**Содержание**

**1**. СУБД – системы управления базами данных и требования

к их функциональным возможностям стр. 2

**2**. Распределенные системы обработки данных стр. 3

*а*) СУБД структуры «сервер – клиент». Особенности построения

и работы стр. 3

б) СУБД компании Microsoft. Особенности построения и работы стр. 5

*в*) СУБД компании Oracle. Особенности построения и работы стр. 6

*г*) СУБД компании Informix. Особенности построения и работы стр. 8

д) СУБД компании Sybase. Особенности построения и работы стр. 10

**3**. Выводы стр. 12

**4**. Список используемой литературы стр. 14

**Системы управления базами данных и определенные требования к их функциональным возможностям**

Для взаимодействия пользователей с БД используются системы управления базами данных (СУБД), которые обеспечивают:

• набор средств для поддержки таблиц и соотношений между ними;

• развитый пользовательский интерфейс, позволяющий вводить и модифицировать информацию, проводить поиск и представлять результаты;

• средства программирования, позволяющие создавать собственные приложения.

Подход к построению СУБД значительно видоизменялся на протяжении почти 40 лет. На смену ВЦ предприятий и АСУП на их основе пришли персональные компьютеры и настольные (персональные) системы управления базами данных, затем с развитием коммуникаций появились распределенные системы и концепции управления крупными предприятиями- корпорациями на основе изучения бизнес-процессов. При этом в условиях динамичного их изменения в последние годы сформулирован *ряд определенных требований к функциональным возможностям СУБД*, производители которых стремятся поддерживать свои продукты на высоком, конкурентоспособном уровне.

1. Обеспечение непротиворечивого определения данных для всех используемых платформ, сетей и БД при сохранении распределенной архитектуры системы в целом, поскольку создание единой базы становится неприемлемым из-за ее громоздскости и высокой стоимости. Это означает, что создаваемые средствами СУБД приложения должны обладать высокой степенью мобильности и легко переноситься на разные компьютерные и сетевые платформы, когда на первый план выходят проблемы синхронизации и целостности. Кроме того, это позволяет в дальнейшем более свободно развивать ресурсы системы.
2. Коммуникационный обмен данными становится асинхронным, а информационные процессы – длительными, поэтому возникает необходимость журнализации состояния баз данных и проведения возможного отката/восстановления для расширения временных рамок (дни, недели).
3. Средства СУБД должны допускать возможность гибкого варьирования архитектурой различных ИС для соблюдения разумного компромисса при разделении функциональных возможностей системы между рабочими станциями клиентов и серверами. Современный уровень технологий (графический оконный интерфейс, мультимедиа-приложения) и возможность интеграции наследуемых частей систем приводят к целесообразности выполнения этих функций на достаточно развитых станциях клиентов. В то же время системы, обладающие «мощными» серверами (когда на последних осуществляется почти вся работа), демонстрируют масштабируемость, требуют меньших усилий для создания новых версий ИС при переконфигурировании данных, обладают повышенными показателями целостности и безопасности данных за счет размещения процедур на сервере. Кроме того, архитектуры, использующие размещение программ-приложений на сервере, благодаря их непосредственной близости к данным характеризуются менее напряженным сетевым трафиком.
4. Создание «менеджеров процессов» может быть эффективным только в таких условиях, когда средства программирования СУБД объектно-ориентированы и возможно создание стабильных приложений при динамичном изменении маршрутизации сквозь эти задачи. Критически важным становится подход многократного использования программного обеспечения, при котором обеспечивается требуемая бизнес-процессами гибкость и легкость создания новых версий ИС.
5. Производителям СУБД следует обеспечить соответствие поставляемых ими продуктов открытым стандартам взаимодействия, так как все чаще информационные системы призываются для реализации потоков запросов в архитектуре «клиент-сервер», охватывающих платформы различных поставщиков. При этом предусмотрительные владельцы ИС выиграют в наибольшей степени, когда будут использовать инструментарий разработки, охватывающий различные СУБД.
6. Расширение бизнес-процессов за пределы одной компании и необходимость создания глобальных информационных связей выдвигает серьезную задачу поддержки высокой степени готовности систем, работающих 24 часа в сутки все 365 дней в году. Это означает, что средства СУБД должны гарантировать администрирование баз данных в «горячем» режиме без остановки основных процессов.

Перечисленные требования к СУБД позволяют взглянуть на существующие в настоящее время на рынке продукты разных производителей под соответствующим углом зрения. Адекватность предлагаемых сегодня СУБД новым требованиям определит для их будущих владельцев и клиентов преимущества создаваемых ИС, их гибкость, мобильность, возможность легкой перестройки и, в конечном счете, способность к выживанию.

**Распределенные системы обработки данных**

***СУБД структуры «сервер-клиент»***

Построение распределенных систем важно, особенно в условиях бессистемного оснащения предприятий компьютерной техникой, когда многие уже работают (или им необходимо работать) с системами, состоящими из множества компьютеров разного типа (серверов, мини-компьютеров, больших машин).

Централизованное хранение данных и доступ к центральной БД в условиях географически распределенной системы требуют устройства соединений между центральным сервером, хранящим данные, и компьютерами-клиентами. Большинство таких компьютеров отделены от центрального сервера медленными и недостаточно надежными линиями связи, и работа в режиме удаленного клиента становится почти невозможной. Этим можно объяснить существующую ситуацию, когда в узлах распределенной системы функционируют группы автоматизированных рабочих мест (APM), абсолютно не связанные друг с другом.

Содержательная сторона задачи обычно требует обмена данными между группами АРМ, так как изменения в какие-либо данные могут вносить в одной группе, а использоваться – в другой. На практике обмен информацией реализуется регламентной передачей файлов – через модемное соединение или «с курьером».

То, что данные доставляются к месту назначения не системными средствами, а путем экспорта/импорта файлов, требует участия человека в процессе обмена, а это влечет за собой невысокую оперативность поступления данных и требует использования внешних механизмов контроля целостности и непротиворечивости. В результате возрастает вероятность появления ошибок. Кроме того, реализация всех алгоритмов обмена данными и контроля в этом случае возлагается на прикладных программистов, проектирующих APM. Объем работ по программированию и отладке подпрограмм обмена соответствует числу различных APM. Это также приводит к повышению вероятности сбоев в работе системы.

В современных технологиях APM объединены в локальную сеть. АРМ клиента выдает запросы на выборку и обновление данных, а СУБД исполняет их. Запросы клиента в соответствии с требованиями задачи сгруппированы в логические единицы работы (транзакции). Если все операции с базой данных, содержащиеся внутри транзакции, выполнены удачно, транзакция в целом также выполняется успешно (фиксируется). Если хотя бы одна из операций с БД внутри транзакции осуществлена неудачно, то все изменения в БД, происшедшие к этому моменту из-за транзакции, отменяются (происходит откат транзакции). Такое функционирование обеспечивает логическую целостность информации в базе данных.

При распределенной обработке изменения, проводимые приложением-клиентом, могут затрагивать более чем один сервер СУБД. Для поддержания целостности и в этом случае необходимо применение того или иного транзакционного механизма, реализуемого системными средствами, а не прикладной программой.

Но основной недостаток систем, построенных на распределенных транзакциях,- высокие требования к надежности и пропускной способности линий связи. Поэтому альтернативой распределенным транзакциям является репликация (дублирование) данных. В таких системах одна и та же информация хранится в различных узлах. Согласование значений и распространение данных по узлам осуществляются автоматически. В зависимости от условий, предусмотренных разработчиком, репликация может осуществляться либо сразу после наступления некоторого события, либо через заранее заданные интервалы времени, либо в определенный момент времени. Если узел, в котором выполняется репликация, в данный момент недоступен, информация об этом сохраняется в вызывающем узле и репликация осуществляется после восстановления связи. Более того, гарантируется сохранение заданного вызывающим узлом порядка выполнения репликации.

Современные СУБД используют так называемый *системный журнал*, в который вносят записи об изменениях в базе данных и завершении транзакций. Журнал используется сервером БД для отката или прокрутки транзакций после сбоев и для резервного копирования. Измененные данные из журнала передаются репликационному серверу, обслуживающему этот узел, который в соответствии с описанием тиражирования и подписками отправляет данные в эффективном специальном протоколе по месту назначения, т.е. к соответствующим репликационным серверам в удаленных узлах.

Именно в этом сечении – между репликационными серверами – связь может быть медленной или недостаточно надежной. Передаваемые данные в составе транзакций при недоступности узла-получателя записываются в стабильные очереди на диске и затем передаются по мере возможности.

В одной базе данных могут содержаться как первичные данные, так и данные-копии. Приложение-клиент, работающее со своей СУБД, может вносить изменения напрямую (операторами INSERT, DELETE, UPDATE) только в первичные данные. Для изменения копии данных предназначен механизм асинхронного вызова процедур. Для работы этого механизма в нескольких базах данных создаются процедуры с одинаковым именем и параметрами, но, разумеется, с различным текстом. В одной базе данных процедура помечается как предназначенная к репликации. Вызов этой процедуры вместе со значениями параметров через журнал и механизм репликации передается к узлам-подписчикам, и в их базах данных вызывается одноименная процедура с теми же значениями параметров.

Система, хранящая вторичные данные, может быть любой из ряда систем, доступных через шлюз: будь то Oracle, Informix, DB2, RMS или ISAM и т.д. Система, хранящая первичные данные, требует наличия менеджера журнала транзакций (Log Transfer Manager – LTM). Интерфейс LTM является открытым, и в скором будущем, возможно, подобные модули будут созданы для нестандартных источников данных.

Вообще тиражирование данных может найти самое разнообразное применение:

• для разгрузки сервера, выполняющего активное обновление данных от сложных запросов, связанных с поддержкой принятия решений;

• для консолидации данных от подразделений в центре;

• для обмена данными по медленным или ненадежным линиям связи;

• для поддержания резервной базы данных;

• для построения сети равноправных узлов, обменивающихся данными.

Подчеркнем, что репликационный сервер тиражирует транзакции, а не отдельные изменения в базе данных. Метод тиражирования транзакций гарантирует целостность внутри транзакций и, как следствие, невозможность нарушения ссылочной целостности. Схема обновления первичных данных и копий данных исключает возможность возникновения конфликтов; последние могут быть вызваны лишь неправильным проектированием системы или сбоем.

Большинство производителей современных промышленных СУБД в той или иной степени обеспечивают поддержку распределенной обработки транзакций. Распределенная обработка данных основывается на синхронных и асинхронных механизмах обработки распределенных транзакций. Эти механизмы могут использоваться и совместно. Поскольку каждый механизм обладает своими сильными и слабыми сторонами, его выбор зависит от требований конкретной подзадачи.

*Задача масштабирования* рано или поздно встает в любой организации, и это вполне объяснимо. Никто и никогда не покупает аппаратуру про запас, с большими резервами по вычислительной мощности. В то же время объемы хранимых данных и количество реально работающих приложений имеют тенденцию к неуклонному увеличению. Поэтому лучше всего изначально остановить свой выбор на такой аппаратной конфигурации, которая в дальнейшем будет легко наращиваться и развиваться.

Этому требованию в наибольшей степени отвечают компьютеру с симметричной многопроцессорной (SMP) или массивно-параллельной (MPP) архитектурой, на которых при увеличении количества процессоров обеспечивается практически линейный рост производительности. Обработка нескольких запросов, вложенных циклов внутри одного запроса, загрузка и сортировка данных, создание индексов и т.д. – все это выполняется параллельно на различных процессорах. Более того, одновременно реализуется эффективная динамическая балансировка загрузки системных ресурсов (процессоров, оперативной и дисковой памяти).

***СУБД компании Microsoft***

Microsoft SQL Server предоставляет широкие возможности: встроенную поддержку приложений Интернета, улучшенные механизмы распределенных транзакций, тиражирование в разнородных средах, расширенные средства распределенного управления и, что особо важно, новую архитектуру динамической блокировки.

В случае с динамической блокировкой, предложенной в SQL Server 6.5, используются интеллектуальные функции машины баз данных для оптимизации и детализации блокировок в зависимости от потребностей приложений, обращающихся к базе данных. Технология динамической блокировки рассматривается фирмой Microsoft в качестве стратегического направления в развитии SQL Server. Речь идет о реализации системы IRL (Insert row-level locking). В целом же стратегия динамической блокировки в SQL Server включает в себя:

• систему динамического выбора уровня блокировки (строка, страница, набор страниц или таблиц);

•блокировку на уровне строки для всех типов транзакций (INSERT, UPDATE, DELETE, SELECT);

•использование смешанных страниц, группирующих связанные строки из различных таблиц.

Microsoft, безусловно, стремится к тому, чтобы продемонстрировать преимущества архитектуры новой системы. Одним из наиболее частых аргументов оказывается именно наличие интеллектуальной системы динамической блокировки, построенной на уровне машины баз данных и избавляющей независимых разработчиков программного обеспечения от необходимости выбора эффективной системы блокировок. Это позволяет им полностью сосредоточиться собственно на разработке прикладного программного обеспечения. Упрощается и администрирование базы данных. Наконец, должен снизиться уровень непроизводительных затрат на блокировку.

В качестве модуля хранения и области, защищенной блокировкой, Microsoft SQL Server использует страницы. В большинстве случаев блокировка на уровне страницы оказывается приемлемой, хотя и вероятны случаи конкурентного доступа к страницам с часто используемой многими пользователями информацией.

Пакет «Microsoft SQL Server» ориентирован на эксплуатацию ресурсов сети Интернет и построение корпоративных систем класса Интранет. Выход пакета MS Server замыкает логическую цепочку, которую выстраивала компания Microsoft в сфере Интернета. Она включает в себя: сетевую основу распределенных вычислений (Windows NT Server); совокупность интегрированных приложений-серверов (BackOffice) технологииWin32 и OLE; средства разработки (Visual Basic, Java, Java Script, Front Page, Internet Studio), набор настольных приложений (Office) и системы, призванные поддерживать работу конечного пользователя,- Windows 95 и Windows NT Workstation.

***СУБД компании Oracle***

Архитектура, положенная в основу сервера Oracle, полностью реализует преимущества открытых систем и является дальнейшим развитием ставшей уже традиционной технологии «клиент-сервер». *Система управлениями базами данных Oracle RDBMS (сервер многомерной реляционной базы данных)* предоставляет пользователям широчайший набор возможностей для создания прикладных информационных систем:

• различного масштаба – от однопользовательских систем и систем уровня небольшого подразделения до глобальных межрегиональных распределенных систем обработки информации;

• различного назначения – от систем, ориентированных на работу в режиме реального времени (On-Une Transaction Processing – OLTP), до систем поддержки принятия решений (Decision Support Systems – DSS) и систем, основанных на концепциях «хранилищ данных» (Data Warehouse).

Oracle RDBMS соответствует самым строгим требованиям, предъявляемым к системам хранения и обработки информации:

• работает и обеспечивает масштабируемость и переносимость созданных на ее основе прикладных систем более чем на ста аппаратных и операционных платформах – от персональных ЭВМ до больших машин и многопроцессорных компьютеров с симметричной параллельной (SMP) и массивно-параллельной (MPP) архитектурами, а также поддерживает кластерную архитектуру;

• реально поддерживает многопроцессорные платформы (распараллеливание построения индексов, сортировки, запросов, копирования и восстановления данных). При этом с ростом количества процессоров производительность систем растет практически линейно;

• обладает исключительными возможностями по совместному использованию различных аппаратных, операционных, сетевых платформ и позволяет объединять все информационные системы предприятия в единую интегрированную вычислительную среду;

• позволяет организовать распределенное хранение и обработку данных (системы, основанные на двухфазной фиксации транзакций, на симметричной асинхронной и ассиметричной репликации данных);

• предлагает средства для бесперебойной промышленной эксплуатации (круглосуточной и круглогодичной) и поддерживает работу с очень большими объемами данных (до нескольких терабайтов);

• гарантирует автоматическую защиту от сбоев, случайных повреждений данных и несанкционированного доступа. После сбоев, не повлекших за собой физического повреждения внешних магнитных носителей, система восстанавливается автоматически во время повторного запуска ядра Oracle;

• поддерживает все основные промышленные стандарты;

• обеспечивает мощные декларативные ограничения целостности. Процедурные возможности позволяют работать с хранимыми процедурами, функциями, алертами, триггерами, пакетами процедур;

• включает в себя мощные утилиты для загрузки данных из внешних файлов в таблицы, их копирования, восстановления, перемещения из одной БД в другую, управления использованием дискового пространства;

• автоматически реализует коллективный доступ к данным. Имеет встроенный высокоэффективный механизм блокировок на уровне отдельной строки таблицы (как для собственно данных, так и для индексов);

• оптимизирует операции дискового ввода/вывода;

• проводит «интеллектуальную» оптимизацию запросов на основе анализа автоматически собираемой статистической информации о числе строк в таблице, наличии и селективности имевшихся индексов;

• поддерживает пользовательские расширения языка SQL ( разработчик имеет возможность писать на PL/SQL собственные функции: в дальнейшем на них можно ссылаться в выражениях SQL, как если бы они являлись встроенными функциями этого языка);

• поддерживает динамический SQL;

• позволяет ограничивать ресурсы, доступные пользователю (для каждого конкретного пользователя администратор может контролировать время доступа, операции ввода/вывода, потребление процессорного времени и т.д.);

• определяет привилегии и права доступа на основе механизма «ролей» (роли-поименованные группы привилегий; на их основе можно группировать различные привилегии доступа и затем назначать их конкретным пользователям/группам пользователей;

• полностью поддерживает другие языки, включая русский, сортирует слова по алфавиту, реализует национальные форматы представления данных (числовых, календарных, денежных), преобразует символьные данные из верхнего регистра в нижний и наоборот, выдает системные сообщения и подсказки на национальном языке. В случае неоднородной языковой среды (использования разных кодировок в различных узлах) преобразование данных выполняется автоматически.

В Oracle RDBMS поддерживаются два основных варианта репликации данных: асимметричная асинхронная и симметричная асинхронная.

Асимметричная асинхронная репликация, или, как ее еще называют, «*моментальная копия таблиц»* (Tables Snapshot), подразумевает выделение одного узла системы в качестве владельца основной «мастер-копии» некоторой таблицы. При этом автоматически может поддерживаться неограниченное число ее копий в других узлах системы, однако все они доступны только для чтения, а все изменения могут вноситься лишь в первичную «мастер-копию».

Высокая надежность является одним из самых общих требований, предъявляемых к любой реально работающей информационной системе. Заметим, что обеспечение повышенной надежности не является исключительной функцией только аппаратных или только программных средств – она достигается ими совместно. При этом Oracle RDBMS поддерживает практически весь спектр таких решений:

• использование реализуемых на аппаратном или системном уровне зеркальных дисков;

• создание действующей копии базы данных на резервном сервере;

• работу в архитектуре с параллельными серверами (многомашинные кластеры);

• применение в качестве серверов баз данных отказоустойчивых компьютеров.

Любая информационная система, реализованная средствами Oracle RDBMS, легко масштабируется, причем это может быть достигнуто несколькими различными способами:

• добавлением к системе новых серверов и переносом на них части приложений;

• заменой существующего сервера на более мощный, причем, так как Oracle RDBMS работает практически на всех существующих платформах, новый сервер может иметь принципиально другие архитектуру и операционную систему;

• переходом на кластерную многомашинную архитектуру или, если она уже используется, простым добавлением в кластер еще одного узла;

• увеличением числа процессоров при использовании компьютеров с SMP- или MPP-архитектурой.

Oracle RDBMS автоматически осуществляет декомпозицию простых и сложных SQL-запросов на параллельно выполняемые процедуры. Операции по распараллеливанию запросов являются прозрачными для квалифицированных разработчиков, а специальные опции и подсказки SQL допускают контроль над декомпозицией запроса.

***СУБД компании Informix***

СУБД компании Informix (*Informix-OnLine Dynamic Server)* – модель сервера СУБД, имеющая многопотоковую архитектуру и обладающая мощными средствами параллельной обработки запросов.

Informix базируется на UNIX, но поддерживает, помимо нее, операционные системы Windows NT и NetWare. СУБД компании Informix традиционно использовалась для создания информационных систем малого или среднего масштаба, работающих в режиме оперативной обработки транзакций. Компьютеры, работающие под управлением UNIX, обладают высокой производительностью, что потребовало пересмотра архитектуры сервера. Сервер баз данных Informix-OnLine Dynamic Server имеет многопоточную динамическую архитектуру (DSA), которая была разработана в содружестве с компанией Sequent. Эта архитектура призвана обеспечить максимальную поддержку систем SMP с масштабируемость в соответствии с числом процессоров и других ресурсов.

СУБД, претендующие на роль информационной основы современных предприятий, должны удовлетворять наиболее важным требованиям:

• иметь высокую производительность;

• обеспечивать смешанную загрузку сервера разными типами задач;

• обеспечивать непрерывную доступность данных;

• гарантировать целостность данных, в том числе распределенных;

• осуществлять распределенные вычисления;

• обеспечивать масштабируемость реализуемых на основе сервера ИС.

Архитектура сервера Informix-OnLine DS получила название «*динамическая масштабируемая архитектура»* (DSA). Суть ее заключается в том, что одновременно может выполняться относительно небольшое число серверных процессов, которые разделяют между собой работу по обслуживанию множества клиентов. По сравнению с более ранними моделями сервера Informix, где для каждого клиента создавался индивидуальный серверный процесс, новая модель обладает рядом преимуществ:

• снижает нагрузки на операционную систему;

• сокращает совокупную потребность клиентов в оперативной памяти;

• обеспечивает снижение конкуренции при одновременном использовании системных ресурсов;

• дает более рациональное по сравнению с ОС назначение приоритетов и планирование;

• обусловливает равномерную загрузку наличных процессоров;

• ускоряет обработку сложных запросов за счет параллельного выполнения на нескольких процессорах.

Архитектуру Informix-OnLine DS называют также *многопотоковой.* Для каждого клиента создается так называемый поток, или нить. Поток – это подзадача, выполняемая в рамках одного из серверных процессов. В некоторых случаях для обслуживания одного клиентского запроса формируется несколько параллельных потоков.

Важный оптимизирующий механизм сервера – кеши хранимых процедур и словарей данных. Словари данных, доступные только для чтения, а также хранимые процедурыраспределяются между всеми пользователями сервера, что позволяет оптимизировать совокупное использование памяти. При загрузке в разделяемую память словарь данных записывается в структуры, обеспечивающие быстрый доступ к информации, а хранимые процедуры преобразуются в выполняемый формат. Все это существенно ускоряет выполнение приложений, обращающихся ко многим таблицам с большим числом столбцов и/или ко многим хранимым процедурам. Операции ввода/вывода, как правило, образуют наиболее медленный компонент обработки баз данных, поэтому от их реализации существенно зависит общая продуктивность сервера.

Для оптимизации ввода/вывода и повышения надежности в сервере реализованы следующие механизмы:

• собственное управление дисковой памятью;

• асинхронный ввод/вывод;

• опережающее чтение.

Многопотоковая архитектура сервера послужила базой для реализации *технологии параллельной обработки запросов* (Parallel Data Query – PDQ), обеспечивающей эффективное выполнение сложных запросов, характерных для систем поддержки принятия решений.

PDQ – это технология, которая позволяет разделить обработку одного сложного запроса на ряд процессоров, мобилизовать для его выполнения максимально доступные системные ресурсы, во много раз сокращая сроки получения результата. Перечислим основные типы заданий, на которых проявляется эффект технологии PDO:

• обработка сложных запросов, включающих сканирование больших таблиц, сортировку, соединение, группирование, массовые вставки;

• построение индексов;

• сохранение и восстановление данных;

• загрузка, выгрузка данных, реорганизация баз данных;

• массовые операции вставки, удаления, модификации данных.

Технология PDO используется в основном для быстрого выполнения DSS – запросов и пакетных приложений. Если ее применение ничем не ограничено, то сильно распараллеленное выполнение нескольких сложных запросов приводит к недопустимому замедлению OLTP – приложений, выполняющихся на том же сервере. Управление степенью распараллеливания запросов и долей системных ресурсов, выделяемых для PDO-обработки, может осуществляться динамически. Например, в часы наиболее активной работы приложений OLTP запросы DSS выполняются без распараллеливания или с невысокой степенью распараллеливания. В остальное же время или на серверах, где приложения OLTP отсутствуют, устанавливается режим максимального использования PDQ.

Средства обеспечения надежности и инструменты администрирования Informix-Online DS позволяют свести к минимуму риск и последствия отказов, а также практически исключают плановые простои.

В Informix-OnLine DS реализована возможность дублирования связной дисковой области, выделенной под базу данных, на другой дисковой области. Основная цель этого – обеспечение высокой готовности, дополнительный эффект зеркалирования – оптимизация операций чтения. При сбое диска, на котором находится первичная область, администратор имеет возможность провести восстановление без перевода сервера в режим off-line. Все операции чтения записи продолжаются с зеркальной областью. Синхронизация данных на первичном диске после его включения осуществляется в оперативном режиме.

Тиражирование – это поддержание на другой вычислительной базе копий объектов базы данных. В Informix-OnLine DS реализовано прозрачное тиражирование данных с основного сервера баз данных на вторичный сервер, к которому разрешен доступ только на чтение и который может находиться в другом географическом месте.

Главная цель тиражирования в Informix-OnLine DS – это обеспечение высокой готовности. В случае отказа основного сервера вторичный сервер становится доступен и на запись. Прозрачное перераспределение клиентов при отказе основного сервера не поддерживается, но оно может быть реализовано в рамках приложений.

После восстановления основного сервера возможен сценарий, при котором восстановленный сервер становится вторичным. При этом бывшему вторичному серверу, который уже функционирует в режиме чтения-записи, придается статус основного. Подключенные к нему клиенты продолжают работу. Так обеспечивается непрерывная доступность данных.

При включении сервер всегда проверяет, не произошло ли последнее выключение системы аварийно. В этом случае множество транзакций, выполнявшихся в момент сбоя, остались в незавершенном, некорректном состоянии. Если сервер выявил такую ситуацию, то он запускает процедуру быстрого восстановления, которая обеспечивает возврат системы в корректное состояние.

Informix-OnLine DS позволяет создавать архивные копии данных, а в дальнейшем фиксировать те изменения, которые произошли на сервере с момента создания архива. Изменения сохраняются в файлах журнала транзакций. Важнейшее свойство инструментов архивирования и восстановления – возможность выполнять эти операции параллельно с доступом пользователей к серверу. Кроме того, предоставляются следующие возможности:

• автоматический запуск утилит архивирования по заданному расписанию;

• контроль меток архивных лент в целях исключения ошибок операторов;

• шифрование и сжатие архивируемых данных;

• контроль правильности считываемых данных путем вычисления контрольных сумм.

***СУБД компании Sybase***

Sybase System фирмы Sybase – современная высокопроизводительная СУБД, работающая на платформах РС-серверов для Novell NetWare, Windows NT, OS/2, SCO UNIX по технологии распределенной обработки и доставки данных.

В состав пакета Sybase System входят реляционные СУБД SQL Server и ряд других компонентов – репликационный сервер, отладчик, библиотеки серверной и клиентской части, средства управления и мониторинга и др.

Репликационный сервер использует асинхронную модель репликации транзакций. Правила репликации проектируются одновременно с разработкой модели данных. Затем проводится конфигурирование системы. При работе прикладной программы изменения данных отслеживаются системными средствами, и в соответствии с конфигурацией требуемая информация передается в удаленную СУБД. Репликационный сервер выполняет отдельную задачу, запускаемую одновременно с СУБД. Он имеет свой входной язык и стандартный для продуктов Sybase сетевой интерфейс Open Server. Такое разделение снижает нагрузку на СУБД и делает систему в целом более открытой.

Репликация данных в Sybase использует системный журнал как источник информации о завершенных транзакциях. В узле, содержащем первичные данные, для каждой тиражируемой базы данных запускается специальный модуль- менеджер журнала транзакций (Log Transfer Manager). Он подключается к серверу БД, получает от него уведомления о завершении транзакций.

Для анализа функционирования сервера на РС-клиент Sybase предоставляет модуль SQL Monitor, представляющий в графическом виде процессы загрузки, ввода/вывода и использования памяти сервером, а также характеризующий степень интенсивности транзакций. Для управления сервером применяется набор хранимых процедур и set-команд. Модуль Sybase SQL Server Manager выполняет те же функции, но имеет оконный интерфейс. Для учета информации о действиях пользователей служит модуль Sybase Audit Server, работающий со специальной базой данных, которая доступна для последующего анализа.

Зачастую в информационных системах возникают ситуации, когда требуется обеспечить доступ к источникам данных, которые сами по себе не являются СУБД. Предметом такой задачи может быть, например, обслуживание измерительной аппаратуры, ведущей непрерывный анализ течения какого-либо процесса. Другой вариант использования шлюзов - при необходимости подключать к ИС данные, находящиеся в СУБД третьих фирм (например, Oracle). Компонент OmniConnect, входящий в состав Sybase System, осуществляет унифицированный доступ приложений к разнородным источникам данных. Специальные шлюзовые компоненты организуют работу в системе с любой промышленной СУБД, включая Oracle и Informix. Приложения-клиенты при этом взаимодействуют только с сервером OmniConnect на диалекте SQL фирмыSybase, а необходимая трансляция и преобразование типов данных автоматически осуществляется шлюзовыми модулями.

OmniConnect хранит информацию о размещении таблиц на том или ином сервере БД. Централизованно хранятся и исполняются глобальные хранимые процедуры. Приложение-клиент может осуществлять транзакции, в которых участвуют таблицы из различных БД, а также выполнять процедуры, которые OmniConnect при работе с СУБД, отличными от Sybase, прозрачно преобразует к соответствующему диалекту SQL.

Технология OpenServer, входящая в состав Sybase System, на основе имеющихся библиотек позволяет разрабатывать собственные приложения, использующие поступающие данные от промышленного оборудования. Для приложения-клиента такие программы «выглядят», как хранимые процедуры на Sybase-совместимом сервере базы данных. На базе технологии OpenServer реализуется специальный драйвер, обеспечивающий доступ к ODBC-источникам данных. Приложение может подключаться к этому драйверу как к обычному серверу Sybase и выполнять на нем операции TransactSQLSybase. Драйвер самостоятельно обеспечивает приведение команд TransactSQL к стандарту ODBC. Использование подобных драйверов позволяет включать в систему не только промышленные СУБД других производителей, но и настольные СУБД. Другим примером применения технологии OpenServer может служить реализация доступа к электронной почте.

**Выводы**

**1.** Основой функционирования информационной системы является ее информационное обеспечение, в частности, базы данных (БД) и системы управления базами данных (СУБД). Большинство БД использует табличный способ представления, где данныерасполагаются по строкам (которые называются записями) и столбцами (которые называются полями), однако не любая таблица представляет собой БД, поскольку БД должны содержать только независимую (первичную) информацию. В последнее время наибольшее распространение получили реляционные базы данных.

**2.** Для взаимодействия пользователя с БД используются СУБД. Важный этап в их развитии – СУБД dBASЕ ׀׀׀ и dBASЕ ׀׀׀ PLUS фирмы Ashton Tate, которые, по существу, стали стандартом для программных продуктов данного класса. Современные настольные СУБД входят в состав интегрированных программных продуктов типа Office: Access – в MS Office, Paradox – в Corel Office, Approach – в Lotus SmartSuite.

**3.** С развитием Интернет- и Интернет - технологий появилась возможность доступа к удаленным БД, что потребовало создания стандартного языка (SQL – структурированный язык запросов), который мог бы использоваться так, чтобы функционировать в большом количестве различных видов компьютерных сред.

**4.** В условиях крупного офиса или предприятия в целом важно построение распределенных систем. Централизованное хранение данных и доступ к центральной БД в условиях географически распределенной системы приводят к необходимости установления соединений между центральным сервером, хранящим данные, и компьютерами-клиентами (APM). В современной технологии APM объединены в локальную сеть. APM-клиент выдает запросы на выборку и обновление данных, а СУБД исполняет их. Запросы клиента в соответствии с требованиями задачи сгруппированы в логические единицы работы (транзакции). Если все операции с базой данных, содержащиеся внутри транзакции, выполнены удачно, транзакция в целом также выполняется успешно (фиксируется). Большинство производителей современных промышленных СУБД в той или иной мере обеспечивают поддержку распределенной обработки транзакций. Альтернативой распределенным транзакциям считается репликация (дублирование) данных. В таких системах одна и та же информация хранится в различных узлах. Согласование значений и распространение данных по узлам осуществляется автоматически. В зависимости от условий, специфицированных разработчиком, репликация может осуществляться либо сразу после наступления некоторого события (скажем, модификации строки таблицы), либо через заранее заданные интервалы времени (каждую минуту, каждый час и т.д.), либо в определенный момент времени (например, ночью, когда загрузка и стоимость линий связи минимальны).

**5.** Из современных СУБД, служащих для построения распределенных систем, выделим Microsoft SQL Server, Oracle RDBMS, Informix-On Line Dynamic Server, Sybase System. Microsoft SQL Server предоставляет широкие возможности: встроенную поддержку приложений сети Internet, улучшенные механизмы распределенных транзакций, тиражирование в разнородных средах, расширенные средства распределенного управления и, что особо отмечается, новую архитектуру динамической блокировки. Пакет «Microsoft SQL Server» ориентирован на эксплуатацию ресурсов сети Интернет и построение корпоративных систем класса Интранет. Архитектура, положенная в основу сервера Oracle, полностью реализует преимущества открытых систем и является дальнейшим развитием ставшей уже традиционной технологии «клиент-сервер». Oracle RDBMS предоставляет широчайший набор возможностей для создания прикладных информационных систем. В Oracle RDBMS поддерживаются два основных варианта репликации данных: ассиметричная асинхронная репликация и симметричная асинхронная репликация. СУБД компании Informix – модель сервера СУБД, имеющая многопотоковую архитектуру и обладающая мощными средствами параллельной обработки запросов. СУБД компании Informix традиционно использовалась для создания информационных систем малого и среднего масштаба, работающих в режиме оперативной обработки транзакций. Sybase System – современная высокопроизводительная СУБД, работающая на платформах РС-серверов для Novell NetWare, Windows NT, OS/2, SCO UNIX по технологии распределенной обработки и доставки данных. В состав пакета Sybase System входят реляционная СУБД SQL Server и ряд других компонентов – репликационный сервер, отладчик, библиотеки серверной и клиентской части, средства управления и мониторинга и проч. Репликационный сервер использует асинхронную модель репликации транзакций.

**6.** При построении ИС сразу появляется необходимость в едином информационном пространстве, обработка данных в котором осуществляется в режиме реального времени. Но следует помнить, что на предприятии, как и в реальной жизни, все объекты имеют между собой как прямые, так и более сложные связи, которые к тому же находятся в постоянном развитии. Эти связи и процессы должны каким-то образом отражаться в базе данных, под которой подразумевается не просто статичное хранилище, а информационная модель части реального мира. Это под силу только профессиональным (полномасштабным) СУБД, например фирм Oracle, Informix, Sybase и т.д.

**Список используемой литературы**

1. Банк, Зверев В.С., Информационные системы в экономике: учебник. -М.: Экономистъ, 2006,-447с.
2. Емельянова Н.З., Партыка Т.Л., Попов И.И., Информационные системы в экономике: Учеб. пособие.- М.: ФОРУМ: ИНФРА – М, 2006. – 464 с. : - (Профессиональное образование).
3. Исаев Г.Н., Информационные системы в экономике: [учебное пособие]/Г.Н. Исаев. – М.: Омега – Л, 2006. – 462 с. – (Высшее экономическое образование).