

Основные характеристики СУБД Oracle

1. [Ключевые возможности Oracle Database](#)
2. [Создание базы данных \(файлы параметров\)](#)
3. [Необходимые предпосылки](#)
4. [Создание базы данных ORACLE](#)
5. [Управляющие файлы](#)
6. [Оперативные журналы](#)
7. [Табличные пространства и файлы данных](#)
8. [Сегменты, экстенты и блоки](#)
9. [Экземпляр Oracle](#)
10. [Структуры памяти Oracle](#)
11. [Процессы Oracle](#)

Oracle Database - это объектно-реляционная система поддерживающая некоторые технологии, реализующие объектно-ориентированный подход, то есть обеспечивающих управление создания и использования баз данных.

Ключевые возможности Oracle Database

- Real Application Cluster (RAC) обеспечивает работу одного экземпляра базы данных на нескольких узлах grid, позволяя управлять нагрузкой и гибко масштабировать систему в случае необходимости.
- Automatic Storage Management (ASM) позволяет автоматически распределять данные между имеющимися ресурсами систем хранения данных, что повышает отказоустойчивость системы и снижает общую стоимость владения (TCO).
- Производительность. Oracle Database позволяет автоматически управлять уровнями сервиса и тиражировать эталонные конфигурации в рамках всей сети.
- Простые средства разработки. Новый инструмент разработки приложений HTML DB позволит простым пользователям создавать эффективные приложения для работы с базами данных в короткие сроки.
- Самоуправление. Специальные механизмы Oracle Database позволяют самостоятельно перераспределять нагрузку на систему, оптимизировать и корректировать SQL-запросы, выявлять и прогнозировать ошибки.
- Большие базы данных. Теперь максимальный размер экземпляра базы данных Oracle может достигать 8 экзабайт.
- Недорогие серверные системы. Oracle Database может использовать недорогие однопроцессорные компьютеры или модульные системы из “серверов-лезвий”.
- В новой версии базы данных реализована поддержка переносимых табличных пространств, система управления потоками данных Oracle Streams и модель распределенных SQL-запросов. Для переноса существующих баз данных в среду Grid в них не потребуется вносить изменений, что позволяет быстро начать использовать все преимущества Oracle Database.

Создание базы данных (файлы параметров)

При создании базы данных необходимо подготовить несколько файлов данных операционной системы, которые будут использоваться вместе как единая база данных. База данных создается один раз, независимо от того, сколько файлов данных она имеет, и сколько экземпляров будут обращаться к ней. Процедуру создания базы данных можно также использовать для того, чтобы стереть информацию в существующей базе данных и создать новую базу данных с тем же именем и физической структурой.

Создание базы данных включает следующие операции:

- ✦ создание новых файлов данных, или стирание данных, хранившихся в предыдущих файлах данных
- ✦ создание структур, требующихся ORACLE для доступа и работы с базой данных (словаря данных)
- ✦ создание и инициализация управляющих файлов и файлов журнала повторения для базы данных

База данных создается с помощью предложения, включающего команду SQL CREATE DATABASE; однако, прежде чем выдавать такое предложение, рассмотрите следующие вопросы:

- ✦ Спланируйте ваши таблицы и индексы, и оцените, сколько пространства они потребуют.
- ✦ Спланируйте проблемы защиты вашей базы данных, включая конфигурацию ее онлайн-журналов и архивных журналов (с учетом занимаемого ими пространства) и стратегию резервного копирования.
- ✦ Выберите набор символов базы данных. Вы должны указать набор символов при создании базы данных, и не сможете изменить его до повторного создания базы данных. Все символьные данные в базе данных, включая данные в словаре данных, хранятся в наборе символов базы данных. Если пользователи предполагают обращаться к базе данных, используя другие наборы символов, то набор символов базы данных должен быть надмножеством всех используемых наборов символов.

Необходимые предпосылки

Для создания новой базы данных вы должны иметь:

- ✦ привилегии операционной системы, соответствующие полноправному администратору базы данных
- ✦ достаточное количество памяти для запуска экземпляра ORACLE
- ✦ достаточный объем дискового пространства на компьютере, выполняющем ORACLE

Создание базы данных ORACLE

Для создания каждой новой базы данных ORACLE необходимо последовательно выполнить следующие шаги:

1. Скопировать все существующие базы данных.
2. Создать файлы параметров.
3. Отредактировать новые файлы параметров.
4. Проверить идентификатор экземпляра ORACLE.
5. Запустить SQL*DBA и соединиться с ORACLE как INTERNAL.
6. Запустить экземпляр.
7. Создать базу данных.
8. Скопировать базу данных.

База данных Oracle содержит следующие виды файлов:

- **Управляющие файлы (Control files)** - содержат метаданные о самой базе данных. Эти файлы очень важны для базы данных. Без них не могут быть открыты файлы данных и поэтому не может быть открыт доступ к информации базы данных.
- **Файлы данных (Data files)** - содержат информацию базы данных.
- **Оперативные журналы** (оперативные файлы повторного выполнения) - позволяют восстановить базу данных после сбоя экземпляра. Когда работа базы данных завершается аварийно и при этом не теряются никакие файлы данных, экземпляр может восстановить базу данных на основе информации в этих файлах.

Существуют другие файлы, которые формально не входят в базу данных, но важны для успешной работы БД.

- **Файл параметров** - используется для описания стартовой конфигурации экземпляра.
- **Файл паролей** - позволяет пользователям удаленно подсоединиться к базе данных для выполнения административных задач.
- **Архивные журналы** (архивные файлы повторного выполнения) - содержат историю созданных экземпляром оперативных журнальных файлов (их автономные копии). Эти файлы позволяют восстановить базу данных. Используя их и резервы базы данных, можно восстановить потерянный файл данных.

Остановимся на каждом типе файлов подробнее

Управляющие файлы

Управляющий файл читается при старте экземпляра в ходе монтирования базы данных. В записях этого файла хранятся описания физических файлов, образующие базу данных. Когда файлы добавляются к базе данных, в управляющий файл автоматически вносятся изменения.

Местоположение управляющих файлов задается в параметре инициализации. Для защиты от отказов базы данных, вызванных потерей управляющего файла, следует мультиплексировать управляющий файл и использовать для них, по

крайней мере, три различные физические устройства. Сервер базы данных Oracle сопровождает копии управляющего файла, заданные в параметре инициализации.

Оперативные журналы

В оперативные журнальные файлы повторного выполнения (redo log files) пишутся записи об изменениях, выполняемых в базе данных транзакциями и внутренними операциями сервера Oracle. Они позволяют восстановить целостность базы данных после системных сбоев, вызванных прекращением подачи электроэнергии, дисковых сбоев и т.п. Оперативные журнальные файлы необходимо мультиплексировать, чтобы гарантировать сохранность хранимой в них информации в случае дисковых сбоев.

Журнальные файлы входят в группы журналов. Группа содержит журнальный файл и его мультиплексируемые копии. Каждая такая копия - член журнальной группы, и каждая группа однозначно определяется ее номером. Процесс записи данных повторного выполнения (log writer - LGWR) пишет информацию из журнального буфера в журнальную группу. После заполнения файлов журнальной группы или выполнения операции перехода из одной группы в другую процесс LGWR начинает писать в следующую группу. Журнальные группы используются "по кругу".

Табличные пространства и файлы данных

База данных разделена на логические структурные единицы, называемые табличными пространствами. Они используются для объединения хранимых в них логически связанных структур. Каждая база данных содержит одно или несколько табличных пространств. Для хранения информации, содержащейся в логических структурах табличного пространства, создается один или несколько файлов данных.

Сегменты, экстенды и блоки

Такие объекты БД, как таблицы и индексы, хранятся в табличных пространствах в виде сегментов. Каждый сегмент состоит из одного или более экстендов. Экстенд состоит из смежных блоков данных. Поэтому каждый экстенд может находиться только в одном файле данных. Блоки данных - наименьшие единицы ввода/вывода в базе данных.

Когда база данных запрашивает у операционной системы набор блоков данных, ОС отображает их в свои реальные блоки на устройстве хранения. Пользователю не надо знать физический адрес информации в базе данных. Файл данных может быть также расщеплен и храниться на нескольких дисках с применением зеркалирования.

Размер блока данных устанавливается в момент создания БД. Стандартный размер 8К подходит для многих баз данных. Если БД используется для хранилища данных с большими таблицами и индексами, тогда использование блоков большего размера может дать выигрыш в производительности. Если БД используется для транзакционного приложения, в котором чтения и записи производятся в случайном порядке, тогда лучше задать меньший размера блока.

Максимальный размер блока зависит от ОС. Минимальный размер - 2К (почти во всех случаях его не рекомендуется использовать).

Экземпляр Oracle

Сервер Oracle включает базу данных Oracle и экземпляр. Экземпляр состоит из буферов памяти, образующих системную глобальную область (System Global Area - SGA), и фоновых процессов, которые контролируют и выполняют большую часть невидимой работы при выполнении экземпляра.

Экземпляр простаивает (idle) до момента его старта. При запуске читается файл параметров инициализации и на его основе конфигурируется экземпляр.

Пользователи могут соединяться с базой данных после того, как экземпляр запустится и база данных будет открыта.

Структуры памяти Oracle

Основные структуры памяти, связанные с экземпляром Oracle:

- Системная глобальная область (System Global Area- SGA), разделяемая всеми серверными и фоновыми процессами
- Программная глобальная область (Program Global Area - PGA), частная для каждого серверного и фонового процесса; для каждого процесса выделяется одна PGA.
- Системная глобальная область (SGA) - это область разделяемой памяти, в которой содержатся данные и управляющая информация экземпляра.

SGA содержит следующие структуры данных:

- **Кэш буферов БД (Database buffer cache)** - для блоков данных выбираемых из БД.
- **Журнальный буфер (Redo log buffer)** - для кэширования информации повторного выполнения (используемой при восстановлении экземпляра) до момента их записи в журнальные файлы.
- **Разделяемый пул (Shared pool)** - для кэширования различных структур, которые могут совместно использоваться пользователями.
- **Большой пул (Large pool)** - необязательная область, в которой отводится память для буферов, требуемых большими операциями ввода/вывода.
- **Java-пул**, используемый для Java-кода сеансов и данных внутри виртуальной Java-машины (Java Virtual Machine - JVM).

При запуске экземпляра с помощью Enterprise Manager или SQL*Plus выводится информация о памяти, выделенной для SGA. В рамках динамической инфраструктуры SGA можно без остановки экземпляра менять размеры кэша буферов БД, разделяемого пула, большого пула, Java-пула и пула потоков.

Преконфигурированная база данных уже настроена и использует подходящие параметры распределение оперативной памяти. Однако по мере роста базы данных может возникнуть необходимость внести изменения в эти параметры.

Oracle позволяет выдавать сигнальные сообщения (alerts) для своевременного определения проблем, связанных с размером структур памяти, и содержит советчики (advisors), которые помогают установить подходящие значения для параметров.

Программная глобальная область (PGA) - это область памяти, выделяемая для каждого серверного процесса, содержащая данные и управляющую информацию этого процесса. Серверный процесс - это процесс, который обрабатывает запросы клиента. Каждый серверный процесс имеет свою приватную область PGA, которая создается при старте серверного процесса. Доступ к этой области имеет только этот серверный процесс, чтения и запись в эту область выполняются через код Oracle, вызываемый из этого серверного процесса.

Совокупный размер памяти, выделяемый под области PGA и их содержимое, зависит от того, сконфигурирован ли в экземпляре режим разделяемого сервера. Обычно PGA содержит:

- Приватную область SQL (Private SQL area), в которой находятся информация привязки и структуры памяти, создаваемые при выполнении команд. Каждый сеанс, в котором выполняется команда SQL, имеет приватную область SQL.
- Память сеанса (Session memory), выделяемая для обработки переменных сеанса и другой связанной с сеансом информацией.

Процессы Oracle

Когда вызывается прикладная программа или инструментальное средство, например, Enterprise Manager, Oracle создает серверный процесс для выполнения команд, порождаемых приложением.

Кроме того, Oracle создает набор фоновых процессов для экземпляра. Эти процессы взаимодействуют друг с другом и с операционной системой. Они управляют структурами памяти, записывают информацию на диск в асинхронном режиме ввода/вывода и выполняют общесистемные служебные действия.

Состав работающих в текущий момент фоновых процессов зависит от используемых функциональных возможностей базы данных. Наиболее общие процессы следующие:

- Системный монитор (System monitor - SMON); выполняет восстановление после отказа экземпляра при старте экземпляра.
- Монитор процессов (Process monitor - PMON); выполняет очистку после аварийного завершения пользовательского процесса.

- Процесс записи в БД (Database writer - DBWn); пишет модифицированные блоки из кэша буферов БД в файлы на диск.
- Процесс контрольной точки (Checkpoint - CKPT); сигнализирует DBWn о контрольной точке и изменяет все файлы данных и управляющие файлы, внося в них информацию о самой последней контрольной точке.
- Процесс записи в журнал (Log writer - LGWR); пишет журнальные записи на диск.
- Архиватор (Archiver - ARCn); копирует файлы оперативного журнала в архив после заполнения оперативных журнальных файлов или после выполнения переключения журнала.

Словарь данных

Словарь данных - централизованный набор таблиц и представлений, используемых в режиме "только чтение" для получения данных о БД. В словаре хранится, например:

- логическая и физическая структура БД;
- информация о пользователях БД;
- ограничения целостности;
- данные о выделенном для объектов схем пространстве и сколько из этого пространства используется.

Словарь создается, когда создается база данных, и автоматически изменяется при изменении структур базы данных. Enterprise Manager получает информацию об объектах БД из словаря. Вы можете только выполнять запросы информации из таблиц словаря данных. Enterprise Manager делает тоже самое для вас и представляет информацию в удобном для использования виде. Представление DICTIONARY содержит описание таблиц и представлений словаря данных. В именах представлений обычно имеется один из трех префиксов:

- USER- информация, относящаяся к объектам, принадлежащим пользователю.
- ALL - информация, относящаяся к объектам, доступным пользователю.
- DBA - информация о всех объектах базы данных.

Системные требования

Oracle Database Standard Edition One

Процессор:

- минимум - 1 Гц
- рекомендуется 2.5 Гц или выше

Операционная система:

Базовой операционной системой, необходимой для Oracle BI Standard Edition One, является Win32.

ОБРАТИТЕ ВНИМАНИЕ: Oracle BI Standard Edition One - это 32-битовое ПО. Программа может быть запущена на 64-битовых процессорах, которые поддерживают 32-битовую эмуляцию.

- Windows 2000 Server или Advanced Server (Service Pack 4 (5.0.2.195) или выше)
- Windows Server 2003

Оперативная память:

- минимум - 1.5 Гб
- рекомендуется 2.0 Гб или более

Жесткий диск:

- минимум - 6.0 Гб
- рекомендуется 6.0 Гб или более

Сеть:

Microsoft Windows:

- Microsoft Internet Explorer 6.x
- Microsoft Internet Explorer 7.0
- Firefox 1.5.x
- Firefox 2.0

ОБРАТИТЕ ВНИМАНИЕ: Иврит (HEB) доступен только в Microsoft Internet Explorer.

Linux:

- Firefox 1.5.x
- Firefox 2.0

Sun Solaris:

- Firefox 2.0

Apple Mac OS 10.x:

- Firefox 2.0

Oracle BI Standard Edition One поддерживает следующие исходные базы данных:

- IBM DB2 для UDB EE 8.2;
- IBM DB2 для z/OS 8 (режим NFM);
- Microsoft SQL Server 2000, 2005;

Прочие требования к программному обеспечению:

- Microsoft .NET Framework 2.0
Если Microsoft .NET Framework 2.0 ещё не установлен на необходимой машине, установщик Oracle BI Standard Edition One запустит установщика .NET Framework 2.0 в процессе установки Oracle BI Standard Edition One, даже в "тихом" режиме. Для установки .NET Framework 2.0 примите условия его использования, чтобы разрешить установщику продолжить работу.
- Windows Installer 3.0
Программа Windows Installer (msiexec.exe) позволяет вам устанавливать, изменять и производить операции в Windows Installer из командной строки. Во время установки Oracle BI Standard Edition One программе Windows Installer будет необходимо установить Microsoft .NET Framework 2.0. Если у вас ещё не установлен Microsoft .NET Framework 2.0, перед установкой Oracle BI Standard Edition One убедитесь, что у вас уже стоит Windows Installer.
- Adobe Acrobat Reader
Оптимальна версия 6.0. Oracle Business Intelligence использует Adobe Acrobat Reader на клиентских машинах для просмотра документов в формате PDF.
- Acrobat Flash Player
Оптимальна версия 7.0. Oracle Business Intelligence использует Acrobat Flash Player на клиентских машинах для показа графиков (Flash является форматом по умолчанию). Если Acrobat Flash Player не установлен на клиентской машине, конечному пользователю следует установить его, кликнув ссылку "Скачать плагин", которая будет показана вместо таблиц.
- Блокировка всплывающих окон
Блокировка всплывающих окон в браузерах должна быть отключена.

Oracle Database Lite

Процессор:

- Pentium 4, 3 ГГц ("Мобильный сервер" и "Мобильный набор разработчика") для систем на базе Windows
- процессор 3 ГГц для систем на базе Linux
- Sun SPARC Ultra 1 или выше, 300 МГц или лучше для систем на базе Sun
- процессоры A11, совместимые с AIX (64-bit) для систем на базе AIX
- процессор HP 9000 Series HP-UX для HP-UX 11.0 (64-bit) для систем на базе HP

Операционная система:

для систем на базе Windows:

- Windows Vista Ultimate, Windows XP, Windows 2003

для систем на базе Linux:

- Redhat Enterprise Linux Advanced Server 3.0, сертифицированный для x86
- Redhat Enterprise Linux Advanced Server 4.0 или 5.0, сертифицированный для x86 и AMD64

Привилегии (Grant, role)

Привилегии системного уровня

Существует две основные группы привилегий системного уровня: имеющие в составе имён слово ANY и не имеющие его. Привилегии со словом ANY в имени позволяют пользователю

Привилегии системного уровня могут быть даны либо непосредственно пользователю Oracle, либо роли, которая будет назначена пользователю Oracle.

Если нужно будет дать привилегии системного уровня каждому пользователю Oracle (включая любых пользователей Oracle, которые могут быть созданы в будущем), можно дать системную привилегию пользователю PUBLIC, что означает любого пользователя Oracle.

Привилегии системного уровня могут быть также предоставлены пользователю Oracle с использованием WITH ADMIN OPTION в конце оператора GRANT. Это позволяет пользователю Oracle, получившему такую системную привилегию, передавать её любому другому пользователю Oracle. По сути дела он становится ещё одним администратором этой привилегии.

Для изъятия привилегии или роли системного уровня у пользователя Oracle, можно использовать команду REVOKE. Ею можно определить сразу несколько системных привилегий и имён пользователей.

На любые объекты, созданные в то время, когда действовала эта привилегия, её отмена не повлияет. Кроме того, если системная привилегия изъята у пользователя PUBLIC, не будет затронут другой пользователь, которому эта системная привилегия была предоставлена непосредственно.

Любой пользователь предоставивший системную привилегию посредством WITH ADMIN OPTION, может предоставить эту системную привилегию другому пользователю Oracle – он, по сути дела, является дополнительным администратором этой системной привилегии.

Привилегии объектного уровня

Если пользователь Oracle владеет объектом наподобие таблицы, другим пользователям Oracle может быть разрешено использовать этот объект путём предоставления им одной или нескольких привилегий объектного уровня. Например, если пользователь user_1 намеревается позволить пользователю user_2 использовать его таблицу, но только для вставки и обновления строк с помощью команд INSERT или UPDATE, то пользователю user_2 могут быть даны привилегии UPDATE и INSERT.

✦ SELECT – позволяет другому пользователю Oracle сделать запрос к данным таблицы.

- ✦ INSERT – позволяет другому пользователю Oracle вставлять строки в таблицу с помощью команды INSERT.
- ✦ UPDATE – позволяет пользователю Oracle обновлять строки в таблице вне зависимости от того, были ли эти строки созданы этим пользователем или нет. Привилегия UPDATE
- ✦ DELETE – привилегия объектного уровня DELETE позволяет удалять из таблицы любые существующие строки. С использованием представления можно ограничить то, какие строки будут удалены.
- ✦ EXECUTE – даёт возможность пользователю Oracle, владеющему процедурным кодом базы данных (процедурами, функциями или пакетами), позволить другому пользователю Oracle вызывать его процедурные объекты.
- ✦ ALTER – даёт возможность пользователю Oracle изменить определение таблицы или последовательности.
- ✦ INDEX – позволяет пользователю Oracle создавать индексы на таблицы владельца.
- ✦ REFERENCES – позволяет пользователю Oracle обращаться к объектам другого пользователя.

В конце оператора GRANT для привилегии объектного уровня может быть определена фраза WITH GRANT OPTION, которая позволяет пользователю Oracle получившему эту привилегию, передать её другому пользователю Oracle.

Имеется ряд представлений словаря данных, которые показывают, какие привилегии объектного уровня были даны непосредственно пользователю и какие привилегии были предоставлены ему другими пользователями:

- ✦ USER_TAB_PRIVS
- ✦ USER_COL_PRIVS
- ✦ TABLE_PRIVILEGES
- ✦ COLUMN_PRIVILEGES

Представления словаря данных, начинающиеся с ALL_ и DBA_.

Роль(role) – совокупность системных привилегий и привилегий объектного уровня, которые были сгруппированы под одним именем, чтобы облегчить предоставление и отмену этих привилегий. Когда пользователю даётся роль, он получает все привилегии, связанные с ней.

Если пользователь Oracle удалён из базы данных командой DROP USER, можно сохранить присвоенные ему привилегии, если эти привилегии были назначены посредством роли. Затем можно переназначить эти привилегии любому новому пользователю. Если привилегии первоначально не были назначены посредством

роли, то при уничтожении первого пользователя будет потеряна вся относящаяся к нему информация.

С помощью ролей можно расширять и сужать набор привилегий, назначенных пользователю, разрешая и запрещая роли.

Новой роли могут быть назначены привилегии уже существующей роли (роль более высокого уровня наследует все привилегии роли более низкого уровня).

Как только пользователю Oracle был дан набор ролей, предусмотренные ими привилегии могут включаться и выключаться (обычно прикладными программами) посредством разрешения и запрещения ролей.

Вместе с новой базой данных автоматически создаются пять ролей. Это – CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE и IMP_FULL_DATABASE. Первые три предусмотрены для совместимости с предыдущими версиями программного обеспечения Oracle и обычно не требуются.

Управление пользователями базы данных

Центральное место в средствах защиты занимает учётная запись пользователя базы данных Oracle.

CREATE USER – это команда SQL, которая может использоваться для определения учётной записи Oracle в базе данных. После создания учётной записи пользователя Oracle она не может использоваться, пока пользователь не получит, по меньшей мере одну системную привилегию. Системная привилегия CREATE SESSION позволяет пользователю создавать сеанс по отношению к базе данных Oracle. Это – необходимая привилегия, которую должна иметь учётная запись пользователя, без неё учётная запись пользователя Oracle не может использоваться.

При первоначальном создании пользователя Oracle можно определить заданное по умолчанию табличное пространство, в котором будут создаваться объекты пользователя. Если заданное по умолчанию табличное пространство не определено, пользователю будет назначено табличное пространство SYSTEM в качестве заданного по умолчанию, которое будут использовать объекты базы данных. В составе оператора CREATE USER может использоваться фраза DEFAULT TABLESPACE для определения того, что объекты пользователя должны быть помещены в табличное пространство, отличное от SYSTEM. Пользователю Oracle также должна быть назначена квота, которая определяет, сколько памяти он может использовать в табличном пространстве.

Другой способ создания пользователя состоит в том, чтобы предоставить пользователю роли CONNECT, RESOURCE и DBA. Хотя это и быстрый метод, он включён, прежде всего, для совместимости с предыдущими версиями программного обеспечения Oracle. Команда CREATE USER – более предпочтительный метод, поскольку в одной команде можно указать и квоту и другие установки.

Можно использовать команду ALTER USER для изменения таких параметров пользователя, как пароль, заданные по умолчанию временные табличные пространства и квота памяти.

Для удаления пользователя из базы данных используется команда DROP USER, которая удаляет запись пользователя из словаря данных Oracle. Если пользователь Oracle владеет какими-либо объектами базы данных, можно либо

удалить каждый из объектов перед использованием команды DROP USER, либо использовать в DROP USER опцию CASCADE для автоматического уничтожения всех объектов при удалении учётной записи пользователя.

Аудит базы данных

Словарь данных каждой базы данных содержит таблицу с именем SUS.AUD\$, обычно называемую АУДИТОРСКИМ ЖУРНАЛОМ базы данных. Аудиторские записи, генерируемые как результат отслеживания предложений, привилегий или объектов, можно помещать как в аудиторскую таблицу базы данных, так и в аудиторский журнал операционной системы. Использование аудиторского журнала дает возможность просматривать выбранные порции аудиторского журнала с помощью predefined представлений словаря данных, а также использовать инструменты ORACLE, такие как SQL*ReportWriter, для генерации аудиторских отчетов.

Использование аудиторского журнала операционной системы помогает консолидировать аудиторские записи из многих источников, включая ORACLE и другие приложения. Поэтому исследование активности системы может быть более эффективным, так как аудиторские записи

Установка опций аудита

все опции аудита генерируют следующую общую информацию:

- ✦ имя пользователя, выполнявшего отслеживаемое предложение;
- ✦ код действия, указывающий выполненное предложение;
- ✦ объекты, адресуемые в отслеживаемом предложении;
- ✦ дату и время выполнения отслеживаемого предложения;

Аудиторский журнал не сохраняет информации о каких-либо значениях данных, которые могли быть вовлечены в отслеживаемое предложение; например, при аудите предложения UPDATE не сохраняются старые и новые значения данных. Однако, такой специализированный тип аудита можно осуществить для предложений DML, работающих с таблицами, с помощью триггеров базы данных.

ORACLE позволяет устанавливать опции аудита на трех уровнях:

- ✦ предложение аудита базируется на типе предложений SQL, например, на любых предложениях SQL по таблицам (что регистрирует каждое предложение CREATE, TRUNCATE и DROP TABLE)
- ✦ привилегия отслеживает использование конкретной системной привилегии, такой как CREATE TABLE
- ✦ объект отслеживает конкретные типы предложений на конкретных объектах, например, ALTER TABLE по таблице EMP

Групповые обозначения для опций аудита

Для удобства спецификации часто встречающихся групп связанных опций аудита предоставляются специальные обозначения. Эти обозначения сами не являются опциями; они просто позволяют указать одним словом целую группу опций в предложении AUDIT или NOAUDIT.

Включение и выключение аудита базы данных

Любой пользователь базы данных ORACLE может в любой момент установить опции аудита предложений, привилегий или объектов, но ORACLE не генерирует аудиторских записей и не помещает их в аудиторский журнал, если не включен режим аудита базы данных. Обычно за эту операцию отвечает администратор. Аудит базы данных включается и выключается параметром инициализации AUDIT_TRAIL в файле параметров базы данных. Этот параметр может быть установлен в следующие значения:

- ✳ DB - включает аудит базы данных и направляет все аудиторские записи в аудиторский журнал базы данных
- ✳ OS - включает аудит базы данных и направляет все аудиторские записи в аудиторский журнал операционной системы
- ✳ NONE - выключает аудит (умолчание)

Администратор защиты обязан контролировать рост аудиторского журнала и его размер. Когда аудит включен и генерируются аудиторские записи, аудиторский журнал растет за счет двух факторов:

- ✳ числа включенных опций аудита
- ✳ частоты выполнения отслеживаемых предложений

Для контроля за ростом аудиторского журнала вы можете использовать следующие методы:

- ✳ Включать и выключать аудит базы данных. Когда аудит включен, аудиторские записи генерируются и поступают в журнал; когда аудит выключен, аудиторские записи не генерируются.
- ✳ Жестко контролировать возможности осуществлять аудит объектов. Это можно делать двумя различными способами:
 - ✳ Всеми объектами владеет администратор защиты, привилегия AUDIT ANY никогда не назначается никаким другим пользователям. Все объекты схемы могут принадлежать схеме, соответствующий пользователь которой не имеет привилегии CREATE SESSION.
 - ✳ Все объекты содержатся в схемах, которые не соответствуют реальным пользователям базы данных (т.е. привилегия CREATE SESSION не назначена пользователям, одноименным со схемами), и администратор защиты является единственным лицом, имеющим системную привилегию AUDIT ANY.

✳ Все объекты содержатся в схемах, которые не соответствуют реальным пользователям базы данных (т.е. привилегия CREATE SESSION не назначена пользователям, одноименным со схемами), и администратор защиты является единственным лицом, имеющим системную привилегию AUDIT ANY.

Очистка аудиторских записей из аудиторского журнала

После того, как аудит был включен в течение некоторого времени, администратор защиты может удалить записи из аудиторского журнала, - как для того, чтобы освободить память, так и для облегчения управления этим журналом. Например, чтобы удалить ВСЕ записи из аудиторского журнала, введите следующее предложение:

```
DELETE FROM sys.aud$;
```

Если информация аудиторского журнала должна архивироваться для целей накопления истории, администратор защиты может скопировать соответствующие записи в нормальную таблицу базы данных или экспортировать аудиторскую таблицу в файл операционной системы. Удалять записи из аудиторского журнала базы данных может лишь пользователь SYS, т.е. пользователь, имеющий привилегию DELETE ANY TABLE (или пользователь, которому SYS передал привилегию DELETE по таблице SYS.AUD\$).

Защита аудиторского журнала

Осуществляя отслеживание подозрительной деятельности в базе данных, защищайте целостность записей аудиторского журнала, чтобы гарантировать точность и полноту аудиторской информации. Чтобы защитить аудиторский журнал от несанкционированных удалений, назначайте системную привилегию DELETE ANY TABLE только администраторам защиты. Чтобы отслеживать изменения, выполняемые над самим аудиторским журналом, организуйте аудит аудиторского журнала с помощью следующего предложения:

```
AUDIT INSERT, UPDATE, DELETE
```

```
ON sys.aud$
```

```
BY ACCESS;
```

Обеспечение целостности базы данных

Целостность данных определением правил проверки достоверности данных гарантирующих, что недействительные данные не попадут в ваши таблицы. Oracle позволяет определять и хранить эти правила для объектов БД, которых они касаются, таким образом, чтобы кодировать их только однажды. При этом они активируются всякий раз, когда какой-либо вид изменения проводится в таблице, независимо от того, какая программа выполняет вставки, модификации или удаления. Этот контроль осуществляется в форме ограничений и триггеров БД. Ограничения – это правила, применимые к таблицам во времена или после создания, распространяемые на то, как эти таблицы могут заполняться.

Ограничение целостности устанавливает правила на уровне БД, определяя набор проверок для таблиц системы. Эти проверки автоматически выполняются всякий раз, когда вызывается оператор вставки, модификации или удаления данных в таблице. Если какие-либо ограничения нарушены, операторы отменяются. Поскольку ограничения условности проверяются на уровне БД, они выполняются независимо от того, откуда были инициированы операторы вставки, модификации или удаления. Для таблиц можно задавать следующие типы ограничений целостности:

- ✦ NOT NULL
- ✦ PRIMARY KEY
- ✦ UNIQUE KEY
- ✦ FOREIGN KEY (REFERENCES)
- ✦ CHECK
- ✦ INDEX (ИНДЕКСЫ)
- ✦ TRIGGERS и PROCEDURES

NOT NULL. Это ограничение устанавливается для столбца, чтобы указать, что столбец должен иметь значение в каждой строке, т.е. некоторое непустое значение.

PRIMARY KEY (первичный ключ). Ограничение определяет столбец или группу столбцов, которую можно использовать для уникальной идентификации строки. Никакие две строки в таблице не могут иметь одинаковые значения столбцов первичного ключа. Кроме того, столбцы первичного ключа должны всегда содержать значение. Все эти условия гарантируют то, что в нашем распоряжении будет одна и только одна строка, соответствующая критериям связывания. Первичные ключи могут быть или именованные (пользователем) или неименованные (Oracle составляет имя сам). В первичных ключах не могут использоваться столбцы типа: raw, long, long raw.

UNIQUE (уникальный). Ограничение UNIQUE используется для определения того, что значения в столбце не должны повторяться в другой строке этой таблицы, определяет вторичный ключ для таблицы. Это столбец или группа столбцов, которые можно использовать как уникальную идентификацию строки. Никакие две строки не могут иметь одинаковые значения для столбца или столбцов ключа UNIQUE. Столбцы для ограничения UNIQUE не обязательно NOT NULL. Можно сформировать ограничение таблицы, указав, что в таблице не должна повторяться комбинация столбцов. К примеру: можно в начале объявить стандартно emp_id number(5), person_id date а под конце объявить что: unique(emp_id, person_id) – и получится, что сочетание значений этих полей, должно быть уникальным в каждой строке.

FOREIGN KEY (внешний ключ). FOREIGN KEY, устанавливает отношение целостности между таблицами. Оно требует, чтобы столбец или набор столбцов в одной таблице совпадал с первичным или вторичным ключом другой таблицы. С

момента создания внешнего ключа ссылающегося на первичный ключ некой таблицы удаление таблицы будет – запрещено. И обойти это ограничение можно только удалив ограничение. Внешние ключи могут быть именованные или неименованные.

CHECK. Ограничение CHECK определяет логику проверки, которая должна жать результат true (истина) для оператора вставки, модификации или удаления из таблицы. Ограничение CHECK гарантирует, что значение в измененной строке удовлетворяют заданному набору проверок правильности.

ИНДЕКСЫ (INDEX). Ограничения PRIMARY KEY и UNIQUE автоматически создают индексы на столбцах, для которых они определены, если ограничение активизируется при создании. Если индекс уже существует на столбцах, которые составляют ограничение PRIMARY KEY и UNIQUE, то использует именно этот индекс и Oracle не может создать новый.

TRRIGERS(Триггеры) – с программный элемент хранимый в БД выполняемый автоматически, в определенных ситуациях, не имеющий входных или выходных параметров, что в конечном итоге и является причиной невозможности вызвать его явно, непосредственно, его вызывает только сама база данных Oracle. Выходные данные триггера должны быть также применимы к БД, а не возвращены вызывающей программе или отображены на экране.

Таким образом, целостность базы данных может быть рассмотрена на трех уровнях:

- 1 На уровне типа данных (т.е. соответствия типов данных)
- 2 На уровне ключей (к примеру, соответствие первичных и внешних ключей)
- 3 На уровне триггеров, процедур и/или функций(к примеру, триггера отвечают только за свои области данных).

Запуск и останов базы данных

Для запуска базы данных или инстанции(экземпляра) используйте либо диалоговое окно Start Up Instance, либо команду STARTUP (после того, как соединитесь с ORACLE как INTERNAL). Вы можете запустить экземпляр и базу данных различными способами:

- ☀ запустить экземпляр без монтирования базы данных
- ☀ запустить экземпляр и смонтировать базу данных, но оставить ее закрытой
- ☀ запустить экземпляр, смонтировать и открыть базу данных в одном из следующих режимов:
 - ☀ неограниченном режиме (доступна всем пользователям)
 - ☀ ограниченном режиме RESTRICTED (доступна только АБД)

Кроме того, вы можете форсировать запуск экземпляра, либо заставить экземпляр начать немедленно после запуска полное восстановление носителя.

Прежде, чем запускать экземпляр, нужно подключиться как INTERNAL; также может понадобиться указать, для какой базы данных вы запускаете экземпляр, и специфицировать файл параметров.

Вы также должны подключиться как INTERNAL. Это условие обязательно, независимо от того, используете ли вы графический интерфейс SQL*DBA или команды SQL.

Запуск экземпляра без монтирования базы данных

Можно запустить экземпляр без монтирования базы данных; обычно это требуется лишь при создании базы данных. Чтобы сделать это, используйте одну из следующих опций SQL*DBA:

- ☀ кнопку Nomount в диалоговом окне Start Up Instance
- ☀ команду STARTUP с опцией NOMOUNT

Запуск экземпляра и монтирование базы данных

Вы можете запустить экземпляр с монтированием базы данных, но не открывать базу данных, чтобы выполнить специфические операции сопровождения. Например, база данных должна быть смонтирована, но не открыта, во время выполнения следующих задач:

- ☀ переименования файлов данных
- ☀ добавления, удаления или переименования файлов журнала
- ☀ включения и выключения опций архивирования журнала
- ☀ полного восстановления базы данных

Для запуска экземпляра и монтирования базы данных без ее открытия используйте одну из следующих опций SQL*DBA:

- ☀ кнопку Mount в диалоговом окне Start Up Instance
- ☀ команду STARTUP с опцией MOUNT

При использовании команды STARTUP с параметром NOMOUNT производится только запуск фоновых процессов Oracle и распределение SGA в памяти. В состоянии NOMOUNT базу данных может использовать только DBA. Опция NOMOUNT обычно используется только при создании базы данных.

Запуск экземпляра, монтирование и открытие базы данных

Нормальная работа базы данных означает, что экземпляр запущен, а база данных смонтирована и открыта. Это позволяет всем действительным пользователям

соединяться с базой данных и выполнять типичные операции, требующие доступа к данным. Для запуска экземпляра с монтированием базы данных и ее открытием используйте одну из следующих опций SQL*DBA:

- ☀ кнопку Open в диалоговом окне Start Up Instance

- ☀ команду STARTUP с опцией OPEN

Задание имени базы данных

При запуске экземпляра базы данных специфицируйте имя базы данных, которая будет монтироваться, одним из следующих способов:

- ☀ введите имя базы данных в поле Database Name в диалоговом окне Start Up Instance

- ☀ укажите имя базы данных в команде STARTUP

Задание файла параметров

При запуске экземпляра базы данных выберите файл параметров для инициализации характеристик экземпляра, одним из следующих способов:

- ☀ введите имя файла параметров в поле Parameter File в диалоговом окне Start Up Instance

- ☀ укажите полное имя файла параметров в опции PFILE команды STARTUP

Спецификации имен файлов зависят от операционной системы. Если имя файла не указано, ORACLE использует умалчиваемое имя файла.

Форсированный запуск экземпляра

Форсированный запуск, описанный ниже, следует применять ТОЛЬКО в следующих случаях:

- ☀ Текущую работающую инстанцию не удастся закрыть с помощью опций Normal или Immediate меню Shut Down (или эквивалентных опций предложения SHUTDOWN)

- ☀ Экземпляр не удастся запустить обычным способом

Если вы попали в такую ситуацию, обычно удастся решить проблему путем запуска нового экземпляра в форсированном режиме, используя одну из следующих опций SQL*DBA:

- ☀ переключатель Force в диалоговом окне Start Up Instance

- ☀ команду STARTUP с опцией FORCE

Эти опции в действительности сначала останавливают текущую инстанцию, а затем запускают новую инстанцию (возможно, с монтированием и открытием базы данных).

Запуск экземпляра с монтированием базы данных и полным восстановлением носителя

Если вы знаете, что необходимо восстановление носителя, то вы можете запустить инстанцию так, чтобы она автоматически начала процесс восстановления, используя одну из следующих опций SQL*DBA:

- ✦ переключатель Recover в диалоговом окне Start Up Instance

- ✦ команду STARTUP с опцией RECOVER

Запуск в монопольном или параллельном режимах

Если ваш сервер ORACLE позволяет обращаться к одной базе данных из нескольких инстанций, то вы должны выбрать монтирование базы данных в монопольном или параллельном режимах.

Автоматический запуск базы данных при запуске операционной системы

На многих установках одна или несколько инстанций и баз данных ORACLE автоматически запускаются вслед за загрузкой ОС. Процедуры, позволяющие организовать такой режим, специфичны для каждой операционной системы.

Запуск удаленного экземпляра

Если ваш локальный сервер ORACLE является частью распределенной базы данных, то вам может понадобиться запускать удаленную инстанцию и базу данных. Процедуры запуска и останова удаленных инстанций широко варьируются в зависимости от коммуникационного протокола и операционной системы.

Остановка базы данных

Чтобы инициировать остановку базы данных, используйте либо меню Shut Down, либо команду SHUTDOWN в SQL*DBA.

Остановка базы данных в нормальных обстоятельствах

Нормальная остановка базы данных протекает следующим образом:

- ✦ После получения команды на останов не допускаются новые соединения с базой данных.

- ✦ Прежде чем остановить базу данных, ORACLE ждет отсоединения от нее всех текущих соединенных пользователей.

- ✦ Очередной запуск базы данных не потребует никаких процедур восстановления инстанции.

Для останова базы данных в нормальных обстоятельствах используйте одну из следующих опций SQL*DBA:

- ✦ опцию Normal меню Shut Down
- ✦ команду SHUTDOWN с опцией NORMAL

Немедленная остановка базы данных

В чрезвычайных обстоятельствах вы можете остановить базу данных немедленно. Используйте этот способ останова лишь в случаях, подобных следующим:

- ✦ Скоро произойдет отключение питания.
- ✦ База данных или одно из ее приложений работает неверно.
- ✦ Немедленный останов базы данных протекает следующим образом:
- ✦ Обработка текущих предложений SQL от клиентов немедленно прекращается.
- ✦ Все неподтвержденные транзакции откатываются. (Если есть длинные неподтвержденные транзакции, этот способ останова может оказаться достаточно продолжительным, несмотря на свое название.)
- ✦ ORACLE не ждет отключения текущих соединенных пользователей; ORACLE неявно откатывает активные транзакции и разрывает все пользовательские соединения.
- ✦ Очередной запуск базы данных может потребовать восстановления инстанции (которое ORACLE выполнит автоматически).

Для немедленного останова базы данных используйте одну из следующих опций SQL*DBA:

- ✦ опцию Immediate меню Shut Down
- ✦ команду SHUTDOWN с опцией IMMEDIATE

Примеры остановки базы данных

Эта секция приводит примеры останова базы данных и инстанции через интерфейс меню и команды SQL*DBA. Во всех примерах предполагается, что АБД уже подключен как INTERNAL.

Меню Shut Down останавливает базу данных.

Команда SHUTDOWN эквивалентна меню Shut Down. Например, следующее предложение является командным эквивалентом меню Shut Down.

Различные режимы работы базы данных

Запуск однопроцессных и многопроцессных инстанций

В большинстве операционных систем вы можете запускать инстанцию ORACLE либо в однопроцессном, либо в многопроцессном режиме, независимо от того, как ORACLE был инсталлирован или запускался последний раз. Если компьютер, на котором выполняется сервер ORACLE, поддерживает многопроцессность, то инстанции баз данных обычно запускаются в многопроцессном режиме, так что много пользователей могут одновременно обращаться к разделяемой базе данных; однопроцессные же инстанции поддерживают лишь одного пользователя в каждый момент. Однако, в некоторых экспериментальных ситуациях, вы можете найти полезным запустить инстанцию в однопроцессном режиме. Некоторые операционные системы (такие как MS-DOS) не поддерживают многопроцессности или разделяемой памяти; в таких системах однопроцессная инстанция является единственной возможностью.

Резервное копирование базы данных

Если над базой данных производят любое из ниже перечисленных структурных изменений, базы данных, непосредственно перед изменениями и после делается соответствующее копирование базы данных:

- ✦ Создание или удаление табличного пространства
- ✦ Добавление или переименование (перемещение) файла данных в существующем табличном пространстве
- ✦ Добавление, переименование(перемещение) или удаление группы или члена онлайн-журнала повторения.
- ✦ Если база данных работает в режиме ARCHIVELOG, то до и после структурного изменения базы данных требуется лишь резервное копирование управляющего файла базы данных (с помощью команды ALTER DATABASE с опцией BACKUP CONTROLFILE). Можно скопировать и другие части базы данных.
- ✦ Если база данных работает в режиме NOARCHIVELOG, то непосредственно перед и после изменения базы данных требуется сделать полное копирование файла базы данных, включая все файлы данных, файлы журнала повторения и управляющие файлы.

Существует, по большому счету, два вида резервного копирования:

1 Непротиворечивое (холодное) резервное копирование, ситуация когда, копии создаются, в случае закрытой БД (close) для пользователей. Копия базы данных, созданной в автономном режиме, содержит: все файлы данных, журналы повторов и управляющие файлы. После останова БД, все файлы базы данной по средствам ОС копируются на один из backup дисков.

2 Резервное(горячее) копирование в оперативном режиме, к примеру, когда БД работает в архивном режиме ARCHIVELOG, БД все время находится в оперативном режиме таким образом доступна пользователям.

☀ Остановка экземпляра БД Oracle – в режиме shutdown normal (игнорирование, новых подключений и ожидание отключение все зарегистрированных пользователей) или shutdown immediate (немедленное прерывание всех соединений, выполнение операции отката на всех транзакциях ожидающих обработки)

- ☀ Копирование всех физических файлов, относящихся к базе данных, управляющие файлы, файлы журнала обновления и файлы базы данных.
- ☀ Закончить работу, перезагрузить базу данных
- ☀ Перевод табличного пространства в режим резервного копирования.
- ☀ Копирование всех файлов базы данных, связанных с табличным пространством.
- ☀ Выведение табличное пространство из режима резервного копирования.
- ☀ Повторение действий с первого по третье, пока не будет выполнено резервное копирование всех табличных пространств.
- ☀ Копирование управляющего файла.
- ☀ Копирование оперативного журнала обновления.

Сопоставление режима ARCHIVELOG и режима NOARCHIVELOG

В режиме ARCHIVELOG:

- ☀ Требуется дополнительное дисковое пространство
- ☀ Управление архивными журналами влечет за собой дополнительные административные непроизводительные затраты.
- ☀ Применимо горячее резервное копирование.
- ☀ В случае отказа носителя может быть выполнено полное восстановление базы данных.

В режиме NOARCHIVELOG:

- ☀ Не требуется дополнительное дисковое пространство или непроизводительные затраты.
- ☀ Может использоваться только холодное резервное копирование.
- ☀ В случае отказа теряется вся работа, выполненная со времени последнего резервного копирования.

Включение и выключение архивирования

Вы устанавливаете первоначальный режим архивирования базы данных во время ее создания. В большинстве случаев, во время создания базы данных вы можете выбрать режим по умолчанию NOARCHIVELOG, потому что нет необходимости архивировать информацию, генерируемую за этот период. После того как база данных создана, решите, нужно ли изменить первоначальный режим

архивирования. После того, как база данных создана, всегда можно переключать режим архивирования базы данных. Однако, как правило, следует выбрать постоянный режим работы базы данных.

Включение автоматического архивирования




Чтобы автоматическое архивирование заполненных групп было включено установите в TRUE значение параметра LOG_ARCHIVE_START в файле параметров базы данных INIT.ORA:

LOG_ARCHIVE_START = TRUE Это значение будет иметь эффект при очередном запуске базы данных.

Выключение автоматического архивирования

Вы можете выключить автоматическое архивирование журнала в любой момент. Однако, выключив автоматическое архивирование, вы должны вручную, периодически и своевременно, архивировать заполняемые группы журнала. Если база данных работает в режиме ARCHIVELOG, автоматическое архивирование выключено, а группы журнала заполняются, но не архивируются, то процесс LGWR не сможет повторно использовать неактивные группы журнала, чтобы продолжать запись информации повторения. Поэтому работа базы данных будет временно приостановлена до тех пор, пока не будет выполнено необходимое архивирование. Автоматическое архивирование может быть выключено как до, так и после запуска инстанции.

Если база данных работает в режиме ARCHIVELOG, то можно копировать индивидуальное табличное пространство или даже индивидуальный файл. Эта возможность полезна, если одна часть базы данных используется более интенсивно, чем другие, - например, табличное пространство SYSTEM или табличные пространства, содержащие сегменты отката. Если сбой диска повреждает один из таких файлов данных, для его реставрации может быть использована ранняя копия, и меньшее число изменений необходимо применить при прокрутке вперед, чтобы

Oracle Corporation	
	
Тип	Публичная компания
Листинг на бирже	NASDAQ: ORCL
Девиз компании	Hardware and Software, Engineered To Work Together
Год основания	1977
Основатели	Ларри Эллисон Боб Майнер Эд Оутс
Расположение	 США: Редвуд Шорз, Калифорния
Ключевые фигуры	Ларри Эллисон (CEO) Сафра Кац (президент) Марк Хёрд (президент) ^[1]
Отрасль	Производство программного обеспечения
Продукция	Программное и аппаратное обеспечение для организаций
Оборот	▲ \$35,622 млрд (2011)
Операционная прибыль	▲ \$12,033 млрд (2011)
Чистая прибыль	▲ \$8,547 млрд (2011)
Число сотрудников	108 тыс. (2011) ^[2]
Сайт	oracle.com 

восстановить файл к состоянию на момент сбоя, т.е. время, затрачиваемое на восстановление, сокращается.

Немного истории о самой организации

Oracle, Oracle Corporation - американская корпорация, крупнейший в мире разработчик программного обеспечения для организаций, крупный поставщик серверного оборудования.

Компания специализируется на выпуске систем управления базами данных, связующего программного обеспечения и бизнес-приложений (ERP- и CRM-систем, специализированных отраслевых приложений). Наиболее известный продукт компании — Oracle Database, который компания выпускает с момента своего основания. С 2008 года корпорация освоила выпуск интегрированных аппаратно-программных комплексов, а с 2009 года в результате поглощения Sun Microsystems стала производителем серверного оборудования, до этого компания выпускала исключительно программное обеспечение.

Компания основана в 1977 году. Имеет подразделения в более чем 145 странах. По состоянию на 2011 год насчитывает 108 тыс. сотрудников. Штаб-квартира корпорации расположена в США, в штате Калифорния, рядом с Сан-Франциско. **1970-е**

Компания основана в 1977 году в городе Санта-Клара, Калифорния под наименованием *SDL* (аббревиатура от англ. *Software Development Laboratories*) *Ларри Эллисоном, Бобом Майнером* (англ. *Bob Miner*) и *Эдом Оутсом* (англ. *Ed Oates*). Все трое основателей работали до этого года в компании Амрех над проектом для ЦРУ США с кодовым названием *Oracle*.

Это кодовое название присвоили разработанной в первые месяцы существования *SDL* СУБД. Первый выпуск СУБД *Oracle v2* получил номер версии *v2* по маркетинговым соображениям. *Oracle v2* была написана на ассемблере для PDP-11, работала под управлением операционной системы RSX. В середине 1979 года авиабаза Райт-Патерсон ВВС США приобрела *Oracle v2* и стала первым заказчиком компании. К этому же времени относится переименование *SDL* в *RSI* (аббревиатура от англ. *Relational Software, Inc.*). *Oracle v2* считается первой коммерческой СУБД с поддержкой языка запросов SQL, и одной из первых реляционных СУБД. Также отмечается влияние на Oracle ранее разработанной в IBM СУБД System R.

Сравнительные характеристики базы данных Oracle с различными базами данных, таких как MSSQL и PostgreSQL

Как было отмечено выше, выбор конкретной архитектуры построения информационной системы включает два основных компонента: выбор серверной платформы (выбор серверной ОС и СУБД) и выбор платформ для клиентских рабочих мест. В данном разделе более подробно остановимся на особенностях выбора конкретной СУБД. При выборе базы данных очень важно выбрать базу данных, которая в наибольшей степени соответствует предъявляемым к информационной системе требованиям, т.е. необходимо определиться какая модель автоматизации реализуется (автоматизация документооборота или

бизнес - процессов). В первую очередь при выборе СУБД необходимо принимать во внимание следующие факторы:

1. максимальное число пользователей одновременно обращающихся к базе;
2. характеристики клиентского ПО;
3. аппаратные компоненты сервера;
4. серверную операционную систему;
5. уровень квалификации персонала.

На сегодня известно большое число различных серверов баз данных SQL. Остановимся более подробнее на следующих четырех ведущих серверных СУБД – Oracle11g, Microsoft SQL, PostgreSQL и сравним их в работе на каждом из основных этапов функционирования:

1. конфигурирование системы,
2. мониторинг,
3. настройка,
4. обработка запросов,
5. разработка серверных и клиентских модулей.

Oracle11g

- **СУБД Oracle Database 11g** поставляется в четырех различных редакциях, ориентированных на различные сценарии разработки и развертывания приложений. Кроме того, корпорация Oracle предлагает несколько дополнительных программных продуктов, расширяющих возможности Oracle Database 11g для работы с конкретными прикладными пакетами. Ниже перечислены существующие редакции СУБД Oracle Database 11g: Oracle Database 11g Standard Edition One характеризуется беспрецедентной простотой эксплуатации, мощностью и выгодным соотношением цены и производительности для приложений масштаба рабочих групп, отдельных подразделений или приложений, работающих в среде интернет/интранет. Работая в различных средах, начиная от односерверных конфигураций для малого бизнеса и заканчивая распределенными средами крупных филиалов, Oracle Database 11g Standard Edition One обладает всеми функциональными возможностями для обеспечения работы критических для бизнеса приложений. Редакция Standard Edition One лицензируется только для серверов, имеющих не более двух процессоров.
- **Oracle Database 11g Standard Edition (SE)** обеспечивает столь же беспрецедентную простоту эксплуатации, мощь и производительность, что и редакция Standard Edition One, поддерживая работу более мощных вычислительных систем с использованием технологии кластеризации сервисов Real Application Clusters. Эта редакция лицензируется для использования на одном сервере с числом процессоров, не превышающим четырех, или на серверном кластере, поддерживающем не более четырех процессоров.
- **Oracle Database 11g Enterprise Edition (EE)** обеспечивает эффективное, надежное и безопасное управление данными таких критически важных для бизнеса приложений, как онлайн-среды, выполняющие масштабную обработку транзакций (OLTP), хранилища данных с высокой интенсивностью потока запросов, а также ресурсоемкие интернет-приложения. Редакция Oracle Database Enterprise Edition предоставляет инструментальные средства и функции, обеспечивающие соответствие требованиям современных корпоративных приложений в области доступности и масштабируемости. Эта

редакция содержит все компоненты Oracle Database, а также допускает расширение посредством приобретения дополнительных модулей и приложений, описанных далее в этой статье.

- **Oracle Database 11g Personal Edition** поддерживает однопользовательскую разработку и развертывание приложений, полностью совместимых с редакциями Oracle Database Standard Edition One, Oracle Database Standard Edition и Oracle Database Enterprise Edition. Предоставив отдельным пользователям мощную функциональность пакета Oracle Database 11g, корпорация Oracle создала базу данных, сочетающую мощь популярнейшей в мире СУБД и простоту эксплуатации, которую вы вправе ожидать от приложения для настольного ПК.

СУБД Microsoft SQL Server

Важнейшие характеристики данной СУБД - это:

1. простота администрирования,
2. возможность подключения к Web,
3. быстрое действие и функциональные возможности механизма сервера СУБД,
4. наличие средств удаленного доступа,

В комплект средств административного управления данной СУБД входит целый набор специальных мастеров и средств автоматической настройки параметров конфигурации. Также данная БД оснащена замечательными средствами тиражирования, позволяющими синхронизировать данные ПК с информацией БД и наоборот. Входящий в комплект поставки сервер OLAP дает возможность сохранять и анализировать все имеющиеся у пользователя данные. В принципе данная СУБД представляет собой современную полнофункциональную базу данных, которая идеально подходит для малых и средних организаций. Необходимо заметить, что SQL Server уступает другим рассматриваемым СУБД по двум важным показателям: программируемость и средства работы. При разработке клиентских БД приложений на основе языков Java, HTML часто возникает проблема недостаточности программных средств SQL Server и пользоваться этой СУБД будет труднее, чем системами DB2, Informix, Oracle или Sybase. Общемировой тенденцией в XXI веке стал практически повсеместный переход на платформу LINUX, а SQL Server функционирует только в среде Windows. Поэтому использование SQL Server целесообразно, по нашему мнению, только если для доступа к содержимому БД используется исключительно стандарт ODBC, в противном случае лучше использовать другие СУБД.

СУБД PostgreSQL

PostgreSQL - это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных.

Основные возможности и функциональность

Надежность PostgreSQL является проверенным и доказанным фактом и обеспечивается следующими возможностями:

- полное соответствие принципам **ACID** - атомарность, непротиворечивость, изолированность, сохранность данных.
 - Atomicity - транзакция рассматривается как единая логическая единица, все ее изменения или сохраняются целиком, или полностью откатываются.
 - Consistency - транзакция переводит базу данных из одного непротиворечивого состояния (на момент старта транзакции) в другое непротиворечивое состояние (на момент завершения транзакции).
Непротиворечивым считается состояние базы, когда выполняются все ограничения физической и логической целостности базы данных, при этом допускается нарушение ограничений целостности в течение транзакции, но на момент завершения все ограничения целостности, как физические, так и логические, должны быть соблюдены.
 - Isolation - изменения данных при конкурентных транзакциях изолированы друг от друга на основе системы версионности
 - Durability - PostgreSQL заботится о том, что результаты успешных транзакций **гарантировано** сохраняются на жесткий диск вне зависимости от сбоев аппаратуры.

Производительность PostgreSQL основывается на использовании индексов, интеллектуальном планировщике запросов, тонкой системы блокировок, системой управления буферами памяти и кэширования, превосходной масштабируемости при конкурентной работе.

- **Поддержка индексов**
 - Стандартные индексы - B-tree, hash, R-tree, GiST (обобщенное поисковое дерево, которое позволяет на пользовательском уровне создавать свои типы данных и индексные методы доступа к ним)
 - Частичные индексы (**partial indices**)
 - Функциональные индексы

Таблица сравнений 1

	SQL Server 2008	Oracle11g
Административное управление	Хорошо	Отлично
Оперативная память	3 ГБ. Для 64-bit ОС Максимум 64 ГБ	Неограниченно
Простота обслуживания	Хорошо	Отлично
Минимальное количество пользовательских лицензий	5	5
Количество виртуальных машин	1 виртуальная машина на 1 процессорную лицензию	Неограниченно
Поддержка различных кодировок	Только Windows и Unicode	Есть

Одновременный доступ нескольких пользователей	Хорошо	Отлично
Рекомендованная стоимость 5 пользовательских лицензий	Ориентировочно при среднерыночной маржинальности: $950 + 5 \cdot 175 = 1825 \$$	$54350 = 1750 \$$
Возможности программирования	Приемлемо	Отлично
Хранимые процедуры и триггеры	Хорошо	Отлично
Внутренний язык программирования	Плохо	Отлично
Построение баз данных	Хорошо	Отлично
Язык SQL	Отлично	Отлично

Таблица сравнений 2

	Oracle Standard Edition One, Oracle Standard Edition	PostgreSQL
Основные отличия		
Поддержка Windows	Windows (все версии)	только Win2000 SP4, WinXP, Win2003
Поддержка других операционных систем	Linux, Solaris, Solaris SPARC, AIX, HP-UX Itanium, HP-UX PA-RISC, z/Linux, z/OS, Mac OS	Linux, Solaris, Mac OS X, FreeBSD, QNX 4.25, QNX 6
Поддержка 32- и 64-bit	Есть	Есть
Распараллеливание запросов по разным ядрам	Есть	Нет
Скорость работы с большими таблицами (миллионы строк)	Быстро	Медленно, особенно если много индексов
Стоимость владения		

Рекомендованная стоимость лицензий	Лицензируется на пользователей (NUP) или на процессор: <ul style="list-style-type: none"> Standard Edition One: NUP – 180\$ Процессор – 5800\$ Standard Edition: NUP – 350\$ Процессор – 17500\$ 	Бесплатно. Под лицензией BSD
Стоимость ежегодной технической поддержки при лицензировании на процессор	22% от стоимости лицензий. Например: 1 сокет: 1276 \$ 4 сокета: 3850 *4=15400\$	Коммерческая поддержка EnterpriseDB: 1 сокет: 1250\$ 4 сокета: 3750 *4 = 15000 \$
Стоимость ежегодной технической поддержки при лицензировании на пользователей (NUP)	22% от стоимости лицензий. Например, для 5 пользователей на 2-х процессорный сервер: 77*5 = 385\$ (редакция Standard Edition One)	Необходимо покупать техническую поддержку на процессоры. И даже в случае 5 пользователей для 2-х процессорного сервера платить: 1250 *2 = 2500 \$
Администрирование		
Доступ к патчам	Требует наличия технической поддержки	Бесплатно
Доступ к консультациям сотрудников технической поддержки	Требует наличия технической поддержки	Требует наличия технической поддержки
Средство администрирования	Web-интерфейс, доступ через браузер	Клиентская утилита pgAdmin, необходима инсталляция. В терминологии Oracle, это скорее средство разработки (аналог бесплатного Oracle SQL Developer), чем средство администрирования.
Настройка прав пользователей	Гибкие возможности	Ограничена. Малое количество ролей. Нет возможности управлять тем, что может делать одна роль с другими ролями