстановить состояние базы данных на момент последнего резервного копирования.

Для того чтобы не допустить возникновение такой ситуации, базу данных и журнал транзакций обычно располагают на *физически* разных дисках, управляемых физически разными контроллерами.

Мониторы транзакций

Монитор транзакций – система управления очередями во многопользовательских системах «клиент-сервер».

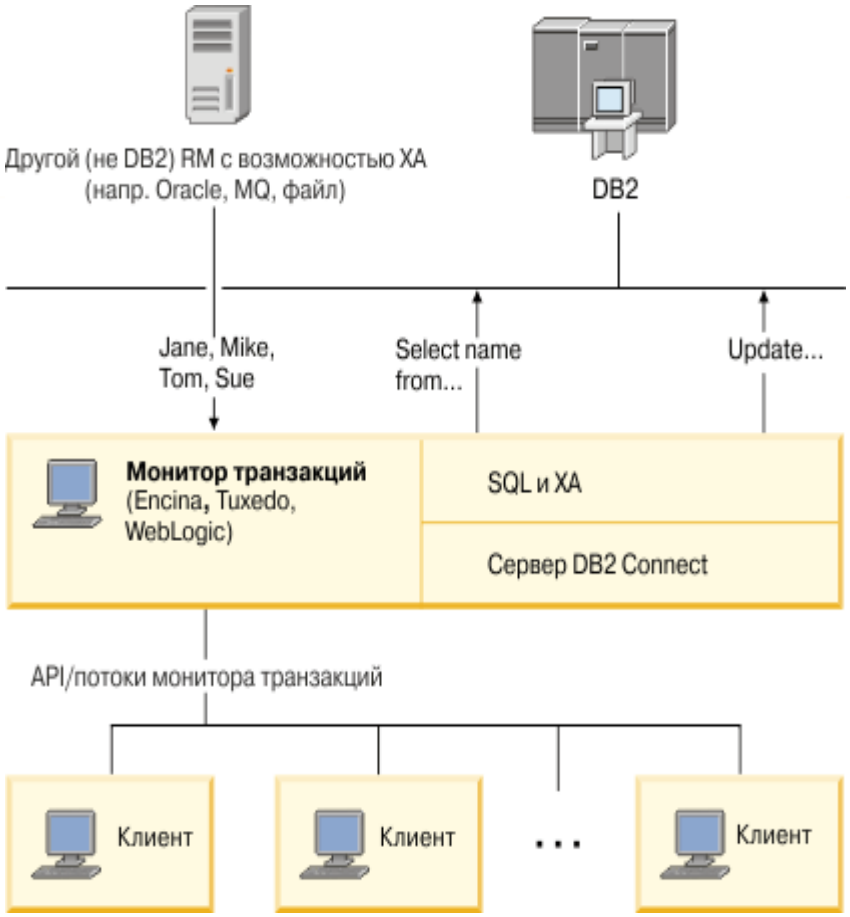
Использование монитора подразумевает определенный стиль в написании приложений.

- Приложение перестаёт быть монолитным, оно представлено группой сервисов;
- Эти сервисы не вызываются напрямую, они вызываются run-time средой монитора, в которую они погружены.

Эффективность достигается за счёт того, что исполняющая среда имеет доступ к внутренней структуре приложения.

Монитор реализует на согласованную многозадачность где единицей планирования является сервис.

Мониторы транзакций. Пример IBM TxSeries CICS



Реализующая коммерческий алгоритм прикладная программа может внутри одной транзакции потребовать изменения данных в нескольких базах данных.

Общий интерфейс для транзакций между монитором транзакций и любыми ресурсами

Эффективное обращение клиентов к гетерогенным базам данных

Спасибо за внимание

следующая лекция

Разграничение доступа в базах данных

ЛЕКЦИЯ 8

Разграничение доступа в базах данных

Содержание лекции 8

- Схема разграничения доступа.
- Предложения SQL GRANT и REVOKE.
- Изолированность пользователей
- Уровни изолированности.
- Метки доступа.
- Способ организации меток доступа для СУБД, не поддерживающих этот механизм.

Схема разграничения доступа

Особенности реализации MLS/DBMS:

- каждый элемент данных в базе данных связан с уровнем доступа;
- доступ пользователя к данным должен контролироваться релевантностью для данного пользователя.

- Уровни доступа L1 и L2.

- 1. Множество всех уровней доступа образует частично-упорядоченную решетку;
- 2. L2 <= L1.

Схема разграничения доступа (продолжение)

- 1. Всем пользователям гарантируется максимальный уровень релевантности данных. Объекты доступны для пользователей уровня доступа, на котором находятся пользователи.
- 2. Объектами являются данные из базы данных.
- 3. Субъекты имеют доступ по чтению для объектов, только если уровень доступа субъекта доминируется уровнем доступа объекта.
- 4. Субъект имеет доступ по записи для объекта, если уровень доступа субъекта доминирует уровень доступа объекта или субъект является привилегированным пользователем.
- 5. Ответ на запрос не должен реализовываться для субъекта, если этот субъект или другие субъекты, которые могут прочитать этот ответ, могут скомбинировать ответ из ранее реализованных ответов и вывести данные на уровне доступа, который не доминирует уровень субъекта.

Распределение прав GRANT

```
GRANT priv_type [(column_list)] [, priv_type
[(column_list)] ...]
ON {tbl_name | * | *.* | db_name.*}
TO user_name [IDENTIFIED BY [PASSWORD] 'password']
             [, user_name [IDENTIFIED BY 'password'] ...]
     [REQUIRE [{SSL| X509}]
     [CIPHER cipher [AND]]
     [ISSUER issuer [AND]]
     [SUBJECT subject]]
WITH [GRANT OPTION | MAX_QUERIES_PER_HOUR # |
                     MAX_UPDATES_PER_HOUR # |
                     MAX_CONNECTIONS_PER_HOUR #]]
```

Хранение прав доступа в MySQL

- **Глобальный уровень** Глобальные привилегии применяются ко всем базам данных на указанном сервере.

- **Уровень базы данных** Привилегии базы данных применяются ко всем таблицам указанной базы данных.

- **Уровень таблицы** Привилегии таблицы применяются ко всем столбцам указанной таблицы.

- **Уровень столбца** Привилегии столбца применяются к отдельным столбцам указанной таблицы.

Отказ от назначения прав в MySQL

- `REVOKE priv_type [(column_list)]`
- `[, priv_type [(column_list)] ...]`
- `ON {tbl_name | * | *.* | db_name.*}`
- `FROM user_name [, user_name ...]`

В MySQL привилегии назначаются для сочетания имя пользователя + удаленный компьютер, а не только для имени пользователя.

В ANSI SQL отсутствуют глобальные привилегии и привилегии уровня базы данных, и ANSI SQL поддерживает не все типы привилегий MySQL.

Структура привилегий ANSI SQL является иерархической. Если удалить пользователя, то все назначенные этому пользователелю привилегии будут отменены.

В MySQL пользователь может применять к таблице оператор INSERT при наличии у него привилегии INSERT только для нескольких столбцов в этой таблице.

При удалении таблицы в ANSI SQL все привилегии для этой таблицы будут отменены.

Изолированность пользователей

Во многопользовательских системах с одной базой данных одновременно могут работать несколько пользователей или прикладных программ.

- Первый уровень - отсутствие потерянных изменений.
- Второй уровень - отсутствие чтения "грязных данных".
- Третий уровень - отсутствие неповторяющихся чтений.

В связи со свойством сохранения целостности БД транзакции являются подходящими единицами изолированности пользователей.

Уровни изоляции транзакции

Установка уровня изоляции

```
SET TRANSACTION ISOLATION LEVEL
      { READ UNCOMMITTED     |
        READ COMMITTED     |
        REPEATABLE READ     |
        SNAPSHOT     |
         SERIALIZABLE     } [ ; ]
```

- **READ UNCOMMITTED** Указывает, что инструкции могут считывать строки, которые были изменены другими транзакциями, но еще не были зафиксированы.

- **READ COMMITTED** Указывает, что инструкции не могут считывать данные, которые были изменены другими транзакциями, но еще не были зафиксированы.

Уровни изоляции (продолжение)

- **REPEATABLE READ** Указывает на то, что инструкции не могут считывать данные, которые были изменены, но еще не зафиксированы другими транзакциями, а также на то, что другие транзакции не могут изменять данные, читаемые текущей транзакцией, до ее завершения.
- **SNAPSHOT** Указывает на то, что данные, считанные любой инструкцией транзакции, будут согласованы на уровне транзакции с версией данных, существовавших в ее начале.
- **SERIALIZABLE** Указывает следующее.
  - Инструкции не могут считывать данные, которые были изменены другими транзакциями, но еще не были зафиксированы.
  - Другие транзакции не могут изменять данные, считываемые текущей транзакцией, до ее завершения.
  - Другие транзакции не могут вставлять новые строки со значениями ключа, которые входят в диапазон ключей, считываемых инструкциями текущей транзакции, до ее завершения.

Спасибо за внимание

следующая лекция

Базы данных и криптографическая защита

ЛЕКЦИЯ 9

Базы данных и криптографическая защита

Содержание лекции 9

- Использование представлений для разграничения доступа к данным.
- Шифрование данных в СУБД.
- Алгоритмы с открытым и закрытым ключами.
- Понятие криптографического ящика В СУБД.
- Цифровая подпись.
- Протокол SSL.

Представления

- Представление (VIEW) - объект данных который не содержит никаких данных его владельца.

- Представление создаётся командой CREATE VIEW.

Представление используется, чтобы ограничить доступ к данным базы данных - с помощью представлений обеспечивается ещё один уровень защиты данных. Пользователю могут предоставляться права только на представление, благодаря чему он не будет иметь доступа к данным, находящимся в тех же таблицах, но не предназначенных для него.

Шифрование данных в СУБД

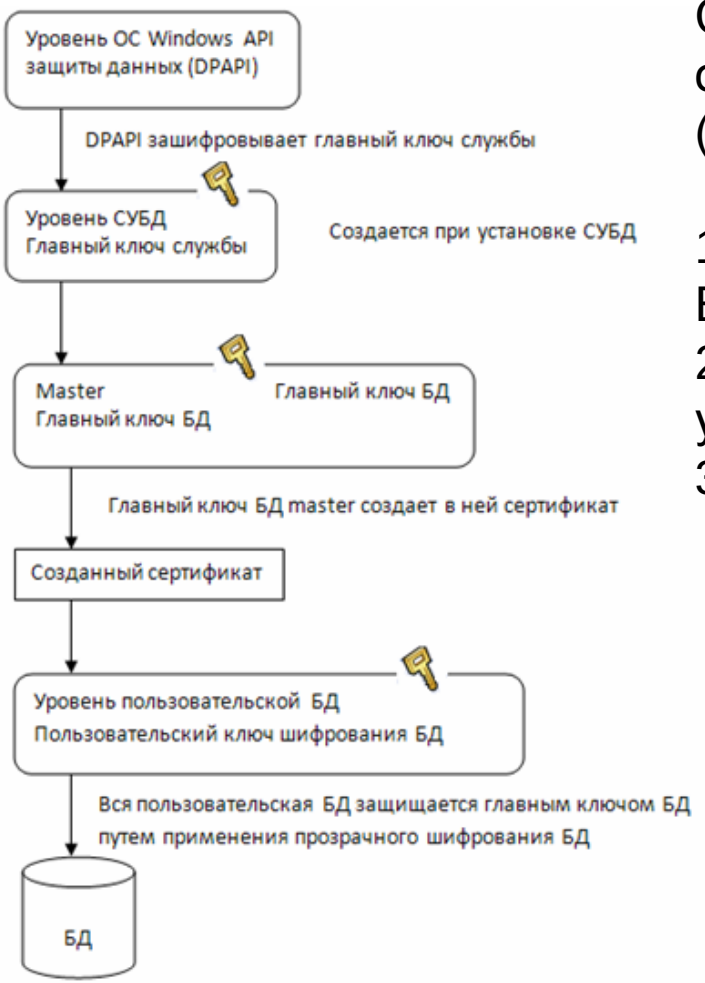
По способу функционирования системы шифрования СУБД делят на два класса:

- системы прозрачного шифрования (включаются администратором);
- системы, вызываемые пользователем (непрозрачное шифрование).

Применяемые алгоритмы шифрования включают DES, 3DES, AES (128, 192, 256 бит).

- Недостатки прозрачного шифрования:
- большая нагрузка на ЦП чем в непрозрачном режиме;
- необходимость администрирования.

Распределение ключей шифрования

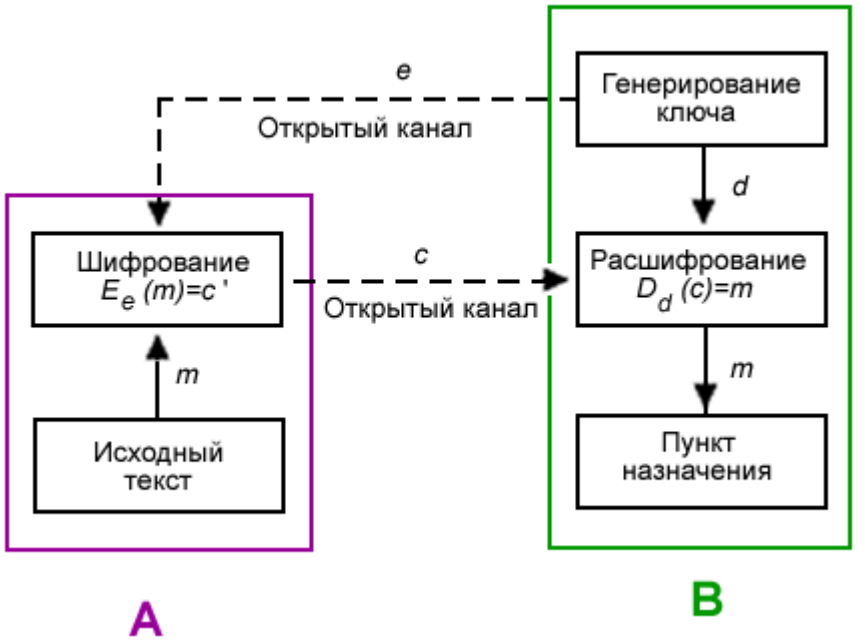


С точки зрения файловой системы есть три основных варианта применения шифрования (и прозрачного, и непрозрачного):

1) Шифрование всего каталога с файлами БД на уровне ОС.
2) Шифрование собственно файлов БД на уровне СУБД.
3) Шифрование столбцов в таблицах БД.

Преимущества непрозрачного шифрования:
- минимальная нагрузка на центральный процессор для шифрования и дешифрования;
- нет необходимости администрирования.

Шифрование данных с открытым ключом



Идея криптографии с открытым ключом очень тесно связана с идеей односторонней функции, то есть таких функций , что по известному довольно просто, тогда как определение из невозможно за разумный срок.

1. Боб выбирает пару и шлёт ключ шифрования (открытый ключ) Алисе по открытому каналу, а ключ расшифрования (закрытый ключ) защищён и секретен (он не должен передаваться по открытому каналу).

2. Чтобы послать сообщение Бобу, Алиса применяет функцию шифрования, определённую открытым ключом : , — полученный шифротекст.

3. Боб расшифровывает шифротекст , применяя обратное преобразование , однозначно определённое значением .

Шифрование данных с закрытым ключом

Закрытый ключ — сохраняемый в тайне компонент ключевой пары.

Примеры использования закрытого ключа:

Электронная подпись

Шифрование сообщений

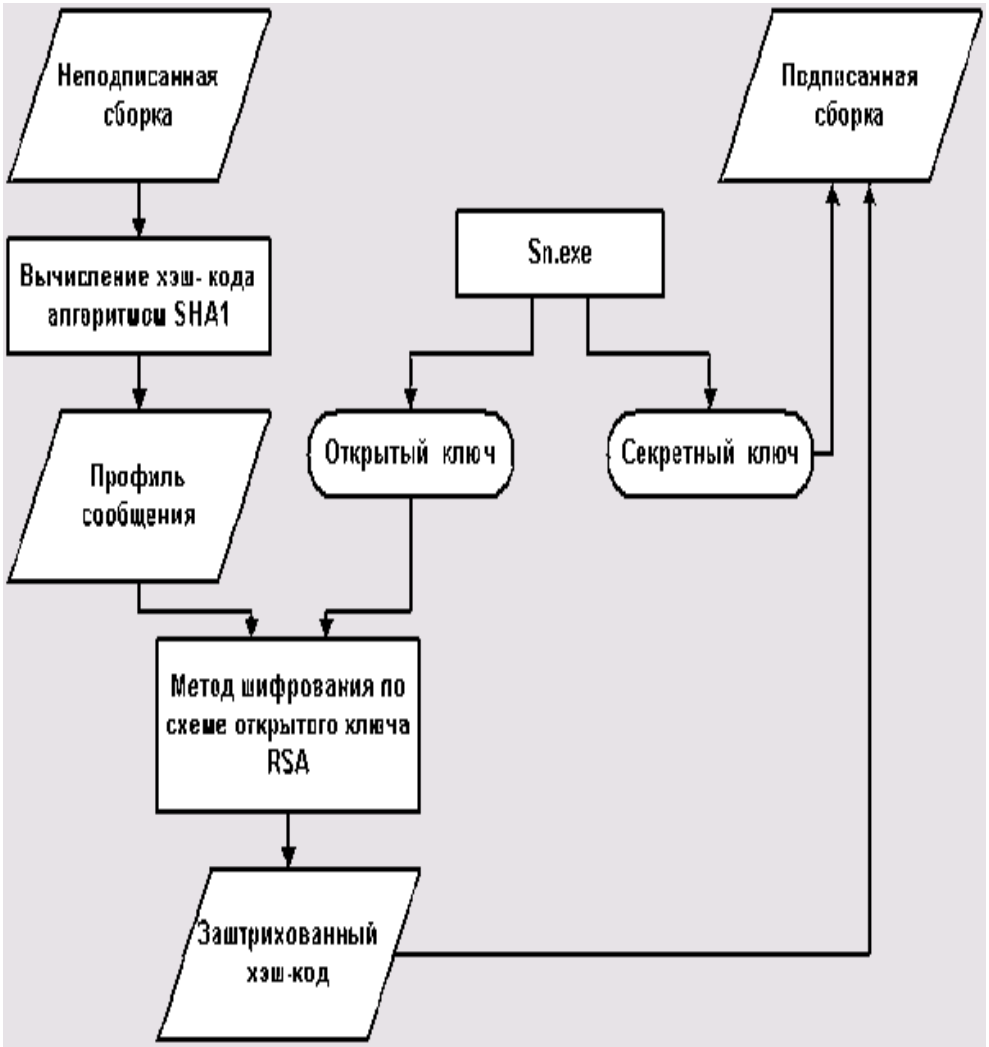
Криптографический ящик. Цифровая подпись.

Электро́нная цифровая по́дпись (ЭЦП) — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи.

Использование электронной подписи позволяет осуществить:

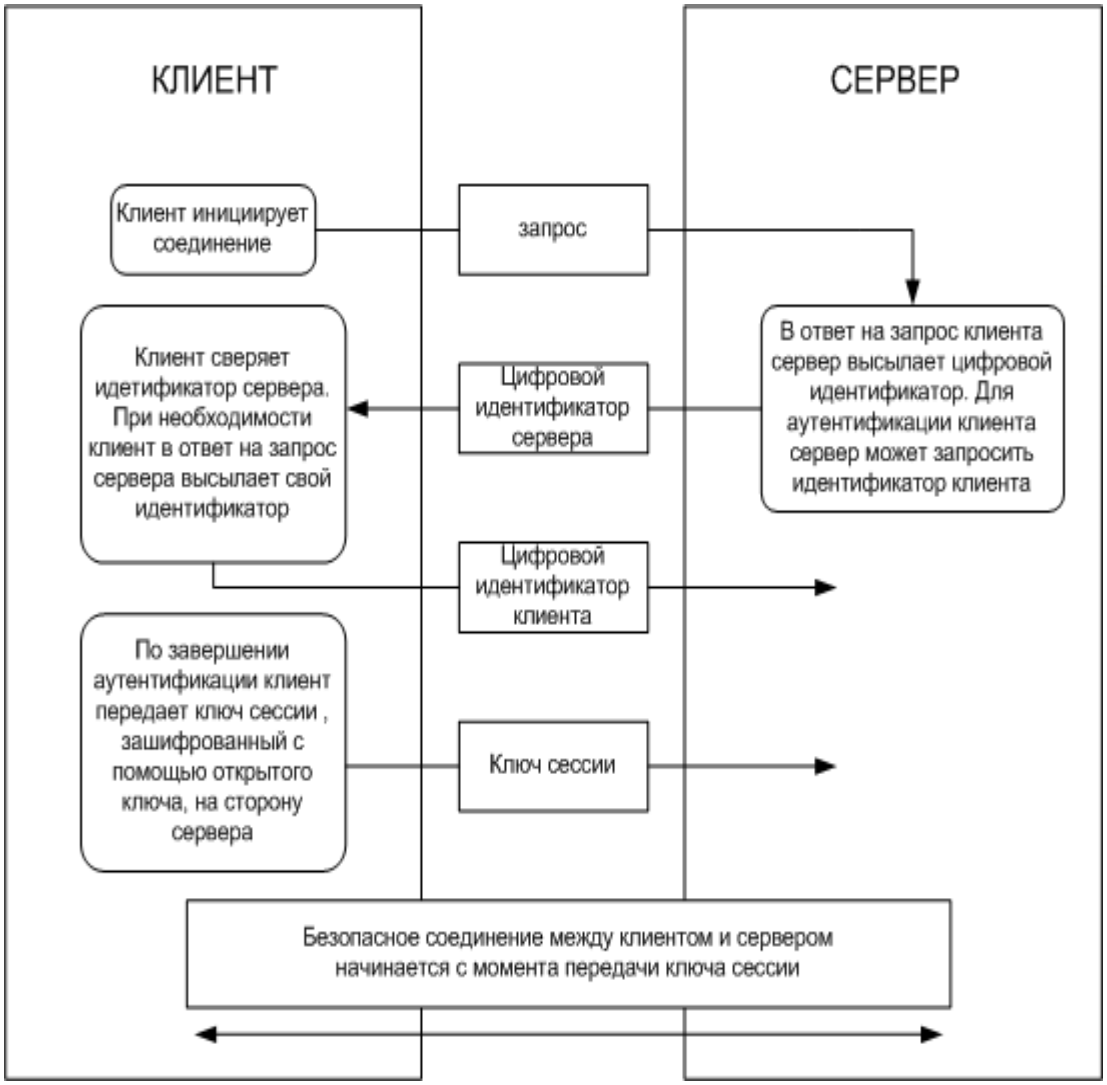
- Контроль целостности передаваемого документа.
- Защиту от изменений (подделки) документа:
- Невозможность отказа от авторства
- Доказательное подтверждение авторства документа

Использование цифровой подписи



- Если предположить, что владелец сумел сохранить тайну секретного ключа, тогда совпадение результатов вычисления доказывает, что никто не смог бы исказить файл после того, как он был подписан в цифровой форме.

Протокол SSL



Протокол SSL обеспечивает защищенный обмен данными за счет сочетания двух следующих элементов:

Аутентификация

Цифровой сертификат привязан к конкретному домену сети Интернет, а центр сертификации проводит проверки, подтверждающие подлинность.

Шифрование

Шифрование - это процесс преобразования информации в нечитаемый для всех вид, кроме конкретного получателя.

Спасибо за внимание

следующая лекция

Базы данных в WEB

ЛЕКЦИЯ 10

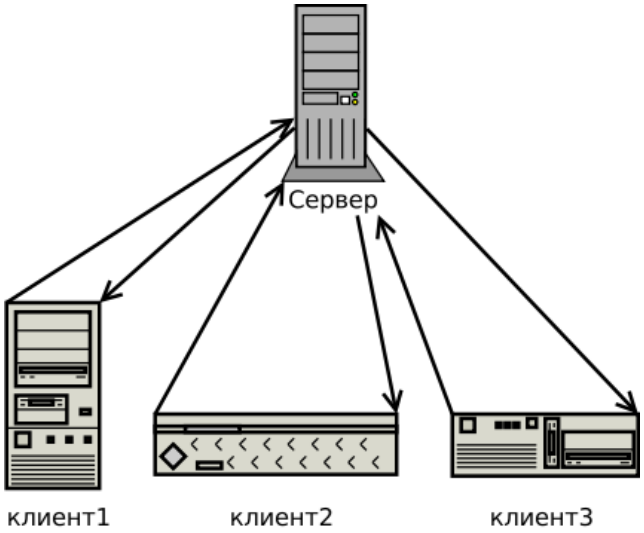
Базы данных в WEB

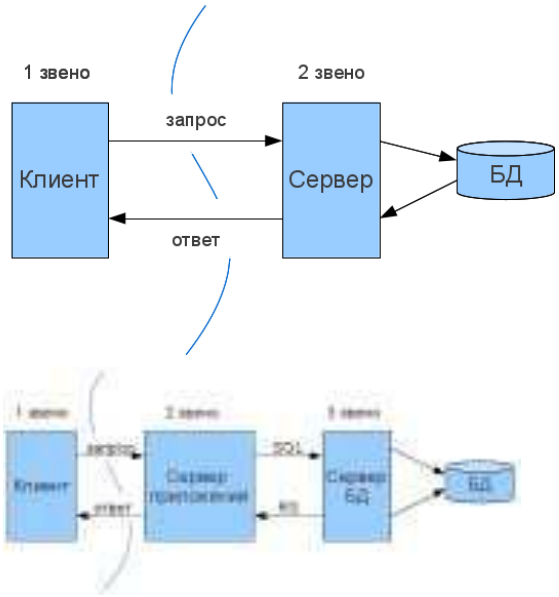
Содержание лекции 10

- Модель клиент – сервер.
- «Тонкий» сервер - «толстый» клиент.
- Многозвенная модель.
- Классическая трехзвенная модель.
- Модель с тонким клиентом. Понятие тонкого клиента.
- Преимущества трёхзвенной модели с тонким клиентом.

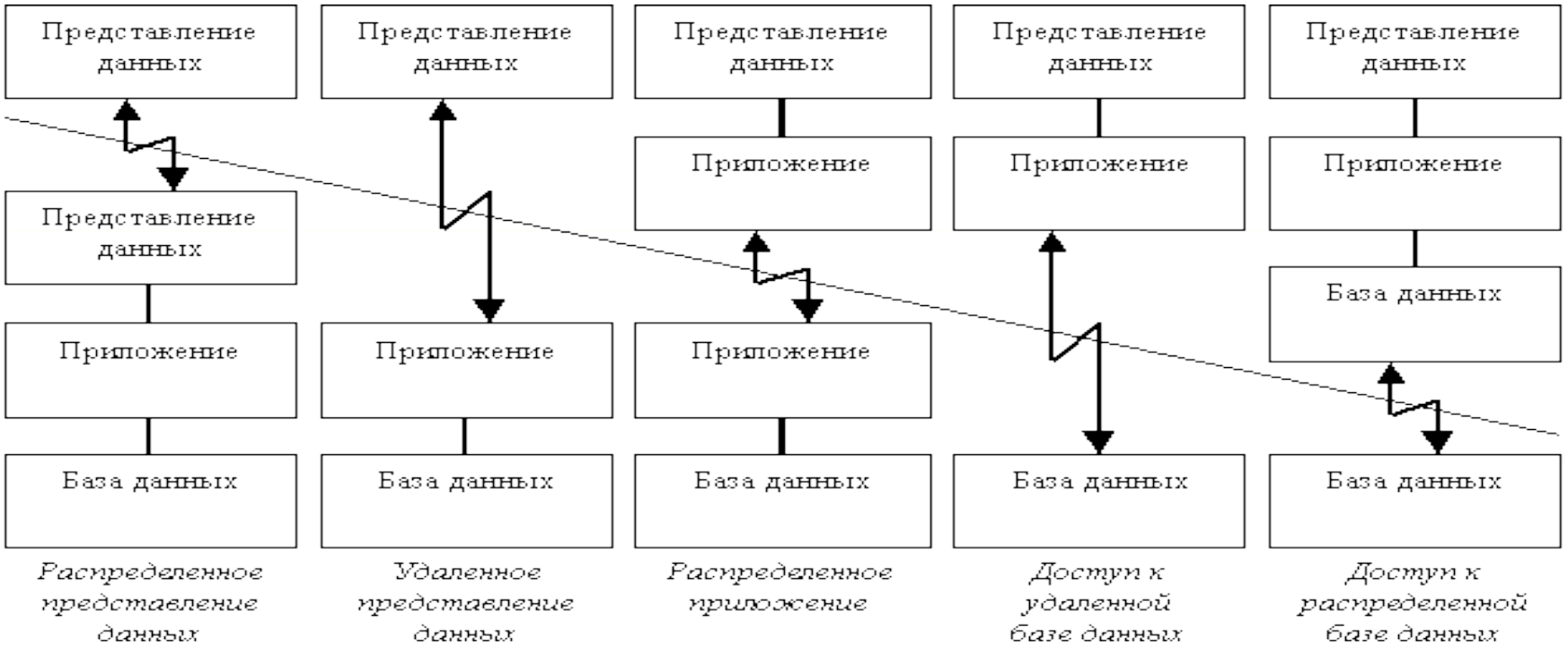
Клиент-серверная архитектура

Клиент-сервер — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.





Реализация модели клиент – сервер



Толстый клиент – тонкий сервер

Толстый клиент в архитектуре клиент-сервер — это приложение, обеспечивающее расширенную функциональность независимо от центрального сервера.

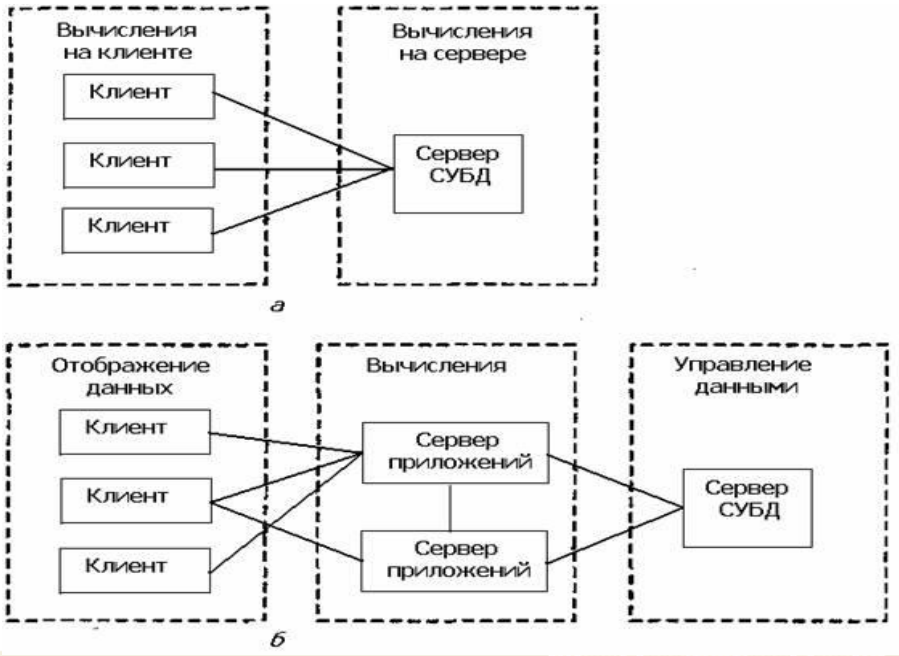
Цель - разгрузить сервер, сделать приложение более масштабируемым.

За сервером остается только следующий функционал:

а) безопасность;

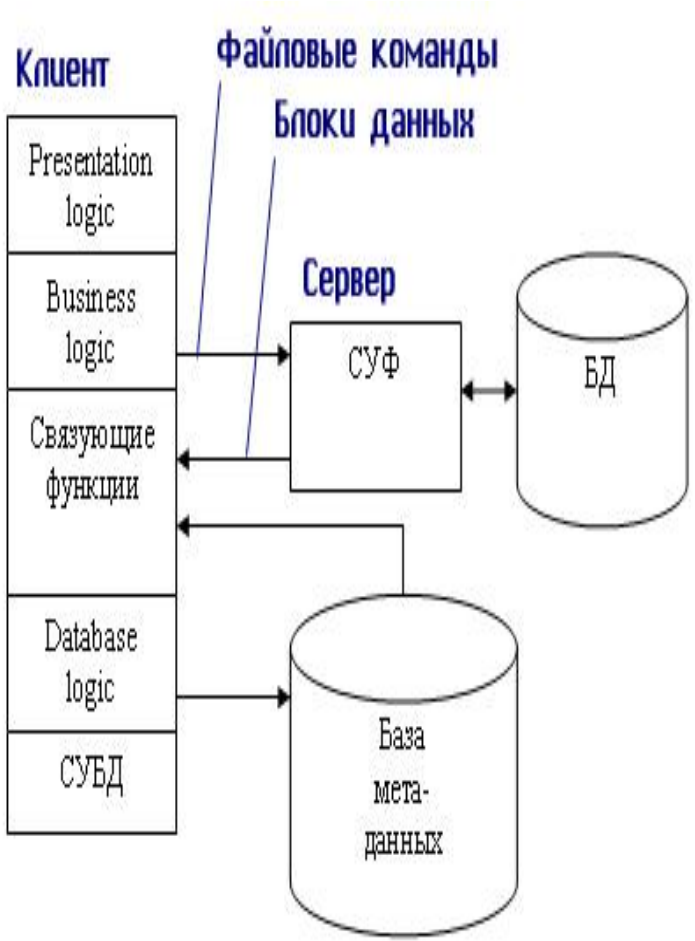
б) бизнес-логика;

в) проксирование запросов к другим ресурсам.



Двухуровневая модель организации БД

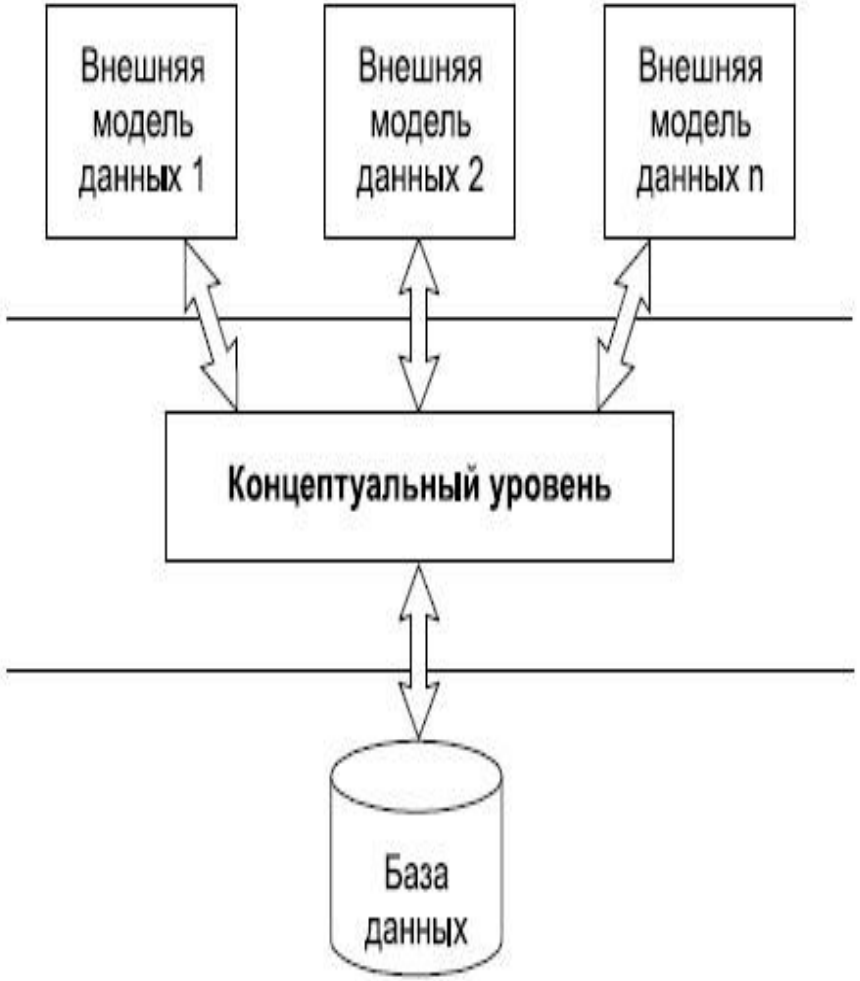
Перекачка информации с сервера на клиент производится до тех пор, пока не будет получен ответ на запрос клиента.



- Недостатки:
- высокий сетевой трафик, который связан с передачей по сети множества блоков и файлов, необходимых приложению;
- узкий спектр операций манипулирования с данными, который определяется только файловыми командами;
- отсутствие адекватных средств безопасности доступа к данным (защита только на уровне файловой системы).

Трёхуровневая модель

- Уровень внешних моделей — самый верхний уровень, где каждая модель имеет свое "видение" данных.
- Концептуальный уровень — центральное управляющее звено, здесь база данных представлена в наиболее общем виде
- Физический уровень — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.



Модель с тонким клиентом

Тонкие клиенты, работающие в терминальном режиме:

Протоколы, используемые тонкими клиентами:

- X11 — используется в Unix
- Telnet — мультиплатформенный
- SSH — мультиплатформенный защищённый аналог Telnet
- NX NoMachine — протокол X11 со сжатием данных
- Virtual Network Computing
- Citrix Independent Computing Architecture(ICA)

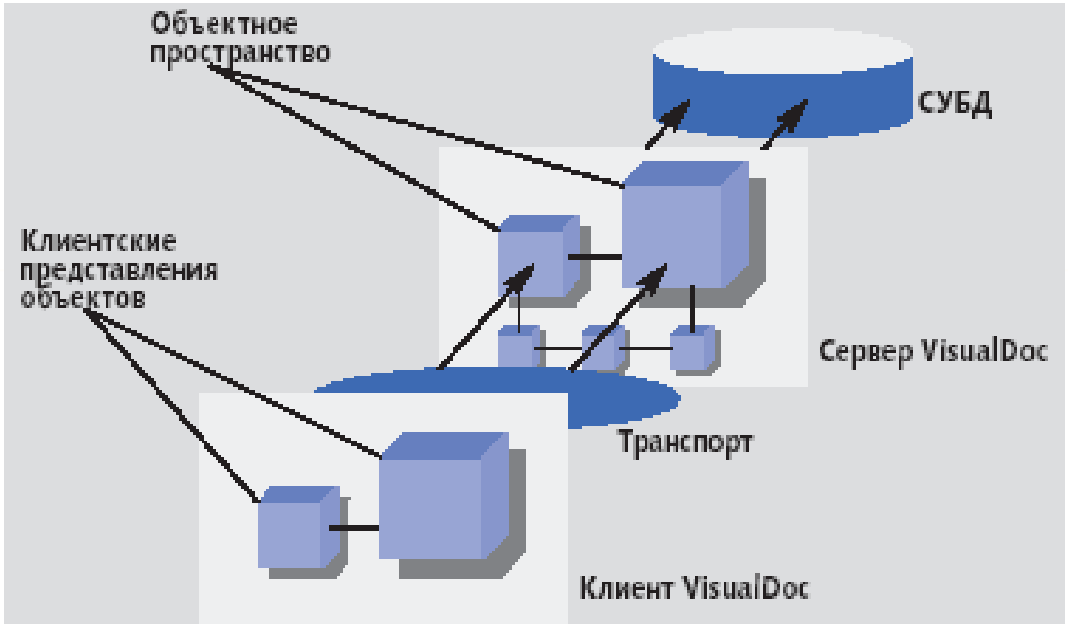
Примеры тонких клиентов:

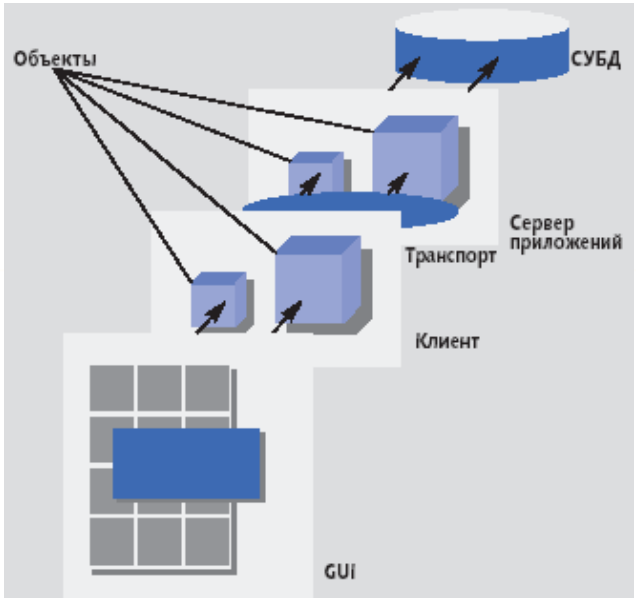
- Thinstation
- LTSP
- OpenThinClient
- Бездисковая станция
- Терминальный доступ
- Virtual Network Computing

Преимущества модели с тонким клиентом

- Основные преимущества в случае применения решений «тонкий клиент»:
- **снижение начальных затрат на приобретение** персональных компьютеров;
- **унификация** – все терминалы имеют одинаковый набор программного обеспечения;
- **простота первоначального внедрения** - нет необходимости настраивать каждый персональный компьютер в отдельности;
- **экономия времени системного администратора.**
- **Масштабируемость -** созданный единожды образ системы для работы всей группы пользователей позволяет при минимальных затратах поддерживать легко масштабируемую сеть.
- **безопасность и отказоустойчивость.** Все модификации операционной системы и прикладных программ никак не влияют ни на других пользователей, ни на образ, хранящийся на сервере.
- **защита от утечек информации** – нет локальных носителей – нет возможности делать копии документов на съемные носители информации.

Пример реализации тонкого клиента





Варианты выбора инфраструктуры на примере тонкого и толстого клиентов на основе критериев:

- скорость канала связи;
- вычислительную мощность и объем памяти клиентских машин и серверов;
- возможность установки на клиентские рабочие места приложений;
- степень унификации системного окружения клиентских рабочих мест;
- допустимое время ожидания реакции.

Спасибо за внимание

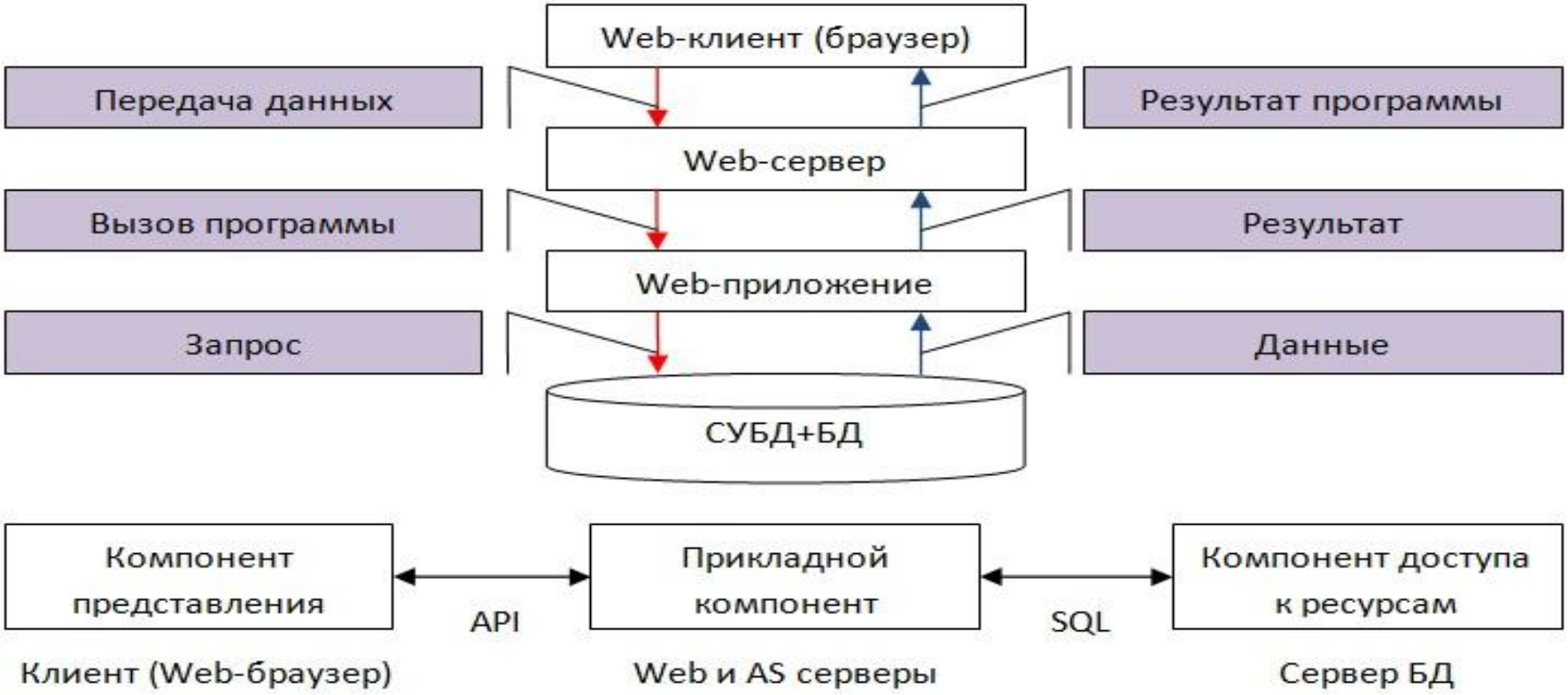
следующая лекция

Реализация баз данных в WEB

ЛЕКЦИЯ 11

Реализация баз данных в WEB

Технология обращения к БД из WEB



- Режимы работы:
  - без сохранения состояния;
  - сессионный.

WEB сервер

Фактически Web-сервер включает несколько других серверов, реализующих необходимые протоколы.

- HTTP (Hypertext Transfer Protocol) – протокол передачи гипертекста
- FTP (File Transfer Protocol) – протокол передачи файлов

- NNTP (Network News Transfer Protocol) – сетевой протокол передачи новостей

- SMTP (Simple Mail Transfer Protocol) – простой протокол передачи почты

Механизм доступа

- Механизм доступа к БД на стороне сервера реализуется за счет наличия стандартизованных средств:

- Поддержки диалоговых форм на уровне гипертекстового документа язык HTML. С каждым документом связан URL, для доступа к которым используется объектно-ориентированный протокол HTTP.

- Возможности запуска серверных программ, взаимодействие которых происходит через стандартный интерфейс CGI или прикладные интерфейсы Web-сервера.

Средства доступа к базам данных

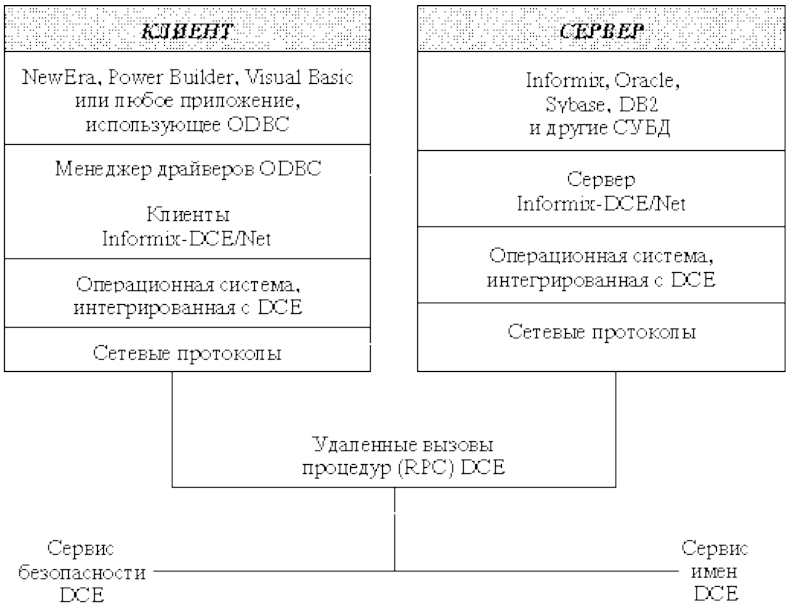
Системные:

- ODBC – доступ через стандартный системный интерфейс.
- JDBC – доступ из приложений Java.
- API –компоненты языков программирования.

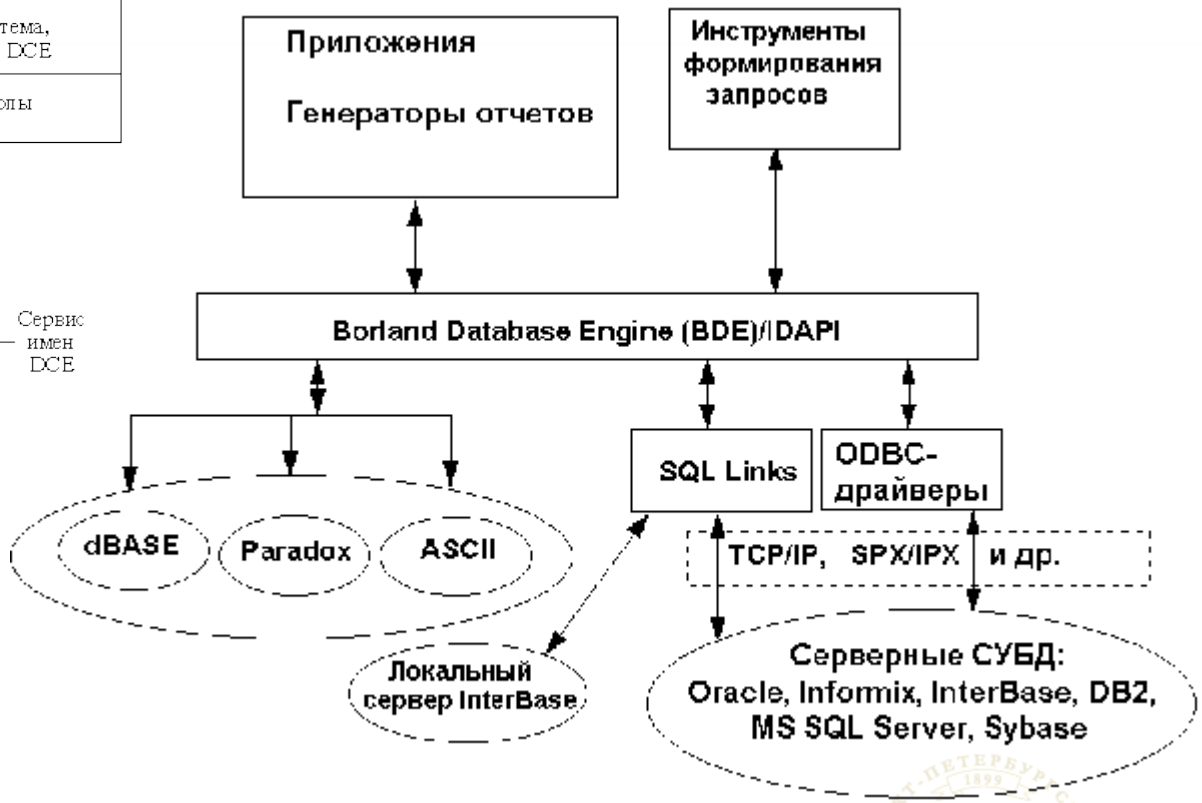
Стандартные функции PHP для работы с MySQL:

- mysql_connect()
- mysql_select_db()
- mysql_close()
- mysql_query()
- mysqlaffected_rows()
- mysql_num_rows()
- mysql_result()
- mysql_fetch_row() mysql_fetch_array()

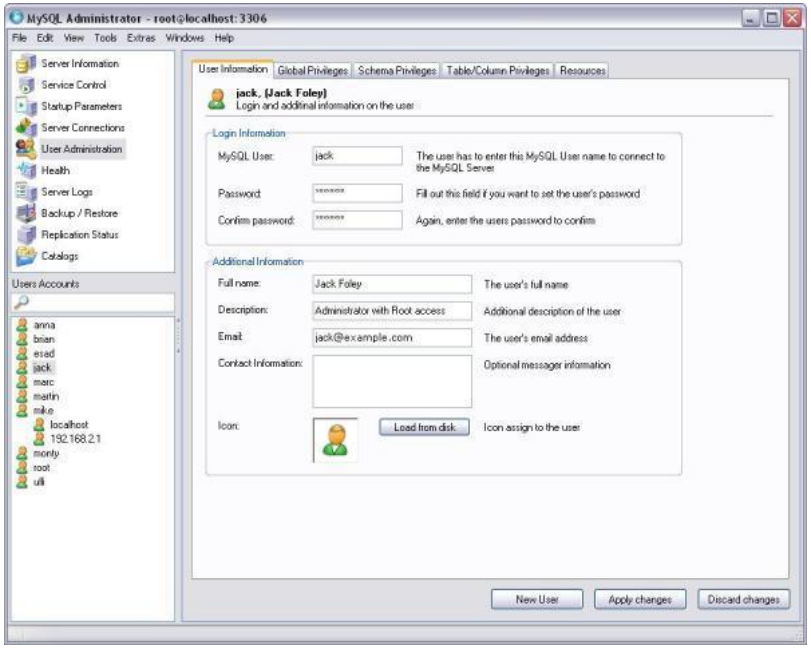
Схема взаимодействия клиента и серверного приложения

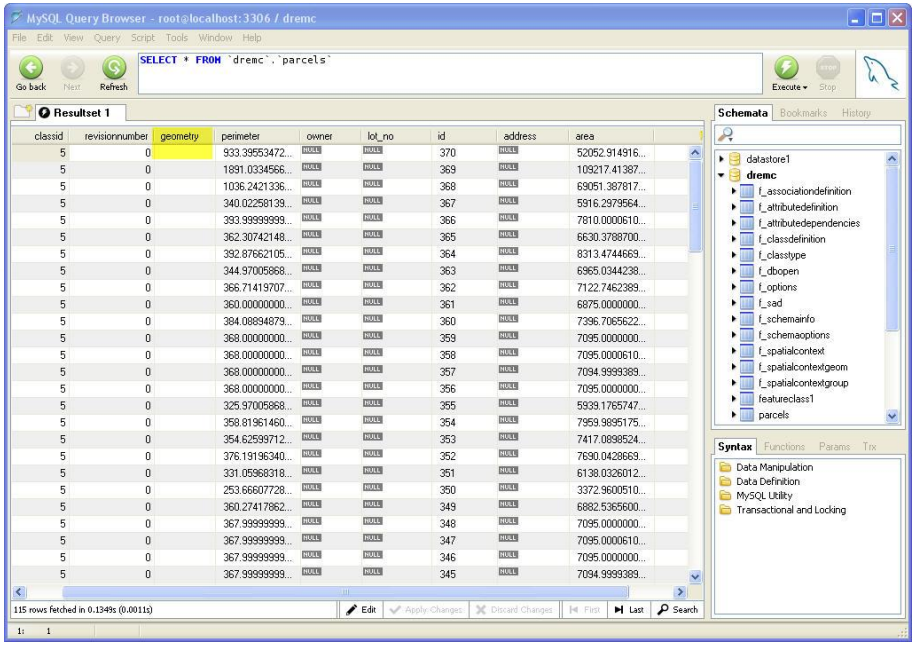


Доступ к БД через ODBC



Пример доступа к СУБД из WEB: MySql Administrator





Утилита может работать как с локально установленным сервером, так и подключаться к удаленному серверу, одновременно можно работать с неограниченным количеством серверов - для каждого из них будет запущена отдельная копия приложения.

Спасибо за внимание

следующая лекция

Распределённые СУБД

ЛЕКЦИЯ 12

Распределённые СУБД

Содержание лекции 12

- Типы разделения данных в узлах распределённой системы.
- Кластеры и географически распределённые системы.
- Способы синхронизации данных.
- Репликация данных.
- Проблемы распределённых баз данных.
- Пример реализации распределённой базы данных.

Распределённые СУБД

Распределённые базы данных — совокупность логически взаимосвязанных баз данных, распределённых в компьютерной сети.

Типы распределённых СУБД:

- Мультибазы - баз данных с глобальной схемой.
- Федеративные базы данных. В каждой базе поддерживается локальная схема импорта-экспорта данных
- Мультибазы с общим языком доступа — распределённые среды управления с технологией «клиент-сервер»

Критерии распределённости

- Локальная автономность. Локальные данные принадлежат локальным узлам и управляется администраторами локальных БД

- Отсутствие опоры на центральный узел. В системе не должно быть узла, без которого система не может функционировать.

- Непрерывное функционирование. Удаление или добавление узла не должно требовать остановки системы в целом.

- Независимость от местоположения. Пользователь должен получать доступ к любым данным в системе, независимо от того, являются эти данные локальными или удалёнными.

- Независимость от фрагментации. Доступ к данным не должен зависеть от наличия или отсутствия фрагментации и от типа фрагментации.

- Независимость от репликации. Доступ к данным не должен зависеть от наличия или отсутствия реплик данных.

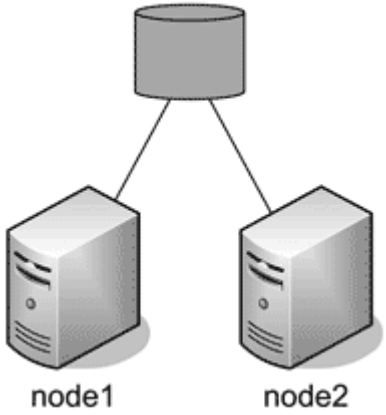
Критерии распределённости

- Обработка распределенных запросов. Система должна автоматически определять методы выполнения соединения (объединения) данных .
- Обработка распределенных транзакций. Протокол обработки распределённой транзакции должен обеспечивать выполнение четырёх основных свойств транзакции : атомарность , согласованность , изолированность и продолжительность .
- Независимость от типа оборудования . СУРБД должна функционировать на оборудовании с различными вычислительными платформами .
- Независимость от операционной системы . СУРБД должна функционировать под управлением различных ОС .
- Независимость от сетевой архитектуры . СУРБД должна быть способной функционировать в сетях с различной архитектурой и типами носителя .
- Независимость от типа СУБД .СУРБД должна быть способной функционировать поверх различных локальных СУБД , возможно , с различными моделями данных.

Методы поддержки распределенности

1. Фрагментация – разбиение БД или таблицы на несколько частей и хранение этих частей на разных узлах РБД .
2. Репликация – создание и хранение копий одних и тех же данных на разных узлах РБД .
3. Распределенные ограничения целостности – ограничения, для проверки выполнения которых требуется обращение к другому узлу РБД .
4. Распределенные запросы – это запросы на чтение, обращающиеся более чем к одному узлу РБД.
5. Распределенные транзакции – команды на изменение данных , обращающиеся более чем к одному узлу РБД.

Кластеры и географически распределённые системы



Способы распределения:
-локальный
- сетевой

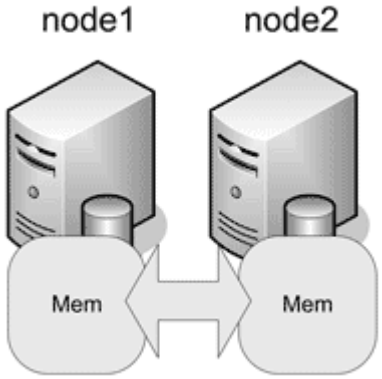
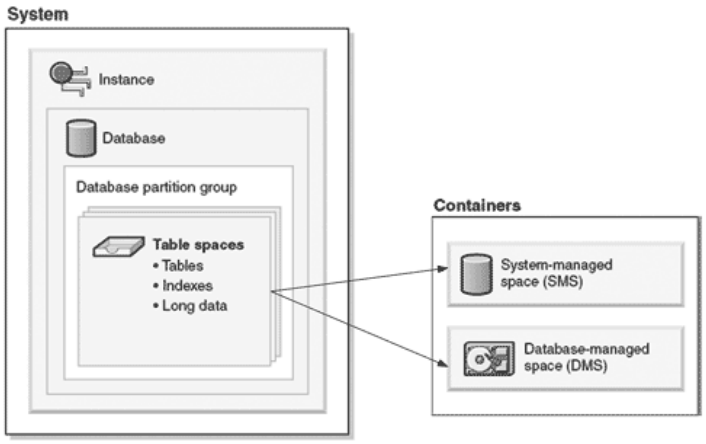
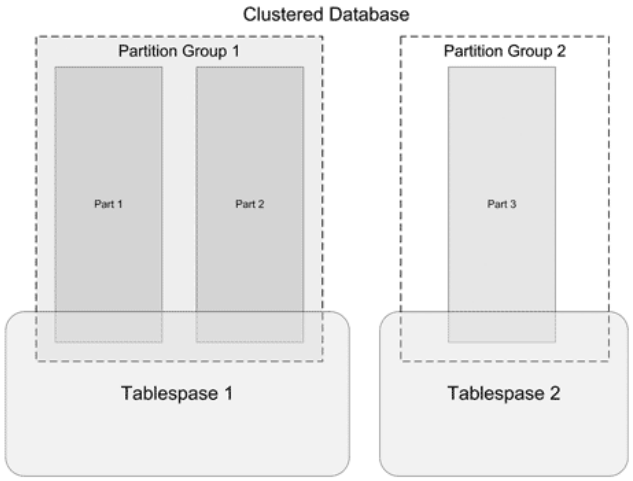


Схема кластеризации DB2



Технология кластеризации DB2



Репликация данных

Репликация – это поддержание двух и более идентичных копий данных на разных узлах РБД.

Реплика может включать всю базу данных (полная репликация), одно или несколько взаимосвязанных отношений или фрагмент отношения .

Достоинства репликации:

– повышение доступности и надежности данных ;

– повышение локализации ссылок на реплицируемые данные.

Недостатки репликации:

– сложность поддержания идентичности реплик;

– увеличение объема памяти для хранения данных.

Поддержание идентичности реплик называется распространение изменений и реализуется службой тиражирования.

Преимущества и недостатки РБД

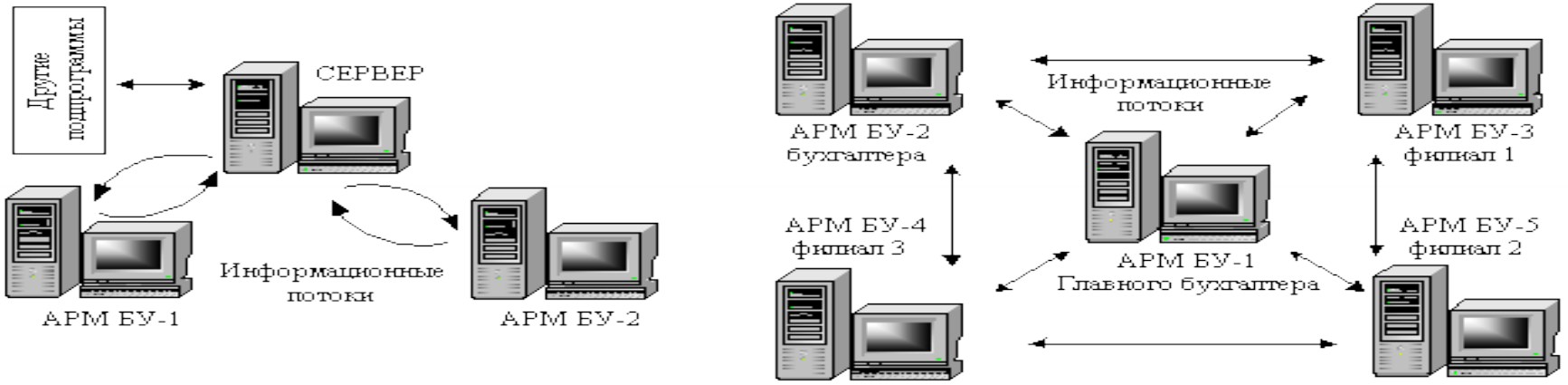
Преимущества:

- Модульность системы
- Экономические выгоды
- Повышение производительности
- Повышение надежности
- Повышение доступности данных
- Разделяемость и локальная автономность
- Отражение структуры организации

- Недостатки:
- Усложнение процедуры разработки базы данных
- Недостаток опыта
- Отсутствие стандартов
- Усложнение контроля за целостностью данных
- Проблемы защиты
- Увеличение стоимости
- Повышение сложности

Примеры распределённой базы данных

Пример структуры аппаратных компонент



Пример концептуальной схемы распределения данных

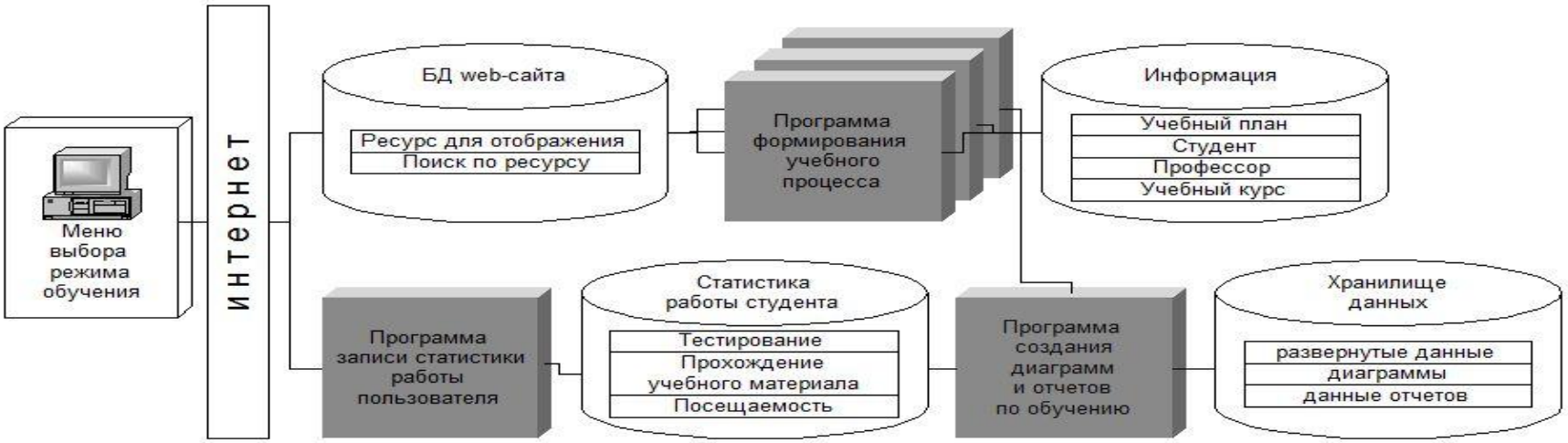


Рис. 1. Подсистема поддержки дистанционного обучения

Спасибо за внимание

следующая лекция

Аппаратная реализация машин баз данных

ЛЕКЦИЯ 13

Аппаратная реализация машин баз данных

Содержание лекции 13

- Особенности архитектур ЭВМ ориентированных на поддержку баз данных.
- Особенности архитектуры.
- Подсистема ввода/вывода.
- Архитектура Oracle Exadata Database Machine
- Архитектура IBM zArchitecture и IBM eServer zSeries (System/390)

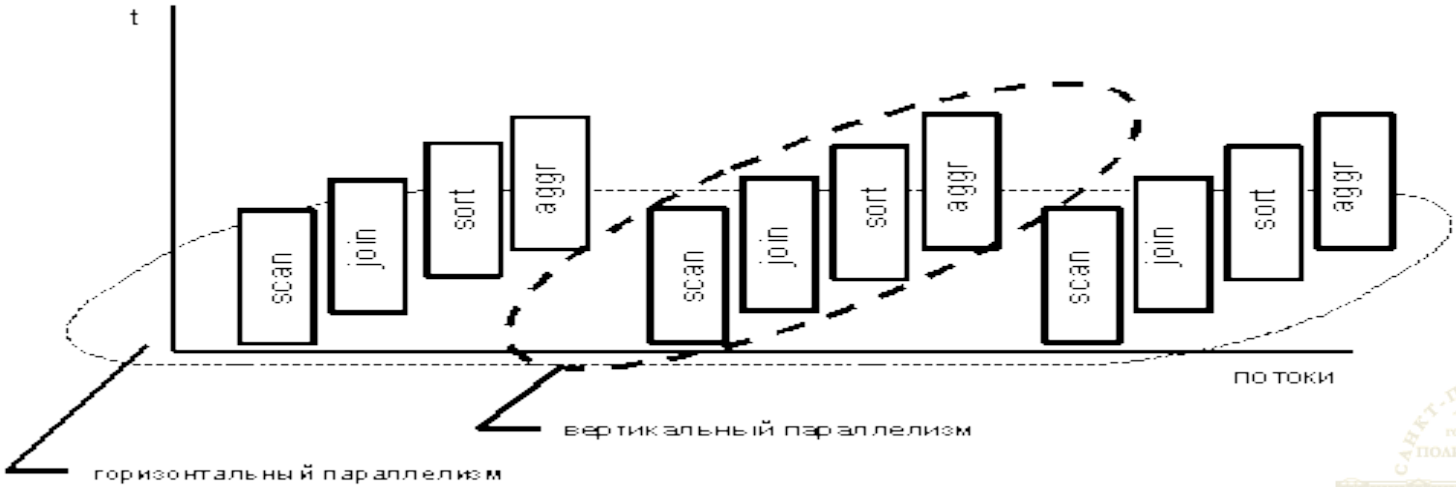
Машины баз данных

Машина баз данных – специализированный компьютер для эффективного хранения данных и быстрого выполнения запросов.

Способ решения – повысить степень параллелизма.

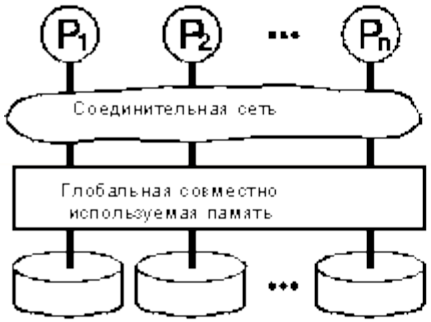
Ускорения

Пакетная расширяемость

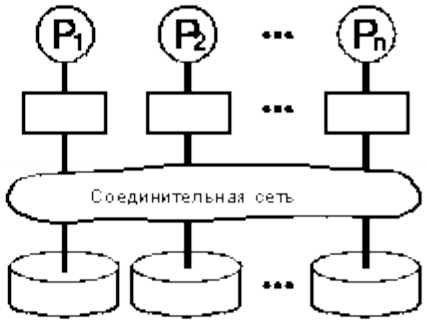


Технологии обеспечения параллелизма

- Аппаратная архитектура. Тенденция к машинам без совместного использования ресурсов



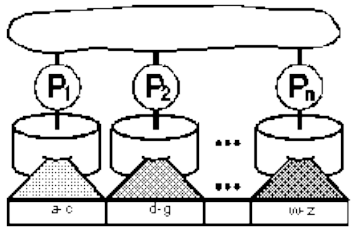
Мультипроцессор с разделением полей



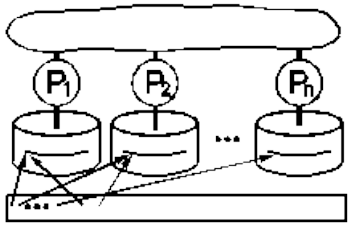
Мультипроцессор с разделением дисков

- Программные решения на основе SQL с использованием параллельного потока данных:

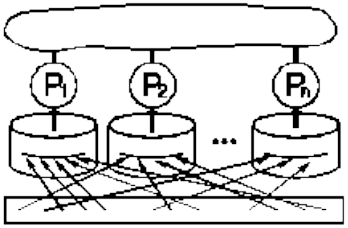
✓ разделение данных;
✓ параллелизм внутри реляционных операторов.
✓ специальные параллельные реляционные операторы.



Упорядоченное разделение



Круговое разделение

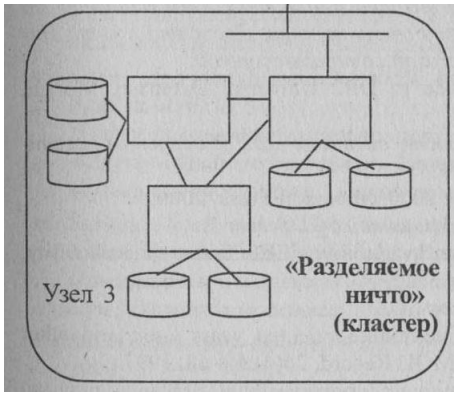


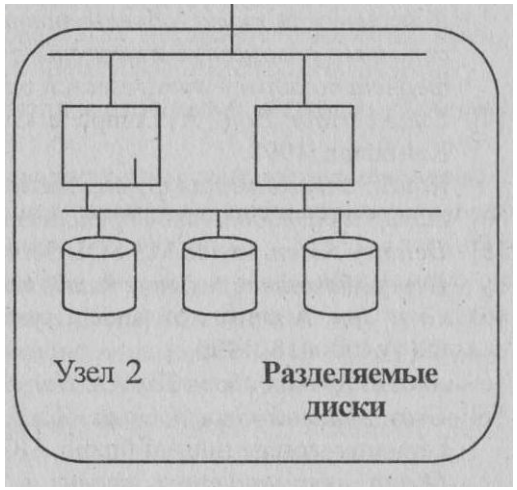
Хаотичное разделение

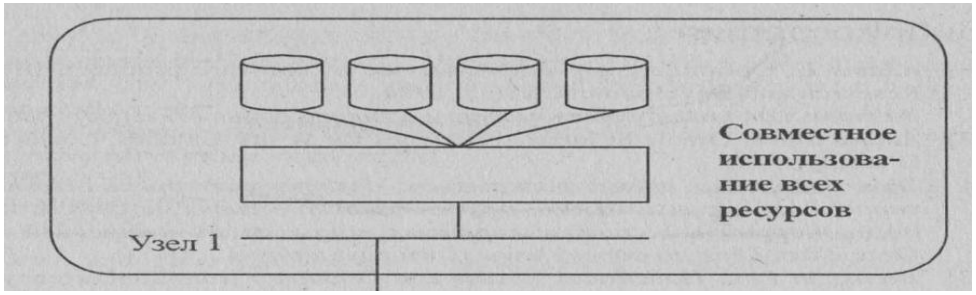
Примеры:
- Teradata
- Tandem Nonstop SQL
- Gamma
- Bubba

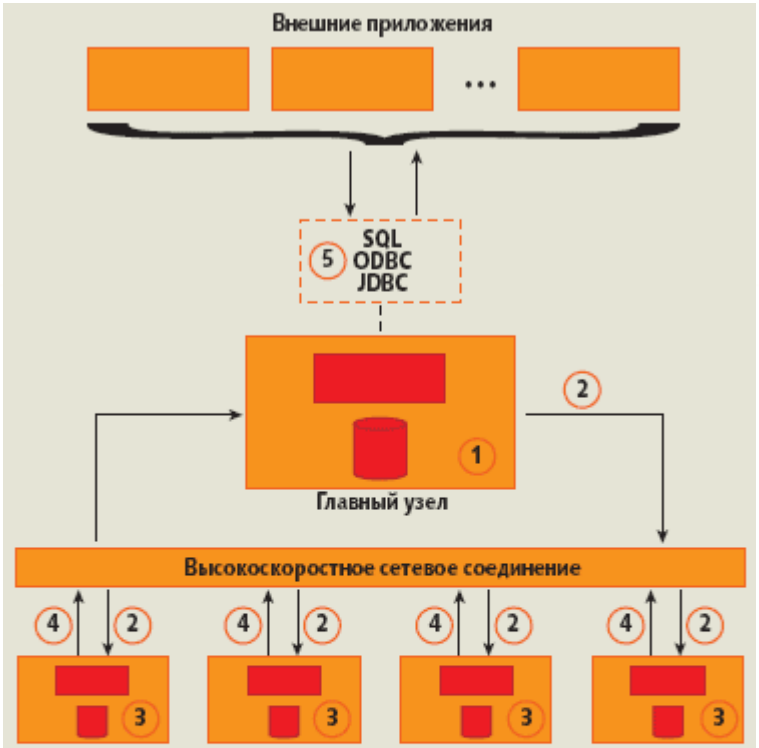
Настройка оборудования ввода-вывода

- 1) конфигурирование дискового массива;
- 2) оптимизация использования кеш-памяти;
- 3) планирование ёмкости и задание размеров;
- 4) конфигурирование логических томов.









1. Запрос, поступивший от внешнего приложения, анализируется на главном узле системы, где происходит оптимизация плана выполнения обработки и декомпозиция запроса.
2. Фрагменты запроса передаются на обработку периферийным узлам.
3. На периферийных узлах (или непосредственно в запоминающих устройствах) осуществляется фильтрация и основной объем обработки данных.
4. Промежуточные результаты возвращаются главному узлу, где осуществляется их сборка.
5. Окончательный результат возвращается инициировавшему запрос приложению.

Проблемы аппаратной компоненты машин баз данных

- Архитектура параллельных систем баз данных, создаваемых из компонентов общего назначения, предполагает перемещение больших объемов сведений от запоминающих устройств к процессорам и в обратном направлении.

- Основная причина медленной обработки сложных запросов — недостаточная пропускная способность тракта «диск — память — процессор».

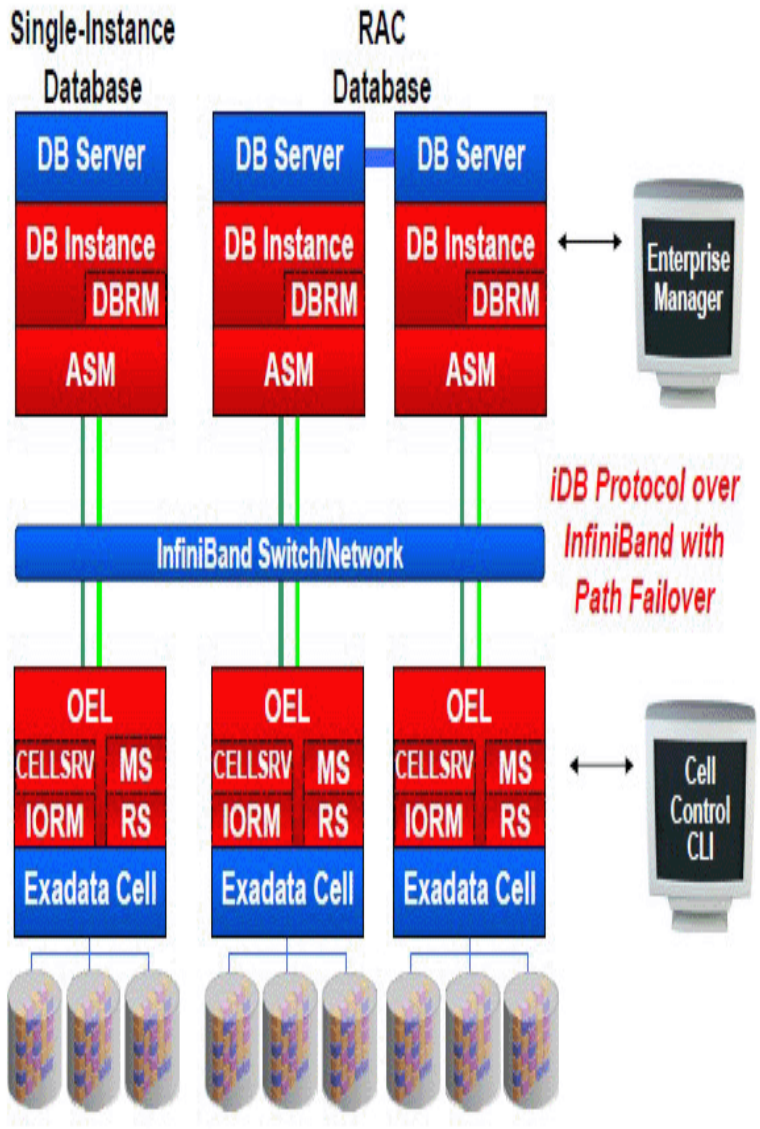
- Необходимость интеграции и тонкой настройки оборудования и программных компонентов обусловливает исключительно высокие требования к профессиональной подготовке обслуживающего персонала и требует постоянной внешней поддержки. В связи с этим растет стоимость реализации, настройки, эксплуатации и развития системы.

- Чрезвычайно высокая стоимость хранилищ данных сдерживает полноценное применение средств бизнес-анализа в средних, а тем более мелких компаниях.

Архитектура Oracle Exadata Database Machine

Программно-аппаратные средства:

- два восьмипроцессорных сервера базы данных с общим количеством в 128 вычислительных ядер процессора Intel и 2 ТБ памяти;
- 14 серверов хранения данных Exadata Storage Server с общим количеством в 168 вычислительных ядер процессора Intel и до 336 ТБ неформатированного дискового пространства;
- свыше 5 ТБ кэш-памяти Exadata Smart Flash Cache для хранения часто запрашиваемых «горячих» данных в целях обеспечения высокой пропускной способности и сверхмалого времени отклика при обработке транзакций;
- несколько уровней компрессии для управления большими объемами данных и сокращения ввода/вывода для данных OLTP-приложений и хранилищ данных;
- внутренний интерфейс 40 Gigabit InfiniBand;
- внешний интерфейс 10 Gigabit Ethernet.

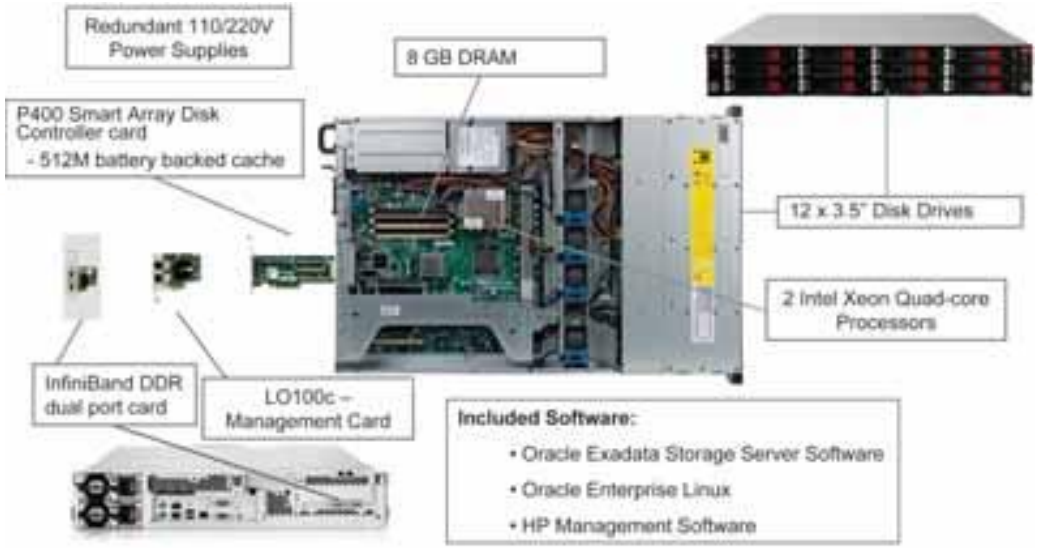


Реализаця архитектуры

Архитектура HP Oracle Exadata Storage Server в составе HP Oracle Database Machine

HP Oracle Database Machine содержит:

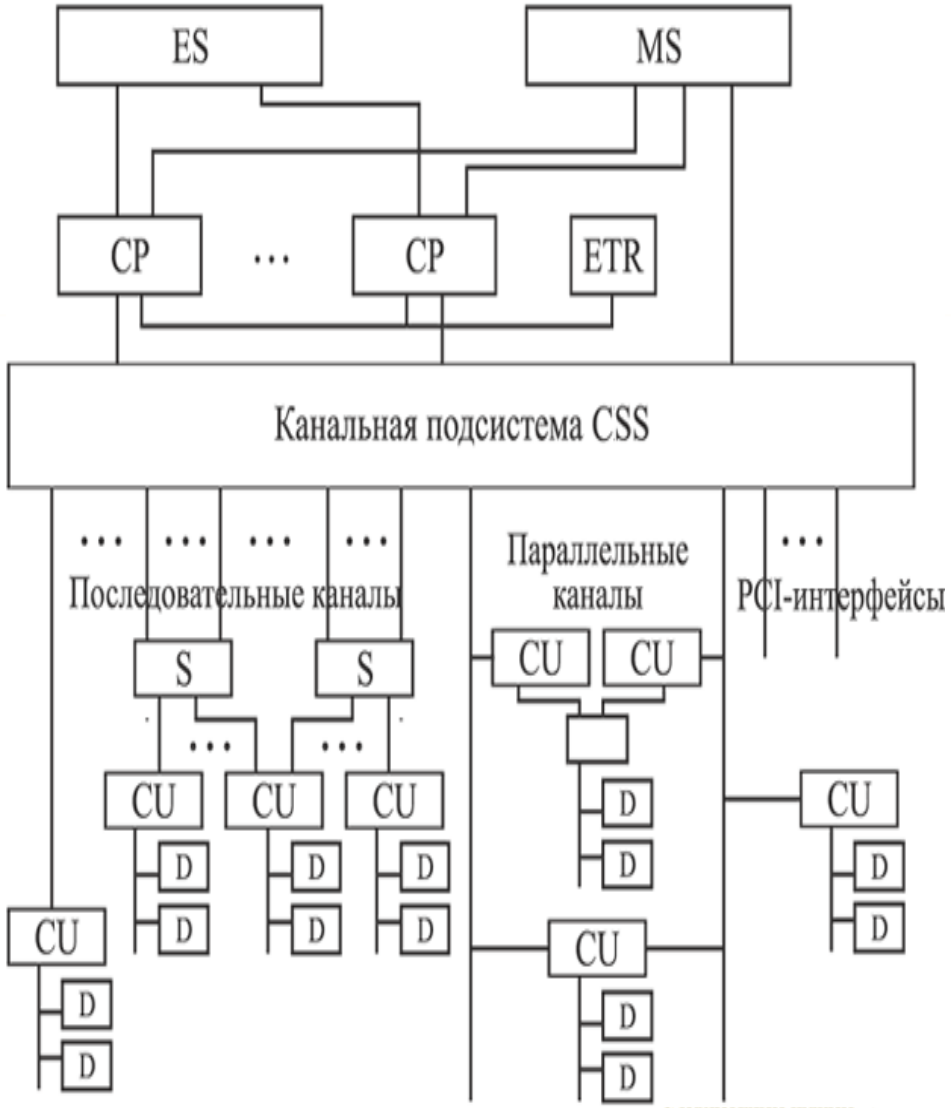
- 8 DL360 Oracle Database серверов (2 quad-core Intel Xeon, 32GB RAM);
- 14 Exadata Storage Cells с дисками SAS или SATA, до 21 Тбайт и 46 Тбайт некомпрессионных пользовательских данных;
- 4 InfiniBand-коммутатора по 24 порта.



Результаты тестирования производительности

Архитектура серверов IBM System Z

Основными компонентами z/Architecture серверов zSeries на уровне программистской модели являются:
- один или несколько центральных процессоров (Central Processor - CP);
основная память (Main Storage - MS);
расширенная память (Expanded Storage - ES);
- внешний таймер (External Time Reference - ETR);
канальная подсистема (Channel Subsystem - CSS);
- контроллеры (Control Unit - CU) периферийных устройств (I/O Device - D)
- канальные пути или каналы, соединяющие канальную подсистему с CU периферийных устройств, и средства их коммутации (S).



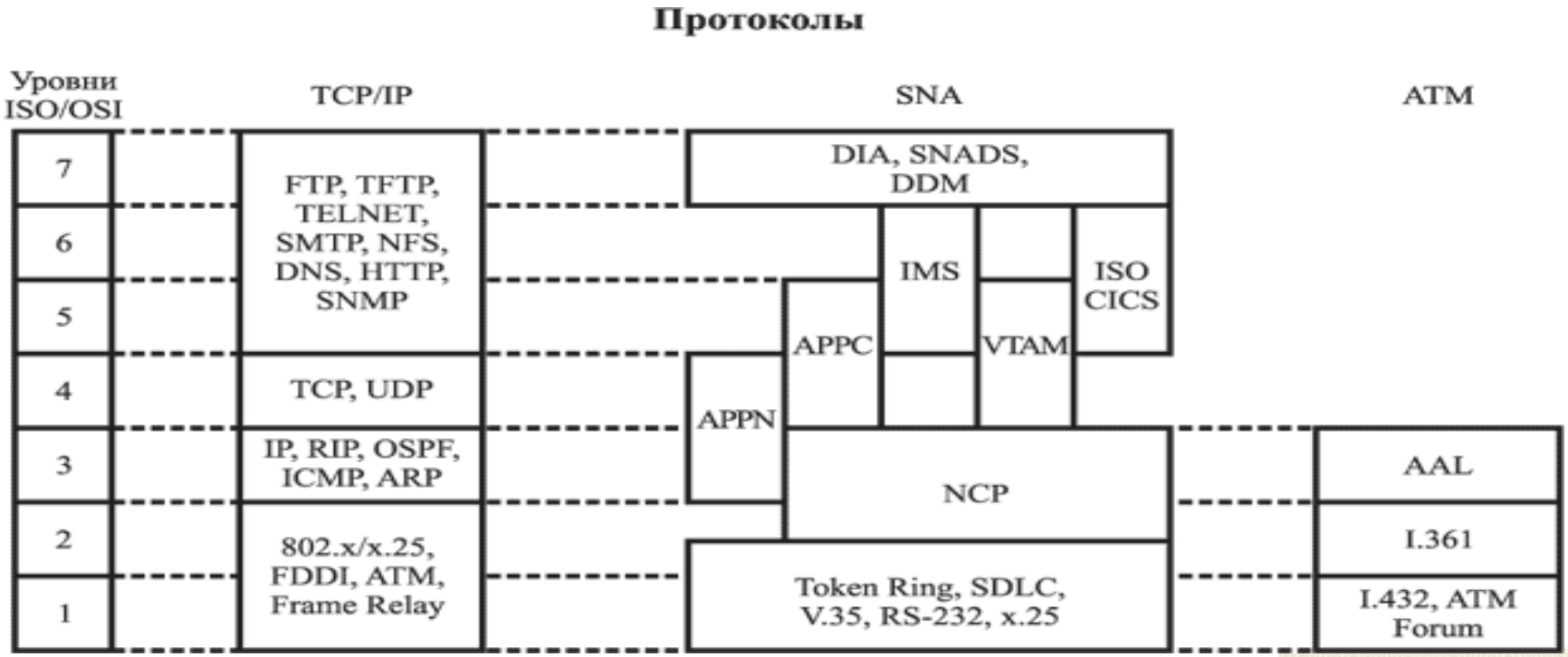
Коммуникационные протоколы System Z

Стек SNA (прикладной):

- DIA (Document Interchange Architecture) - определяет стандарты обмена документами между разнородными вычислительными системами;
- SNADS (SNA Distributed Service) - управляет распространением документов и сообщений
- DDM (Distributed Data Management) - обеспечивает прозрачный удаленный доступ к файлам

Стек ATM (физический):

- временной синхронизации между отправителем и получателем;
- скорости;
- режима соединения.



Спасибо за внимание

следующая лекция

Аппаратные средства надёжного хранения данных

ЛЕКЦИЯ 14

Аппаратные средства надёжного хранения данных

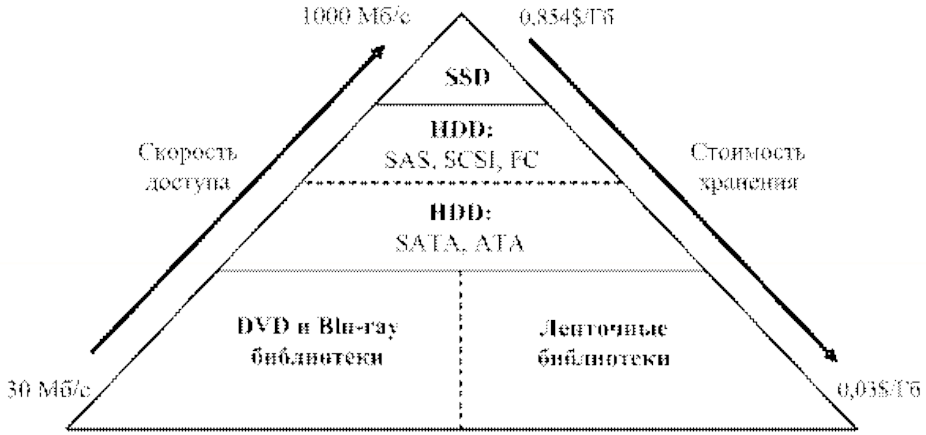
Содержание лекции 14

- Аппаратные средства хранения данных.
- Понятие RAID-массива.
- Уровни RAID.
- Дисковые подсистемы типа IBM ESS Shark.
- Архитектура SAN.
- Библиотеки магнитных лент и CD.
- Подсистемы хранения IBM eServer iSeries (AS/400) и OS/400.

Аппаратные средства хранения больших объёмов данных

Типы устройств:

- магнитную ленту,
- CD, DVD Blu-ray диски,
- накопители на жёстких магнитных дисках,
- накопители на флэш-памяти.



Способ записи:

- однократная;
- многократная.

Параметры сравнения:

- стоимость хранения,
- скорость доступа к данным,
- скорость поиска файла,
- надёжность хранения.

Носитель	Стоимость хранения 1 Гб	Скорость передачи данных
Магнитная лента LTO-5	0,033$	180 Мб/с
Магнитная лента IBM 3592	0,06$	250 Мб/с
CD-R	0,24$	7,62 Мб/с
DVD-R	0,063$	31,7 Мб/с
Blu-ray	0,08$	54 Мб/с
HDD	0,055$	200 Мб/с*
SSD	0,854$	1000 Мб/с*

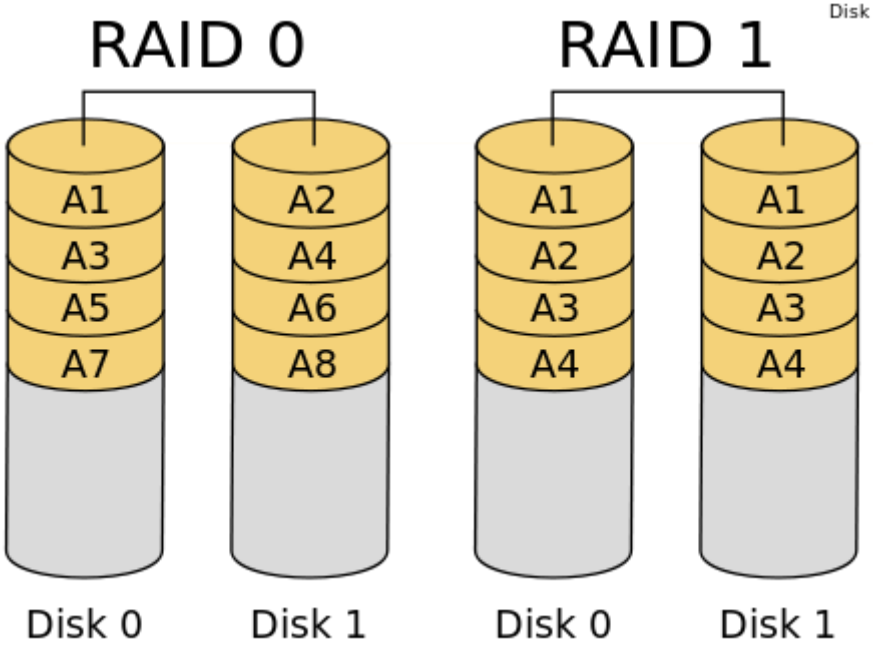
Понятие RAID-массива

- RAID — массив из нескольких дисков, управляемых контроллером, связанных между собой скоростными каналами передачи данных и воспринимаемых внешней системой как единое целое.

- **RAID 0** — дисковый массив повышенной производительности с чередованием, без отказоустойчивости;
- **RAID 1** — зеркальный дисковый массив;
- **RAID 2** зарезервирован для массивов, которые применяют код Хемминга;
- **RAID 3 и 4** — дисковые массивы с чередованием и выделенным диском чётности;
- **RAID 5** — дисковый массив с чередованием и «невыделенным диском чётности»;
- **RAID 6** — дисковый массив с чередованием, использующий две контрольные суммы, вычисляемые двумя независимыми способами;
- **RAID 10** — массив RAID 0, построенный из массивов RAID 1;
- **RAID 50** — массив RAID 0, построенный из массивов RAID 5;
- **RAID 60** — массив RAID 0, построенный из массивов RAID 6.

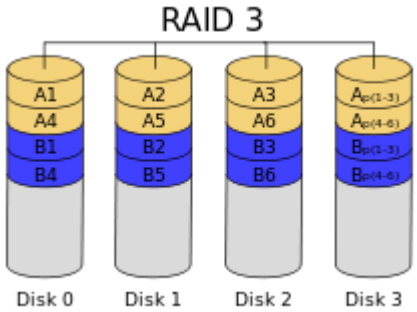
Уровни RAID

Базовые уровни

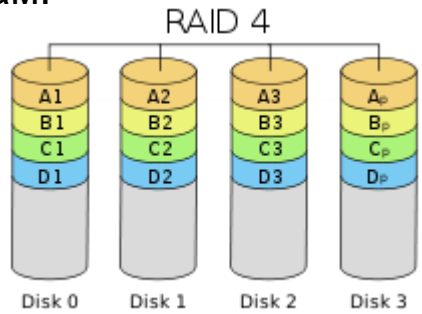


RAID 0 — дисковый массив из двух или более дисков без резервирования.

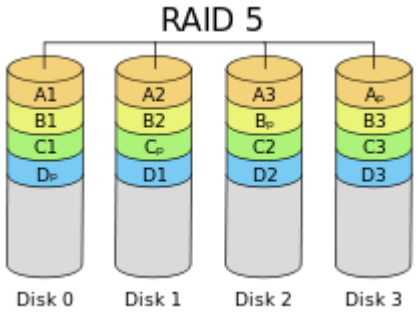
RAID 1 — массив из двух дисков, являющихся полными копиями друг друга.



RAID 3 — из дисков данные разбиваются на байты размером меньше сектора и распределяются по дискам.

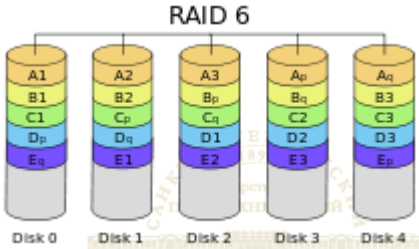


RAID 4 — RAID 3, но отличается от него тем, что данные разбиваются на блоки, а не на байты.



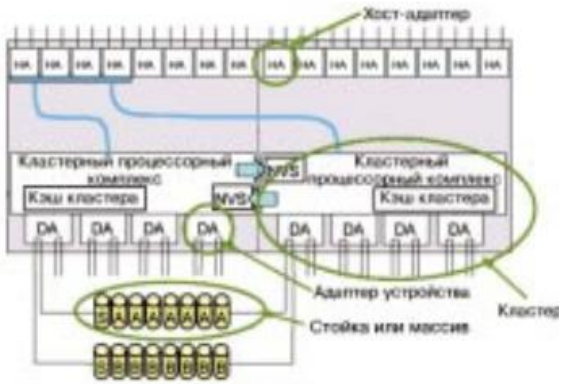
RAID 5 — RAID 3, но блоки данных и контрольные суммы циклически записываются на все диски массива.

RAID 6 — RAID 5, но под контрольные суммы выделяется ёмкость 2-х дисков, рассчитываются 2 суммы по разным алгоритмам.

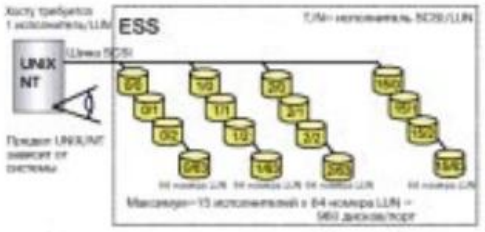


Дисковые подсистемы типа IBM ESS Shark

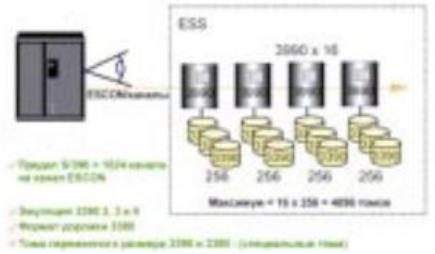
Реализация аппаратных функций:
-Flash Copy;
-Concurrent Copy;
-Per to Per Copy;
-Extended Remote Copy.



Вместимость системы	от 420 Гбайт до 11 Тбайт
Размер кэш-памяти	до 16 Гбайт
Внутренняя пропускная способность	более 1000 Мбайт/с 64 параллельно работающих интерфейса к дискам
Внешние соединения	до 32 параллельно работающих внешних интерфейса SCSI, ESCON; до 16 портов Fibre Channel



реализованных на базе Web.
Если представить себе ESS с точки зрения



Последовательное чтение из кэш (блоки 64К)	230 МБ/с
Последовательное чтение с диска (64К)	160-185 МБ/с
Последовательная запись на диск (64К)	145 МБ/с
Чтение при 100% попадании в кэш (блоки 4К), SCSI интерфейсы	32000 IO/c
Чтение при 100% попадании в кэш (4К) для S/390	24100 IO/c
Случайное чтение без кэш (4К)	9500 IO/c
Случайное чтение/запись 70/30 (4К), 50% попаданий в кэш, SCSI интерфейсы	12000 IO/c
Случайное чтение/запись 70/30 (4К), 50% попаданий в кэш, S/390	11300 IO/c
Случайная запись на диск (4К) минуя кэш	3000 IO/c
Запись 100% в кэш (4К)	12000 IO/c

Архитектура SAN

SAN является высокоскоростной сетью передачи данных, предназначенной для подключения серверов к устройствам хранения данных.

Технологическая основа SAN —

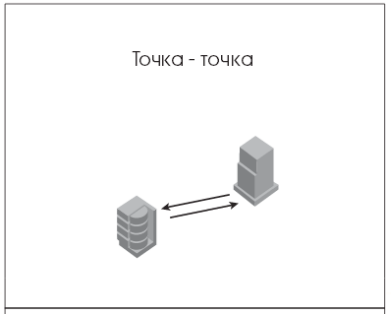
протокол Fibre Channel

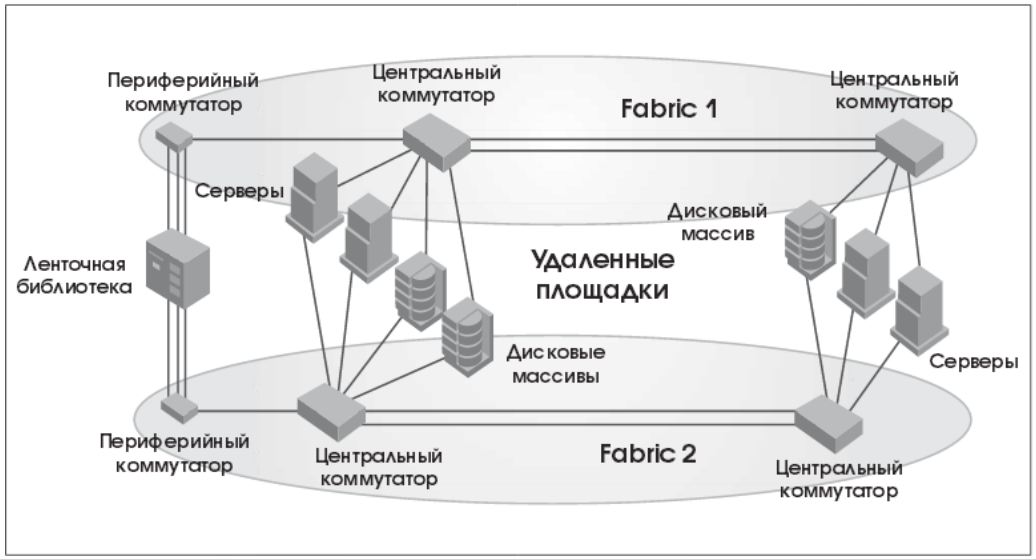
Компоненты SAN

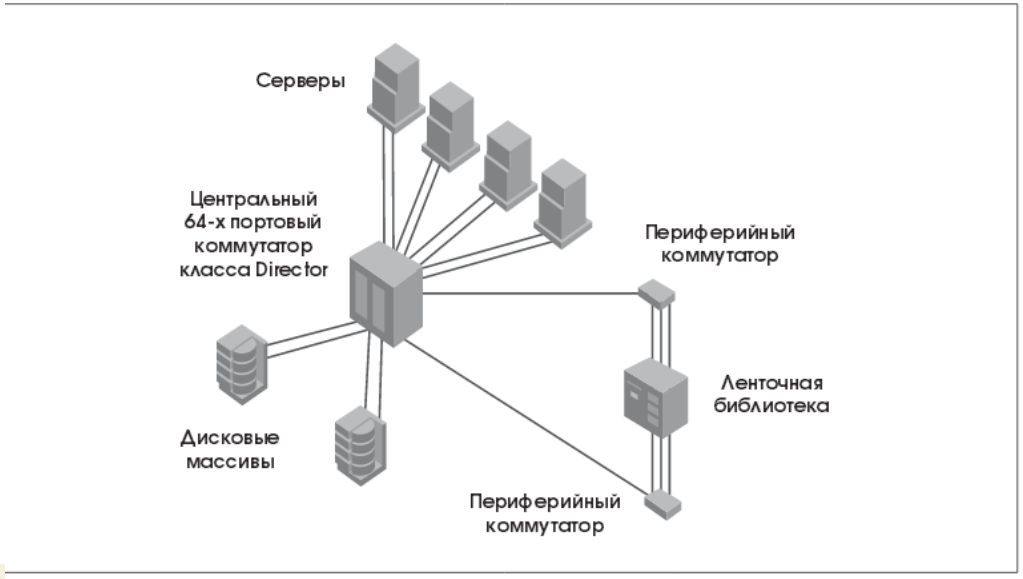
- Host Bus Adaptors (HBA);
- Ресурсы хранения данных;
- Устройства, реализующие инфраструктуру SAN;
- Программное обеспечение.

	SR-1294-MYFL	SR-1201-MYFF	SR-1202-FFX	FR8R/ FT8R
Производитель	Xyratex	Xyratex	Xyratex	Trimm
RAID контроллер	Mylex DAC960FL (до 64MB cache)	Mylex DACFF (до 256MB cache)	Mylex SANArray FFX (до 256MB cache)	Mylex SANArray FFX (до 1GB cache)
Интерфейс Внутренний/внешний	4 внешних FC порта 4 шины U2WSCSI	4 внешних FC порта 4 FC петли	2 внешних FC порта Двойная FC петля + 2 FC петли расширения	2 внешних FC порта Двойная FC петля + 2 FC петли расширения
Max. объем данных (при использовании 73GB HDD)	>2.4 TB (31HDD)	> 13 TB (96 HDD)	>7 TB (180 HDD)	>7 TB (180 HDD)
Производительность, IO op/s	~5000	~15000	~15000	~15000
Пропускная способность, MB/s	~200	~400	~200	~200

Архитектура SAN: примеры реализации





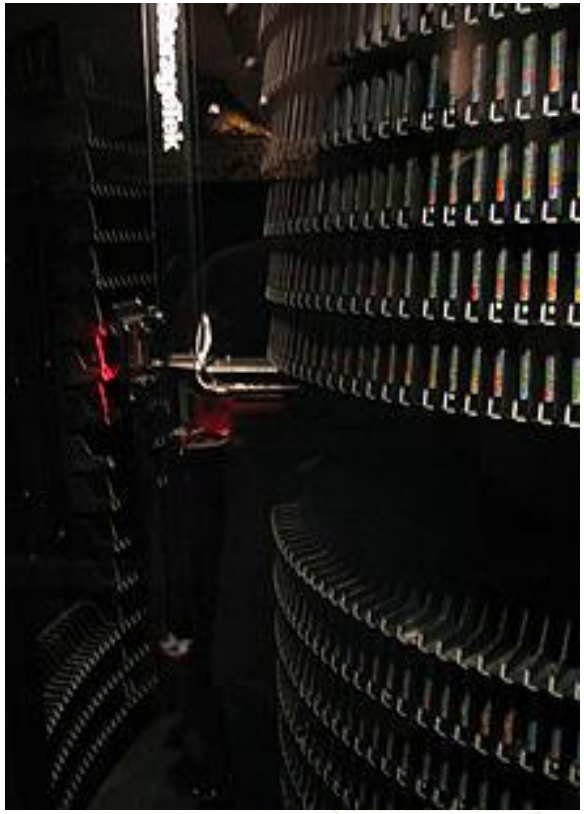


Библиотеки магнитных лент

- Стри́мер — запоминающее устройство на принципе магнитной записи на ленточном носителе, с последовательным доступом к данным.

Привод магнитных лент	Линейная плотность записи
DLT7000	86
Super DLT 1*	133
TR-5	106
Mammoth	78
DDS-3	122

Привод магнитных лент	Эффективность формата
DLT7000	74%
TR-5	76%
Mammoth	58%
DDS-3	59%



Библиотеки архивного хранения

Архивные накопители - класс роботизированных устройств хранения данных, использующих оптический способ записи на соответствующих носителях, включает магнито-оптические, CD-, DVD- и BD-библиотеки.

Преимущества:
- практически неограниченная ответственность;
- большая емкость;
- высока скорость доступа;
- простота интеграции;

Недостатки:
- высокая стоимость;
- дороговизна носителей;

	Hewlett-Packard SureStore 320ex Optical Jukebox	Globalstor GSL 52208
Объем хранимых данных	312,8 Гб	208 Гб
Количество дисководов	4	S
Емкость одного диска	До 5,2Гб	До 5,2Гб
Количество слотов для дисков	64	40
Средняя скорость передачи данных	4,6 Мб/с	3,37 Мб/с
Интерфейс	SCSI-2	SCSI-2

Спасибо за внимание

следующая лекция

Объектные базы данных

ЛЕКЦИЯ 15

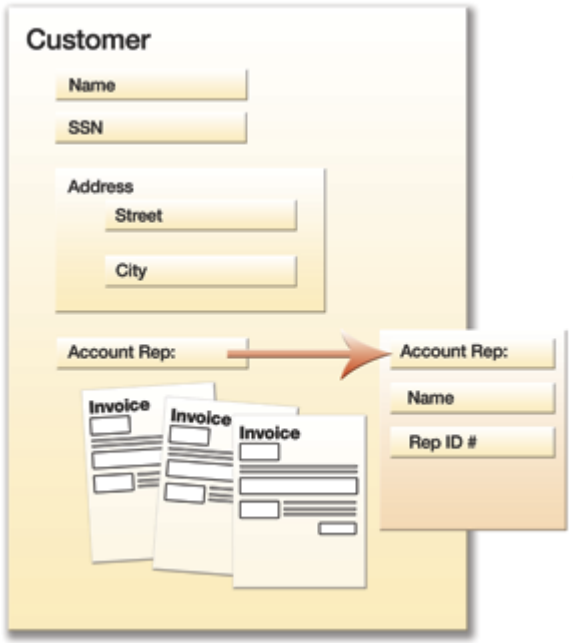
Объектные базы данных

Содержание лекции 15

- Объектные базы данных
- Объектная методология
- Соответствие реляционного и объектного представлений
- Стандарт OGMG
- Объектно-ориентированные СУБД
- Компоненты Cache
- Средства администрирования
- Примеры запросов к СУБД

Объектные базы данных

Объектно-ориентированная база данных — база данных, в которой данные моделируются в виде объектов, их атрибутов, методов и классов



В наиболее общей и классической постановке объектно-ориентированный подход базируется на концепциях:

- объекта и идентификатора объекта;
- атрибутов и методов;
- классов;
- иерархии и наследования классов.

Работа со сложными объектами в приложениях:

мультимедиа,	географических информационных системах,
GAD/GAM,	полнотекстовых базах данных,
CASE,	системах управления большими компьютерными сетями.

Объектная методология

- **Наследование** - это способность порождать один класс объектов из другого. Новый класс (подкласс) сохраняет все свойства и методы своего «родителя», кроме того, он может иметь дополнительные свойства и методы, характерные только для него.

- **Множественное наследование** подразумевает, что подкласс может иметь более одного «родителя».

- **Инкапсуляция** дает возможность трактовать объект в приложении как своеобразный «черный ящик». Независимо от уровня сложности, определенный класс того или иного объекта имеет определенное число общедоступных свойств и методов.

- **Полиморфизм** означает, что методы, принадлежащие к различным классам, могут использовать один и тот же интерфейс вне зависимости от конкретной реализации этих методов.

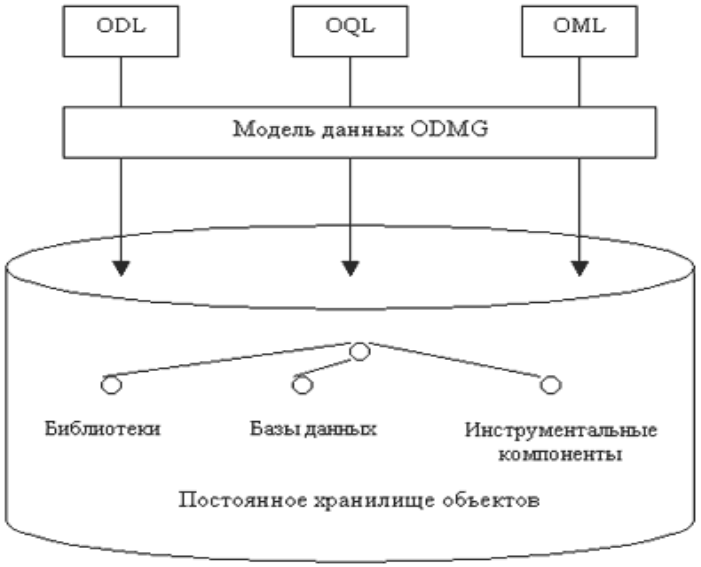
Соответствие реляционного и объектного представлений

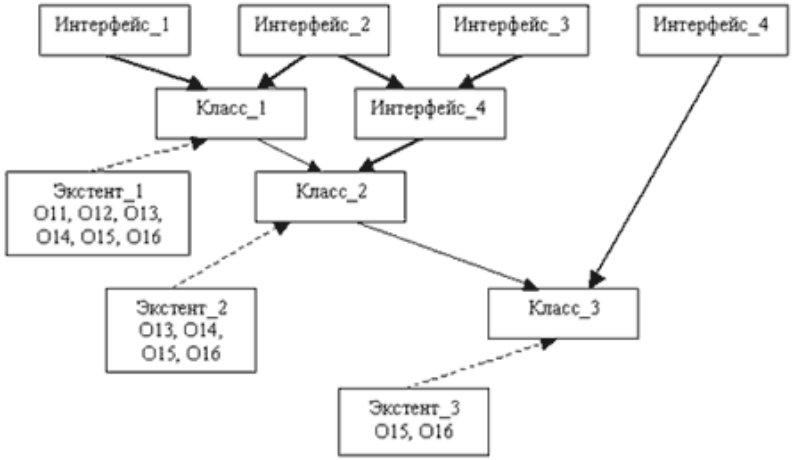
Объектное понятие	Реляционное понятие
Класс	Таблица
Экземпляр	Строка
Идентификатор объекта (OID)	ID-столбец в виде первичного ключа
Свойство-константа	Столбец
Ссылка на хранимый объект	Внешний ключ
Встраиваемые объекты	Индивидуальные столбцы
Коллекция-список	Столбец с полем-списком
Коллекция-массив	Подтаблица
Поток данных	BLOB
Индекс	Индекс
Запрос	Хранимая процедура или представление
Метод класса	Хранимая процедура

Стандарт OGMG

- **Архитектура**

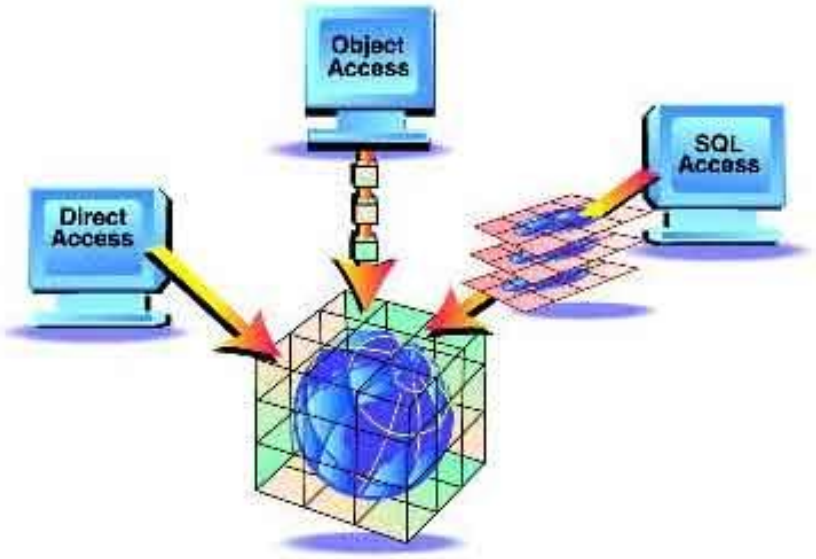
- Объектная модель данных.
- Язык определения данных
- Язык объектных запросов
- Языки манипулирования объектами
- Постоянное хранилище объектов.
- Инструментальные средства и библиотеки.

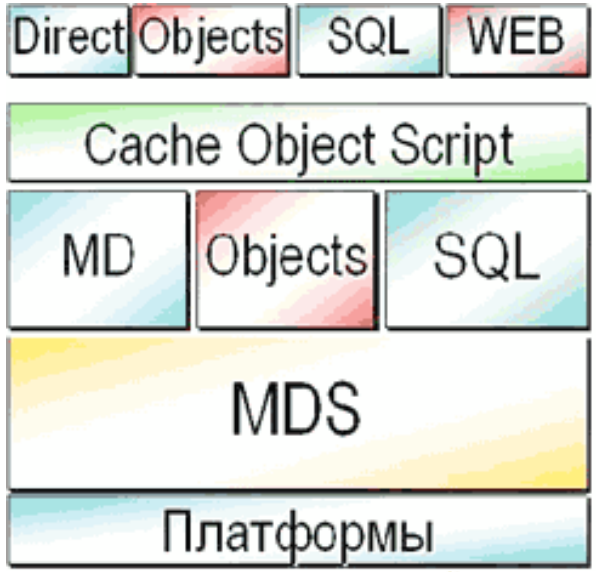




Объектно-ориентированные СУБД

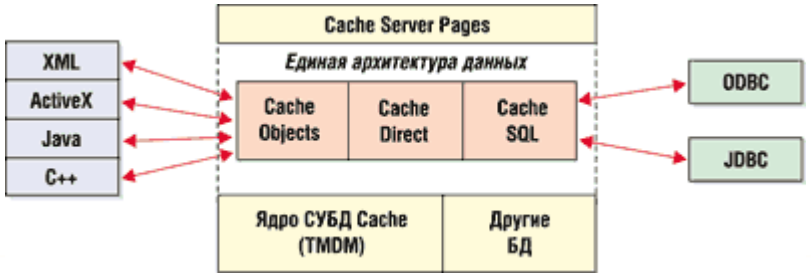
- **Encore** в Брауновском университете (Broun University);
- **Cactis** в Колорадском университете (University of Colorado at Boulder);
- **Thor** в Массачусетском технологическом институте (MIT);
- **Exodus** в Висконсинском университете (University of Wisconsin);
- **Pisa** в университетах Глазго и Св. Эндрю (Universities of Glasgo and St. Andrew).
- **Cache** высокопроизводительная промышленная СУБД, интегрированная с технологией разработки Web-приложений





Компоненты Cache

Описание объекта:
^car("Mercedes","SL600","colors")=3
^car("Mercedes","SL600" ,"colors",1)="black"
^car("Mercedes","SL600" ,"colors",2)="blue"
^car("Mercedes","SL600" ,"colors",3)="white"

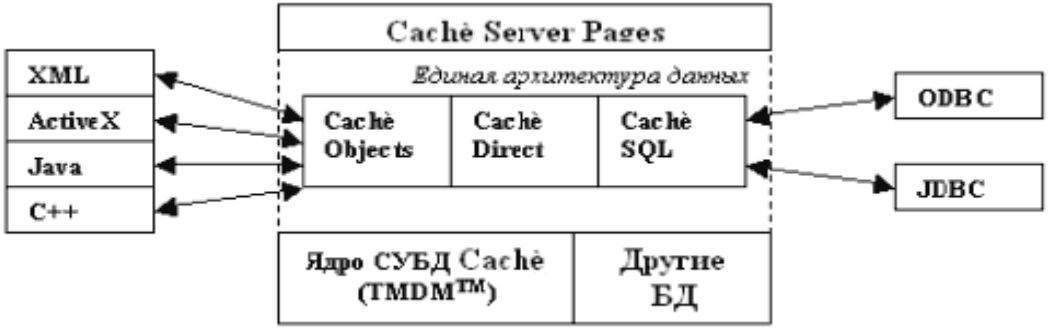


Доступ к данным в варианте SQL методологии:

new id, Surname set Surname="Ivanov"
&sql(SELECT Id into :id FROM Person where Surname=:Surname)

Компоненты ядра:

TMDM. Многомерное ядро системы.
Сервер Cache' Objects..
Сервер Cache' SQL..
Сервер прямого доступа.



Средства администрирования

- Создать новую БД, удалить или изменить настройки существующей БД
- Определить область (Namespace) для существующей БД.
- Определить CSP-приложение.
- Определить сетевое окружение Cache. В Cache реализован собственный протокол DCP (Distributed Cache Protocol) для работы с сетью распределенного окружения БД.
- Настроить систему Cache. Разработчику предоставляется возможность конфигурирования различных компонентов системы.

Пример запросов к ООБД

Объектный доступ к данным

Одна из важнейших особенностей Caché — возможность объектного доступа к данным за счет использования Caché Managed Provider

В результате в .NET-приложении используются абсолютно те же классы, что и были описаны в Caché. пример работы с объектами Caché в .NET-приложении выглядит так:

- CacheConnection CacheConnect = new
- <p>CacheConnection();
- <p>CacheConnect.ConnectionString = "Server =
- <p>localhost; "
- <p>+ "Port = 1972; " + "Namespace = SAMPLES; "
- <p>+ "Password = sys; " + "User ID = _system;"; <p>CacheConnect.Open();
- <p>Sample.Person person = Sample.Person.OpenId
- <p>(CacheConnect, "1");
- <p>Console.WriteLine("TinyProxy output: \r\n "
- <p>+ person.Id() + ": "+ person.Name);
- <p>person.Close();
- <p>CacheConnect.Close();

Спасибо за внимание

следующая лекция

Нереляционные базы данных

ЛЕКЦИЯ 16

Нереляционные базы данных

Содержание лекции 16

- Нереляционные базы данных.
- Псевдореляционные базы данных.
- Постреляционные СУБД.
- Малые СУБД, основанные на инвертированных списках.
- Иерархическая СУБД IBM IMS.
- Язык DL1.

Не реляционные базы данных

- NoSQL в информатике — термин, обозначающий ряд подходов, направленных на реализацию хранилищ баз данных, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL.

- Основано на решениях:
- Применение различных типов хранилищ.
- Возможность разработки базы данных без задания схемы.
- Использование многопроцессорности.
- Линейная масштабируемость.
- Инновационность: «не только SQL» открывает много возможностей для хранения и обработки данных.
- Независимость скорости доступа от объёма данных.

Типы хранилищ нереляционных БД

Хранилище «ключ-значение»

Примеры хранилищ — Berkeley DB, MemcacheDB, Redis, Riak, Amazon DynamoDB.

Хранилище семейств колонок.

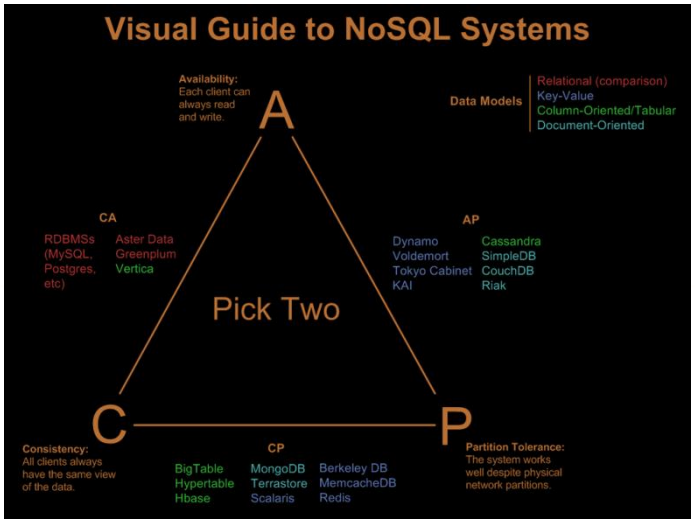
Примерами СУБД данного типа являются: Apache HBase, Apache Cassandra, Apache Accumulo, Hypertable.

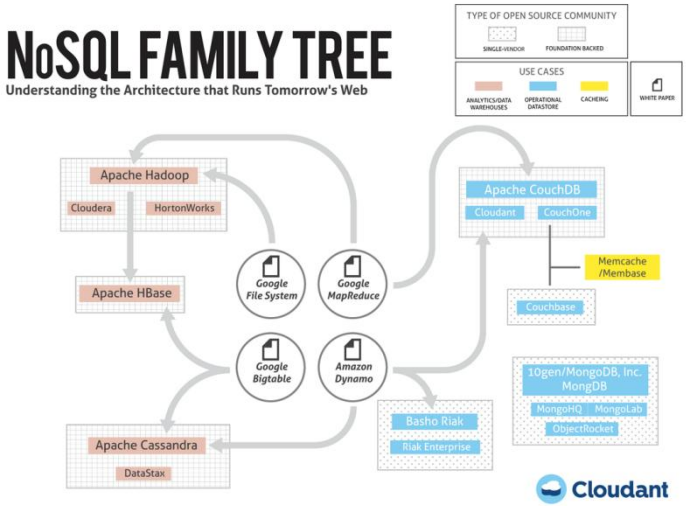
Документо-ориентированная СУБД

Примеры СУБД данного типа — CouchDB, Couchbase, MarkLogic, MongoDB, eXist, Berkeley DB XML.

Базы данных на основе графов

Примеры: Neo4j, AllegroGraph, Bigdata (RDF-хранилище), InfiniteGraph.





Постреляционные СУБД

Постреляционная модель данных представляет собой расширенную реляционную модель, в которой отменено требование атомарности атрибутов.

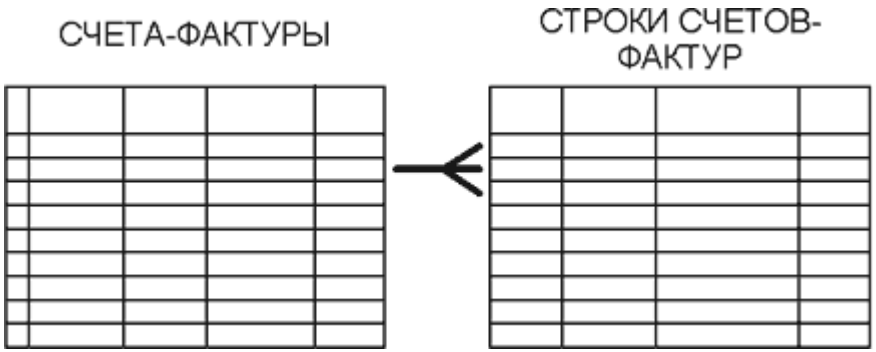
Расширение системы типов данных

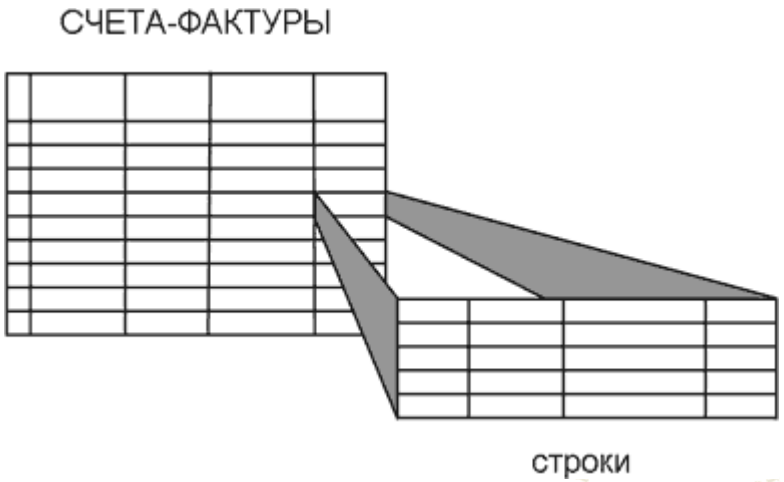
могут создаваться новые типы:

- описывается его интерфейс (функции, сигнатуры);
- реализация.

конструктор составных типов:

- структуры, множества, списки, массивы, объединения;
- ортогональность конструкторов (список массивов, объединения множеств);
- система образования пользовательских систем встроенных типов однообразны.





Формат хранения данных DBF

DBF — формат хранения данных, основанный на автономности таблиц.

Файл содержит типизированные данные.

Кортежи могут быть помечены как логически удалённые.

Физически удаление происходит после упаковки.

Файл состоит из:

- - заголовка;
- -блоков кортежей.

Является средством хранения и обмена данными в системах:

- -Dbase;
- - Clipper;
- - FoxPro for DOS or Windows;
- - Microsoft Excel.

```
// опишем базу данных
$dbname = "wplan.dbf";
$def =
    array(
        array(  "group",        "C",    10      ),
        array(  "dep_acr",      "C",    8       ),
        array(  "dep_code",     "C",    2       ),
        array(  "kaf_acr",      "C",    8       ),
        array(  "kaf_code",     "C",    2       ),
        array(  "discip",       "C",    100     ),
        array(  "formeduc",     "C",    15      ),
        array(  "orders",       "C",    15      ),
        array(  "spec",         "C",    6       ),
        array(  "standardid",   "C",    1       ),
        array(  "edep_acr",     "C",    8       ),
        array(  "edep_code",    "C",    2       ),
        array(  "ekaf_acr",     "C",    8       ),
        array(  "ekaf_code",    "C",    2       ),
        array(  "spring",       "N",    12,10   ),
        array(  "autumn",       "N",    12,10   ),
        array(  "account",      "C",    12      )
         );
```

Особенности использования в средах программирования

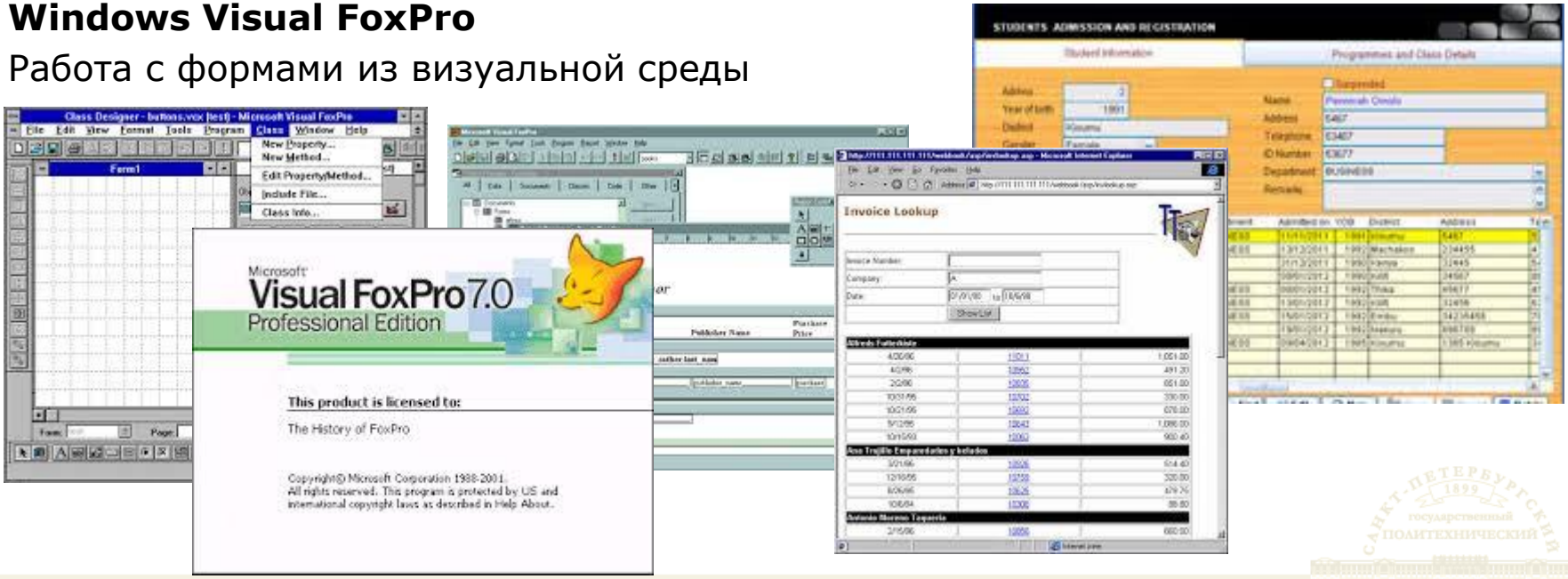
Clipper

Совместное использование базы данных и текстового полноэкранного интерфейса

```
USE Customer SHARED NEW
clear
@ 1, 0 SAY "CustNum" GET Customer->CustNum PICT "999999" VALID Customer->CustNum > 0
@ 3, 0 SAY "Contact" GET Customer->Contact VALID !empty(Customer->Contact)
@ 4, 0 SAY "Address" GET Customer->Address READ
```

Windows Visual FoxPro

Работа с формами из визуальной среды



Особенности использования в средах программирования

Языки программирования

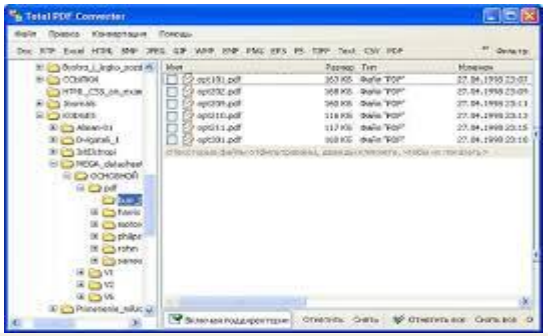
Экспорт и импорт данных в PHP

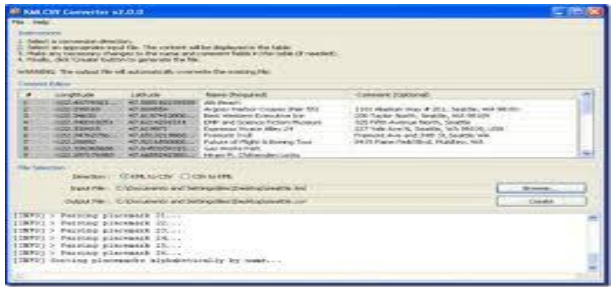
```
if (convert_cyr_string($record_array[5], 'a', 'w') != 'Военная подготовка')
   {
      dbase_add_record ($base_header, $record_array);
   }
```

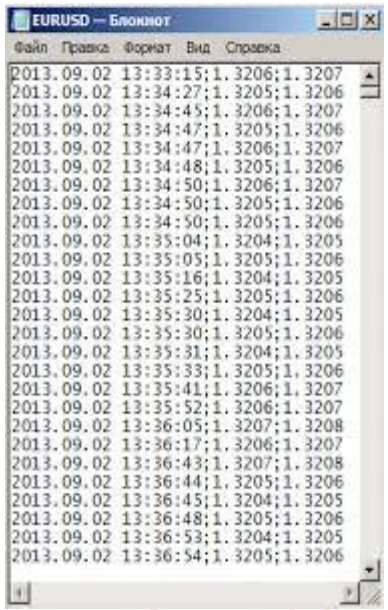
Использование в как средства обмена данными в Excel

Пример

```
rem конвертировать csv файл
csv_to_excel.exe test.csv test.xls
rem конвертировать все csv файлы в папке
FOR %%c in (oracle_3567_*.csv) DO csv_to_excel.exe %%c %%c.xls
```





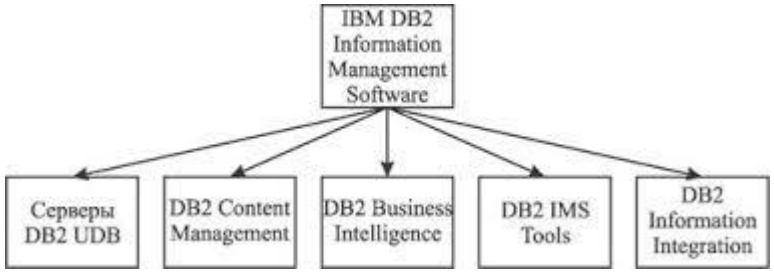


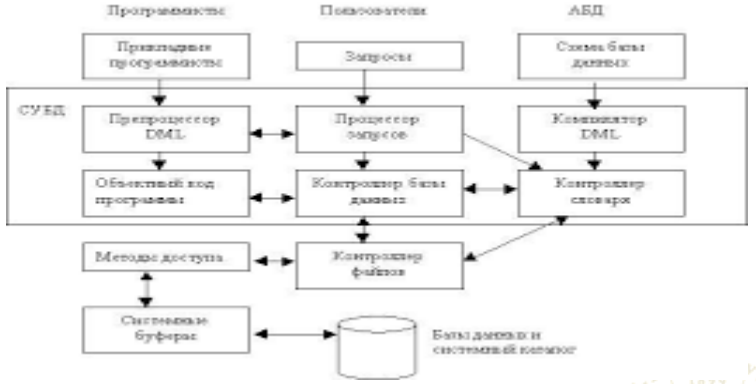
Иерархическая СУБД IMS

IMS реализует модель бизнеса «по требованию».

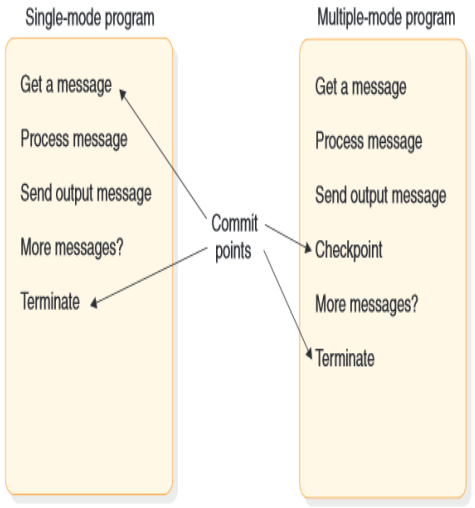
IMS Database Manager и IMS Transaction Manager (IMS TM) V10 обеспечивает:

- Измение способа ведения бизнеса благодаря интеграции информационных ресурсов.
- Создание приложений для бизнеса «по требованию».
- Создние масштабируемой, доступной, безопасной и легкоуправляемой среды.
- Применение существующих данных для принятия взвешенных решений.



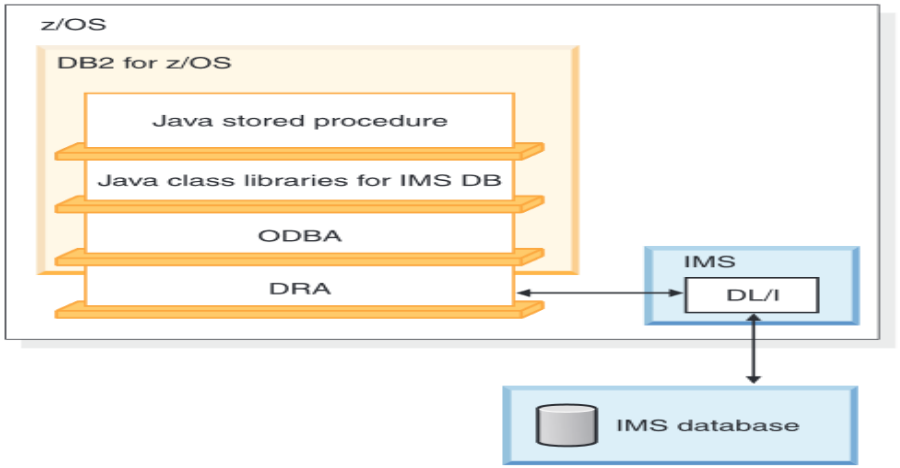


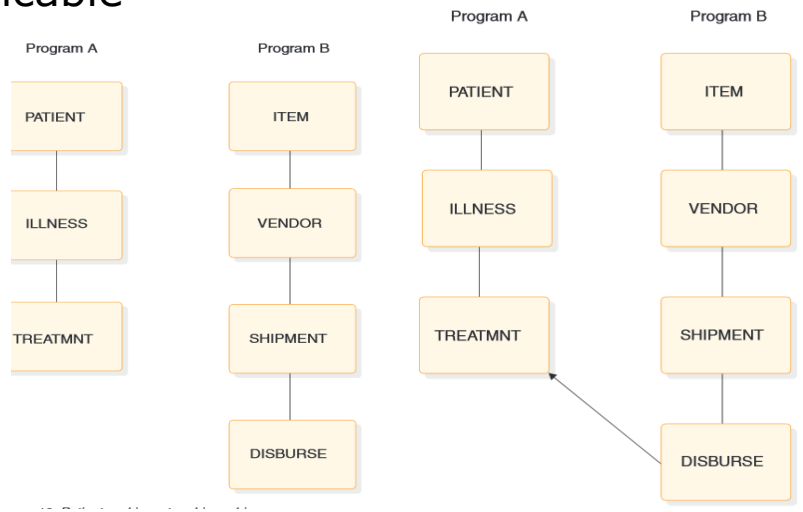
Язык DL1



DL/I calls for database management
Use these DL/I calls with IMS DB to perform database management functions in your application
Each call description contains:

- A syntax diagram
- Definitions for parameters that are available to the call
- Details on how to use the call in your application program
- Restrictions on call usage, where applicable





Спасибо за внимание

следующая лекция

Постреляционные базы данных

ЛЕКЦИЯ 17

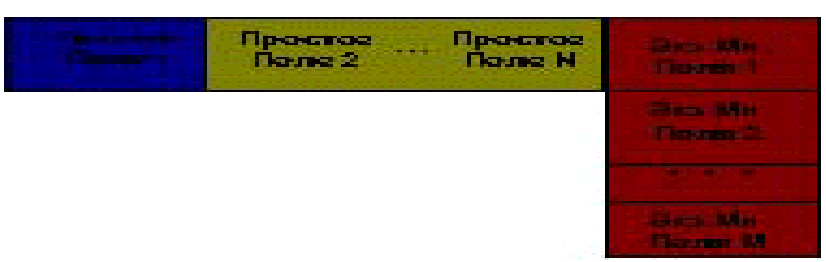
Постреляционные базы данных

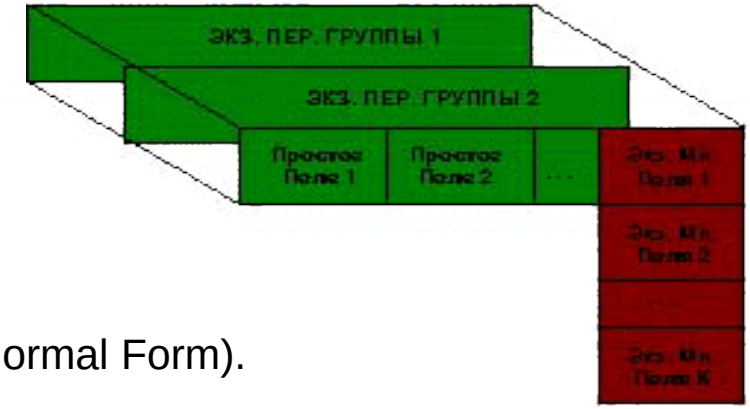
Содержание лекции 17

- Постреляционная СУБД ADABAS/NATURAL.
- Не первая нормальная форма.
- Основные принципы, лежащие в основе темпоральных баз данных.
- Понятие времени в темпоральных базах данных.
- Модели, используемые в темпоральных базах данных.

Постреляционная СУБД ADABAS

ADABAS - основа семейства программных продуктов фирмы Software AG для создания корпоративных баз данных.





Особенности хранения данных:

- Не первая нормальная форма (NF2 - Non-First Normal Form).
- Модель данных сущность-связь (E/R модель).
- Обработка и управление произвольными текстами.
- Обработка и управление географическими данными.

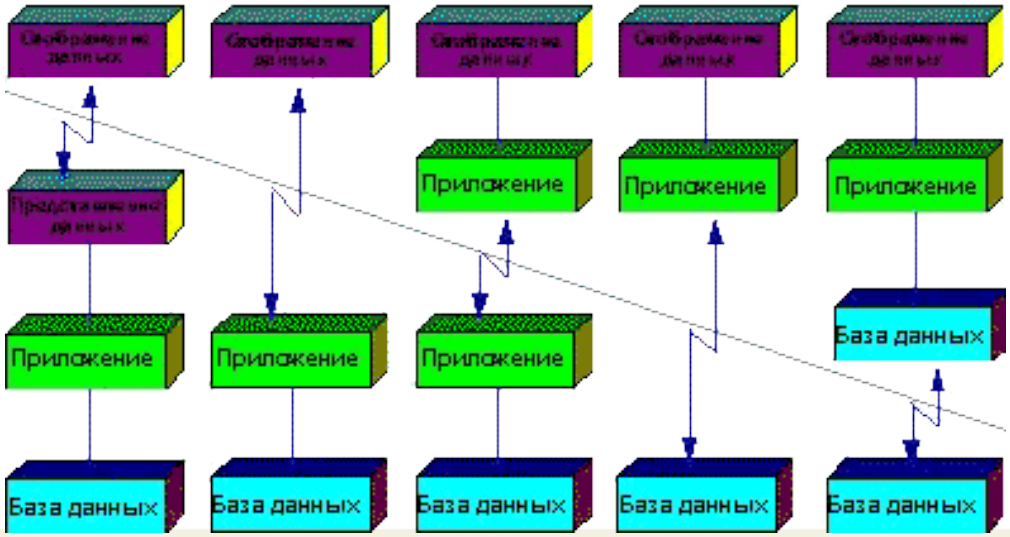
Особенности реализации ADABAS в корпоративных информационных системах

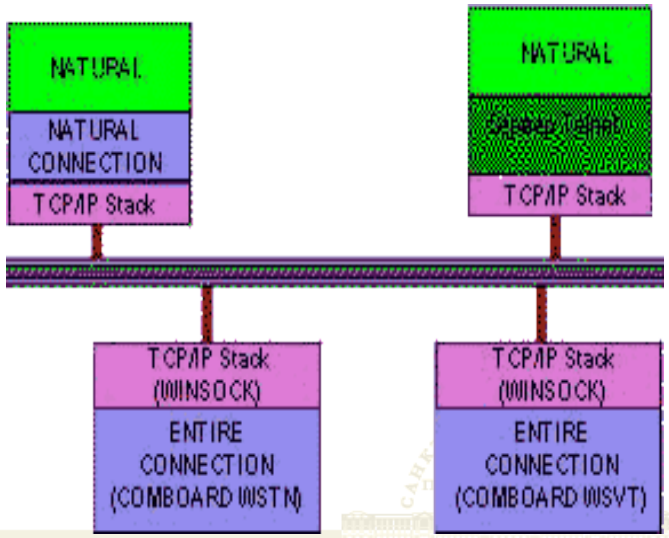
- распределенное отображение данных;
- удаленное отображение данных (эмуляция терминала);
- распределенное приложение (серверы приложений);
- доступ к удаленной базе данных (серверы баз данных);
- доступ к распределенной базе данных (интеграция/репликация баз данных).

Модель клиент-сервер ADABAS

Конфигурации программно-технических средств

•IBM mainframe,
•UNIX- суперсерверы;
•серверные ЭВМ (СЭВМ) среднего и малого классов;
•блоковые (типа IBM 3270) и символьные (типа VT 100 или ASCII) терминалы;
•высокопроизводительные ПЭВМ
•опорную сеть и протоколы (FDDI/Ethernet и/или TCP/IP, SPX/IPX и др.);
•программные продукты других фирм для доступа к базам данных, обработки и коммуникации, поддерживающие общепризнанные протоколы межпрограммного взаимодействия и связные протоколы (ORACLE, INFORMIX, SYBASE, NOVELL, IBM, MICROSOFT и др.).



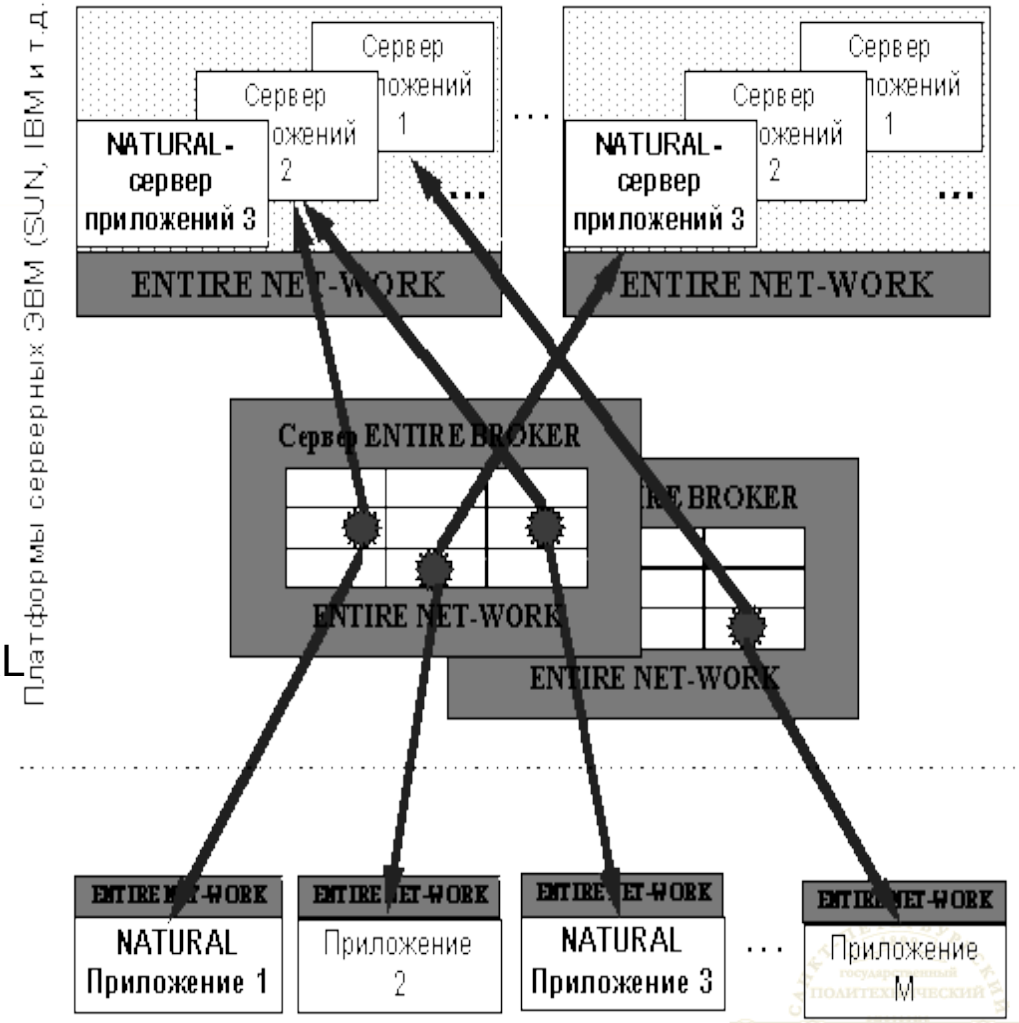


Инфраструктура распределённой среды ADABAS

ENTIRE BROKER выполнен в виде ядра -коммуникационного сервера, регистрирующего доступные в сети *серверы приложений и клиентов*, запрашивающих данные.

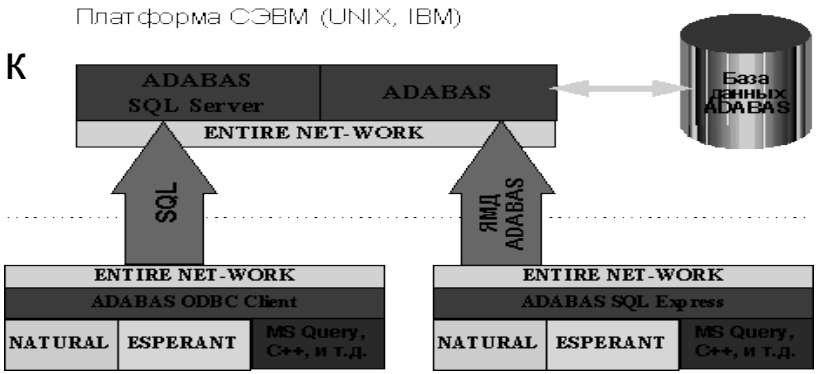
В качестве языков, на которых может быть разработано приложение использованы наиболее распространенные языки, поддерживающие CALL-интерфейс.

Сочетание ENTIRE BROKER, ENTIRE NET-WORK и NATURAL обеспечивает *прозрачность* для клиентов местонахождения сервера создавая администраторам прикладных систем условия для их правильного масштабирования.

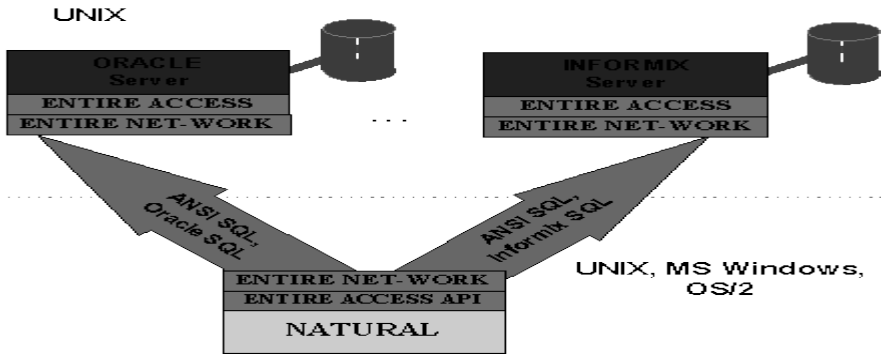


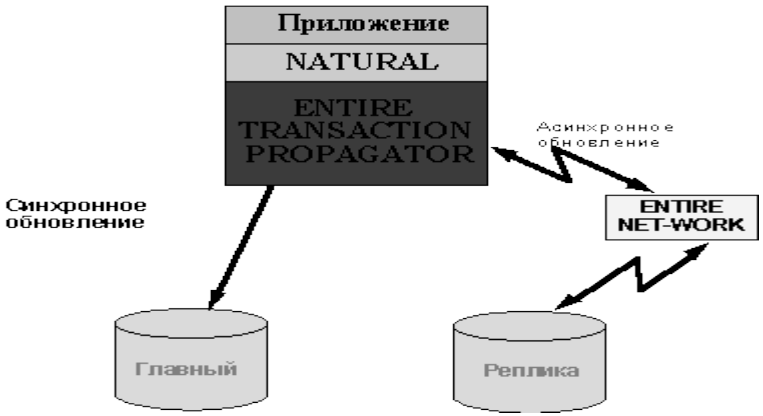
Доступ к удаленной базе данных

Стандартная конфигурация для доступа к серверам баз данных ADABAS со стороны клиентов на рабочих станциях под Windows для сетевого доступа.



Реализация доступа из прикладной системы с использованием языка манипулирования данными (ЯМД) СУБД ADABAS для локального доступа.





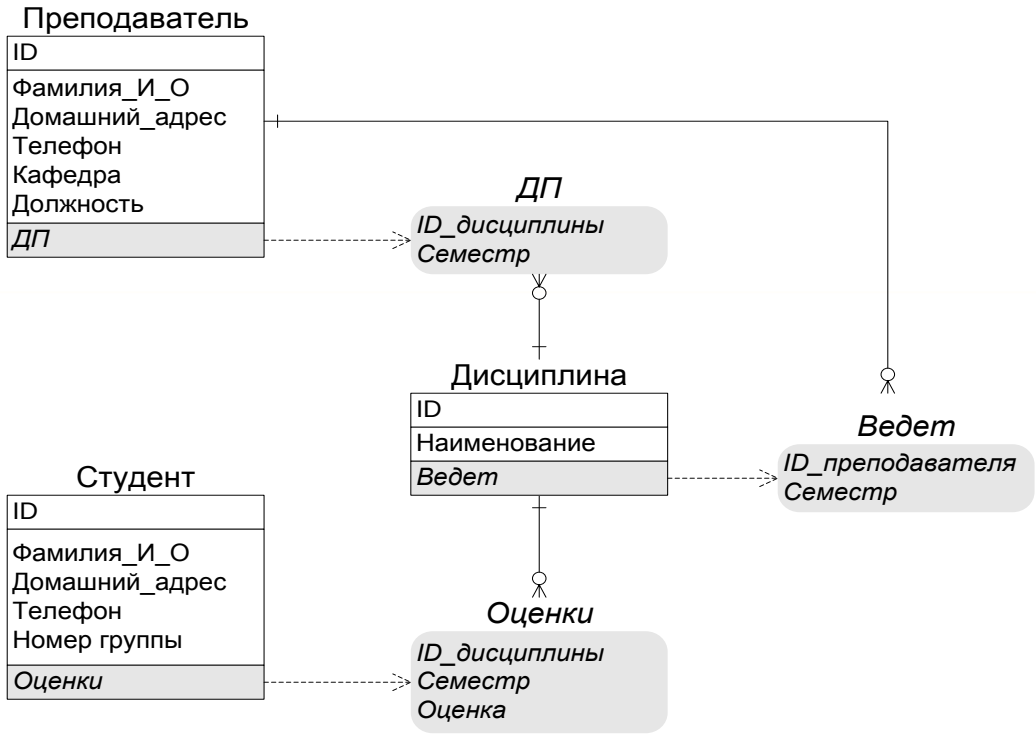
Доступ к распределенной базе данных для интеграции и репликации данных

Непервая нормальная форма

Ненормализованная форма

ПОДПИСЧИК	ИЗДАНИЕ
Иванов	Правда, Известия, Коммерсант
Петров	ДАН

ПОДПИСЧИК	ИЗДАНИЕ
Иванов	Правда
Иванов	Известия
Иванов	Коммерсант
Петров	



Достоинства:

- более естественное описание предметной области
- меньшие трудозатраты

Недостатки:

- повышенные требования к вычислительным ресурсам

Тенмпоральная база данных

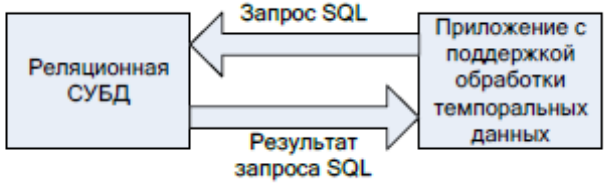
1. Информационная система обрабатывает темпоральные (изменяющиеся во времени) данные.
2. Информационная система накапливает историю изменения темпоральных данных.

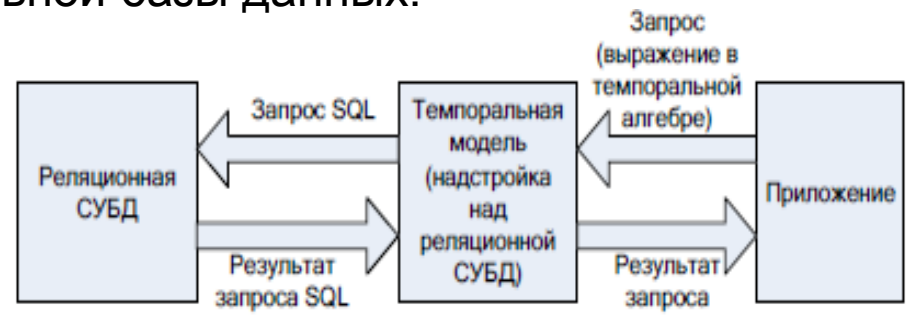
Условия хранения темпоральных данных:

$\forall t_1, t_2,\ t_1 < t_2,\ \exists t_3 : t_1 < t_3 < t_2.$

$\forall t_1, t_2,$ если $t_2 = \delta(t_1),$ то не $\exists t_3 : t_1 < t_3 < t_2.$

Технология создания темпоральной базы данных:

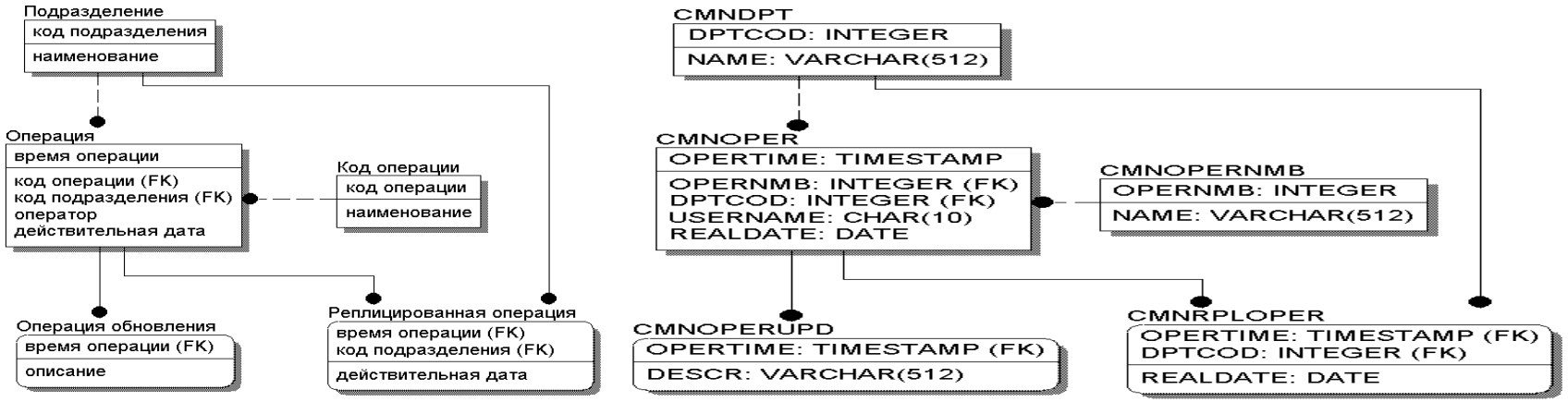




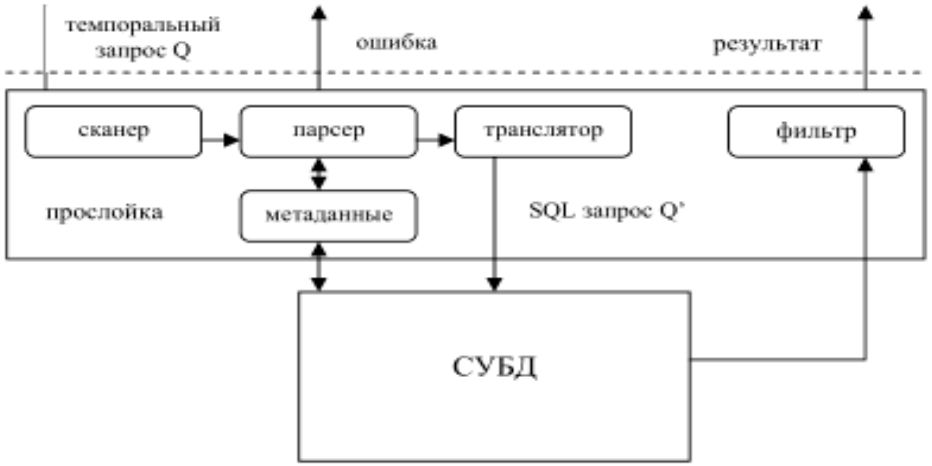
Цель функционирования темпоральной базы данных: определение времени фиксации определенного события или факта и времени выполнения какого-либо действия или операции.

Понятие времени в темпоральных базах данных

- Редко изменяемые таблицы:
  - Классификаторы.
  - Таблицы с нарастающим итогом.
- Изменяемые таблицы:
  - Терминальные таблицы. Это таблицы, на которые нет ссылок из других таблиц.
  - Нетерминальные таблицы. Это таблицы, на которые есть ссылки из других таблиц.



Модели темпоральных баз данных



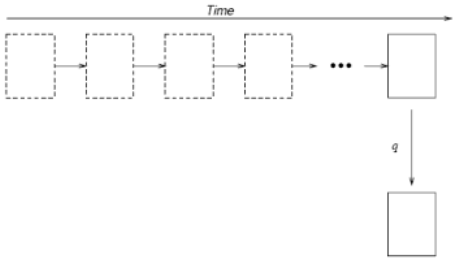
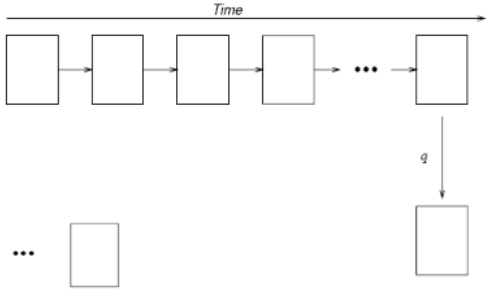


Рис. 8. Работа с таблицей без темпоральной поддержки



	действительное время		транзакционное время	
	ALAN	BOB	ALAN	BOB
19.12.2005				
20.12.2005			100	
...			100	
31.12.2005			100	
01.01.2006	100		100	
...	100		100	
28.02.2005	100		100	
01.03.2005	100	300	100	
02.03.2005	100	300	100	
03.03.2005	100	300	100	300
...	100	300	100	300
15.09.2005	**100**	**300**	**100**	**300**
...	100	300	100	300
27.01.2006	100	300	100	300
28.01.2006	100	300	100	500
...	100	300	100	500
31.01.2006	100	300	100	500
01.02.2006	100	400	100	500
...	100	400	100	500
05.02.2006	100	400	100	500
06.02.2006	100	400	100	400
...	100	400	100	400
сейчас	100	400	100	400

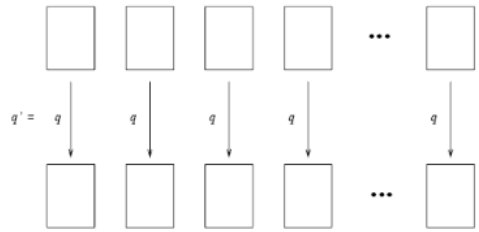


Рис. 10. Обработка "последовательных" запросов к темпоральной таблице

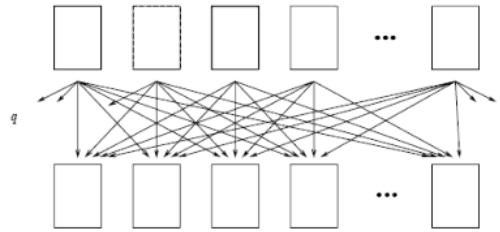


Рис. 11. Обработка "произвольных" запросов к темпоральной таблице

Спасибо за внимание

следующая лекция

Библиотечные базы данных

ЛЕКЦИЯ 18

Библиотечные базы данных

Содержание лекции 18

- Электронные библиотеки.
- Электронная библиотечная система.
- Электронное издание.
- Библиотечный каталог.
- Библиотечно–библиографические СУБД.
- Полнотекстовые СУБД.
- СУБД IBM STAIRS/BookManager.

Электронные библиотеки

- Электронная библиотека — упорядоченная коллекция разнородных электронных документов, снабженных средствами навигации и поиска.

- Форматы хранения: TXT; RTF и DOC; Mobipocket .PRC.

Направления: образоватеьные, научно-образовательные, научные.

Примеры:

Всемирная цифровая библиотека, Lib.ru, Альдебаран, Либрусек, Флибуста

Scopus, Web Of Science, РИНЦ.

Электроные библиотечные системы

- Электронно-библиотечная система — это предусмотренный федеральными государственными образовательными стандартами высшего профессионального образования России обязательный элемент библиотечно-информационного обеспечения учащихся вузов.
- Требования у ЭБС:

1 возможность индивидуального неограниченного доступа к содержимому электронно-библиотечной системы из любой точки, в которой имеется доступ к интернету;

2 возможность одновременного индивидуального доступа к содержимому электронно-библиотечной системы в соответствии с требованиями федеральных государственных образовательных стандартов высшего профессионального образования;

3 возможность полнотекстового поиска по содержимому электронно-библиотечной системы;

4 возможность формирования статистического отчёта по пользователям;

5 представление изданий с сохранением вида страниц (оригинальной вёрстки);

6 возможность доступа к зарубежным периодическим научным изданиям.

Электронные издания

Сетевые ЭИ:

К сетевым ЭИ могут быть отнесены:

- отдельные произведения (оригинальные и электронные представления печатных);
- сборники (в т.ч. материалы конференций);
- компьютерные средства обучения, БД;
- ГИС (географические информационные системы);
- сайты.

Сложно определяются БД (база данных), которые, не укладываются даже в самое широкое понимание книги, т.к. каждая БД – это отдельная информационная система, представляющая собой программный комплекс с набором данных/записей/документов.

Библиотечный каталог

- Впервые термин «библиотечный каталог» был закреплен в ГОСТ 7.76–96 «Комплектование фонда документов. Библиографирование. Каталогизация. Термины и определения». ЭК представляет собой «машиночитаемый библиотечный каталог, работающий в реальном времени и предоставленный в распоряжение читателя».

Электронный библиотечный каталог (ЭБК) — совокупность программных и аппаратных средств по обеспечению деятельности библиотеки по заказу, каталогизации, поиску, выдаче книг, решения различных задач по отчётности и книгообеспеченности читателей и др. как в локальной вычислительной сети, так и через web-сопряжение.

Функции каталога

- Администрирование;
- Комплектование + Заказ и регистрация периодики;
- Каталогизация + Картотека статей;
- Читательский поиск;
- Картотека читателей (с поддержкой штрих-кодирования);
- Книговыдача (с поддержкой штрих-кодирования);
- Электронная библиотека полнотекстовых ресурсов;
- Книгообеспеченность учебного процесса;
- Генератор штрих-кодов для печати (на А4);
- BookScan — создание электронных версий отсканированных книг;
- Web-модуль для удалённого доступа к электронной библиотеке.

Библиографические БД

- **документографическая (документальная) БД** (document database) — БД, содержащая библиографические записи и являющаяся информационной составляющей электронного каталога. Ее разновидностями являются: библиографическая БД (bibliographic database), которая содержит библиографические описания документов,
- **реферативная БД** (abstract database), которая содержит библиографические описания документов и рефераты;
- **полнотекстовая БД** (full-text database), в которой хранятся записи полнотекстовых документов или их частей.

Примеры:

Автоматизированная библиотечно-информационная система Руслан. (СПБГПУ)

ИНТЕГРИРОВАННАЯ БИБЛИОТЕЧНО-ИНФОРМАЦИОННАЯ СИСТЕМА ИРБИС (ГПНТБ России, Москва)

Состав библиотечной информационной системы

- Система ИРБИС ориентирована на работу в локальной вычислительной сети (ЛВС) и представляет собой совокупность взаимосвязанных автоматизированных рабочих мест (АРМ) пяти типов:
- **АРМ "КОМПЛЕКТАТОР"** - представляет собой рабочее место библиотечного работника, выполняющего функции по комплектованию и учету фондов библиотеки на основе ведения специальной базы данных (БД);
- **АРМ "КАТАЛОГИЗАТОР"** - представляет собой рабочее место библиотечного работника, выполняющего операции каталогизации и систематизации изданий, т.е. функции по формированию баз данных Электронного каталога;
- **АРМ "ЧИТАТЕЛЬ"** - представляет собой рабочее место конечного пользователя Электронного каталога и предназначен для всеобъемлющего поиска в Электронном каталоге, просмотра/печати найденной информации и формирования заказа на выдачу найденной литературы;
- **АРМ "КНИГОВЫДАЧА"** - представляет собой рабочее место библиотечного работника, выполняющего функции по выдаче литературы в соответствии с формируемыми заказами и ее возврату;
- **АРМ "АДМИНИСТРАТОР"** - представляет собой рабочее место специалиста, выполняющего системные операции над базами данных в целом, направленные на поддержание их в актуальном состоянии.
- Остановимся более подробно на каждом из АРМов.
- **АРМ "КОМПЛЕКТАТОР"** - обеспечивает решение следующих задач на основе ведения специальной БД.

Полнотекстовые СУБД

- Оперируют "документами": наборами атрибутов (ключ и соответствующее ему значение). Значения могут быть в свою очередь вложенными документами или массивами.
- Документы одного типа могут иметь общие атрибуты.
- А могут и не иметь - отсутствует жёстко заданная схема.

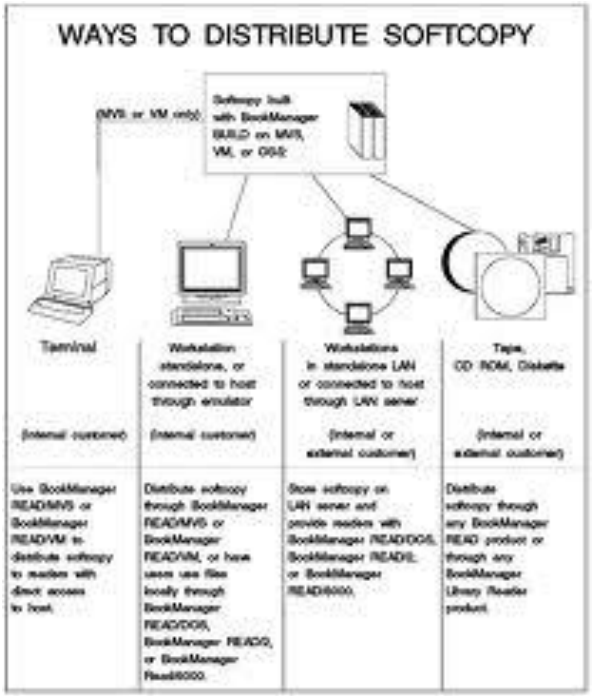

```
{
   "idleTimes": {
      "average": 31.3724,
      "values": null
   },
   "average": 145.442535,
   "throughput": [
      {
         "cars": 2401,
         "rate": 1200.5,
         "pos": "start"
      },
      {
         "cars": 2351,
         "rate": 1175.5,
         "pos": "end"
      }
   ],
   "averageSpeed": 11.530327,
   "stdeviation": 9.606599
}
```

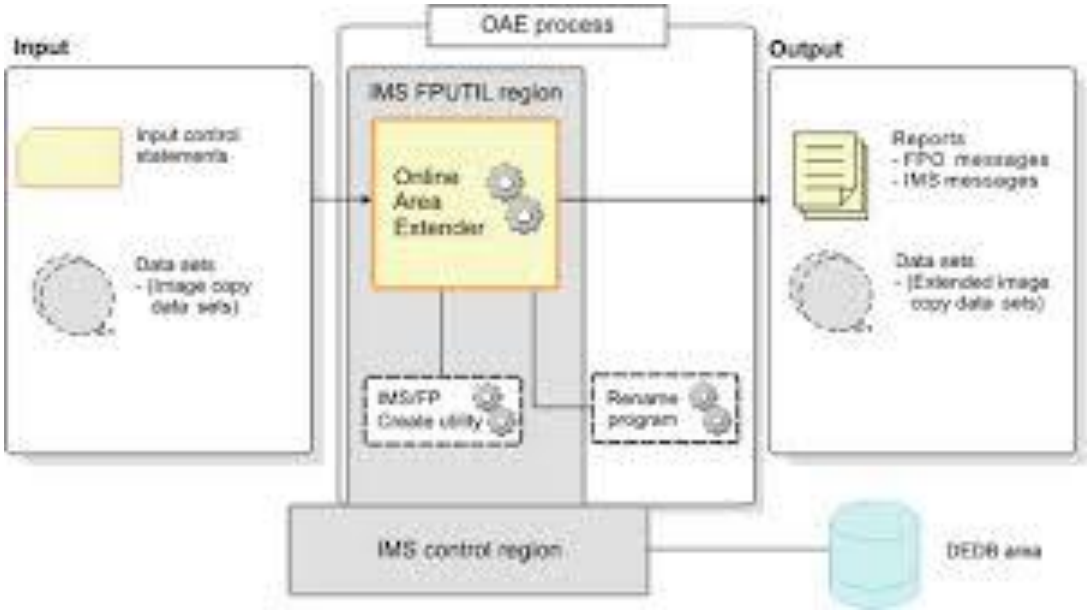

Полнотекстовые БД:

- удобны при хранении плохо структурированной информации
- масштабируются намного лучше RDBMS, и таким образом неизбежны при обработке очень крупных объёмов данных
- эффективны для аналитической обработки данных
- часто не имеют транзакционных механизмов.

СУБД IBM STAIRS/BookManager

- Http://www-03.ibm.com/systems/z/os/zos/library/bkserv/r13pdf/





Спасибо за внимание

Это была последняя лекция курса