

Экзамен по СУБД

1. Понятие СУБД. Основные категории СУБД. Архитектура ANSI-SPARC.

СУБД — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Основные функции СУБД:

- Управление данными на дисках.
- Управление данными в ОП (в т.ч., кэширование).
- Журналирование изменений, резервное копирование, восстановление после сбоев.
- Поддержка языков БД (DML + DDL).

Состав СУБД:

- Ядро — отвечает за управление данными.
- Процессор языка БД — транслирует запросы с высокоуровневого языка на низкоуровневый.
- Подсистема поддержки времени исполнения.
- Сервисные программы.

Классификация СУБД:

По модели данных:

- Иерархические — данные представляются в виде дерева. Пример — LDAP / AD, реестр Windows.
- Сетевые — используют сетевую модель данных. Частный случай — графовые СУБД. Примеры — HypergraphDB, OrientDB.
- Объектно-ориентированные — используют ОО-модель данных. Пример — InterSystems Caché.
- Реляционные и объектно-реляционные — используют реляционную модель данных (возможно, с частичной поддержкой ООП). Примеры — Oracle, MySQL, PostgreSQL.

По степени распределённости:

- локальные;
- распределённые.

По способу доступа к БД.

- Файл-серверные — данные находятся на файл-сервере, СУБД — на каждом клиентском компьютере. Примеры — M\$ Access, dBase, FoxPro.
- Клиент-серверные — СУБД находятся на сервере вместе с данными. Примеры — Oracle, M\$ SQL Server, Caché.
- Встраиваемые — СУБД встраивается в приложение, хранит только его данные и не требует отдельной установки. Примеры — SQLite, BerkeleyDB.

Архитектура АНСИ-СПАРК

Архитектура ANSI-SPARC (также **трёхуровневая архитектура**) определяет принцип, согласно которому рекомендуется строить **системы управления базами данных**(СУБД).

Проект архитектуры был выдвинут в **1975 году** подкомитетом **SPARC ANSI**.

Три уровня СУБД:

1. внешний (пользовательский)
2. промежуточный (концептуальный)
3. внутренний (физический)

В основе архитектуры ANSI-SPARC лежит концептуальный уровень. Концептуальный уровень описывает данные и их взаимосвязи с наиболее общей точки зрения, — **концепции архитекторов** базы, используя **реляционную** или другую **модель**.

Внутренний уровень позволяет скрыть подробности физического хранения данных (носители, файлы, таблицы, триггеры ...) от концептуального уровня. Отделение внутреннего уровня от концептуального обеспечивает так называемую **физическую независимость** данных.

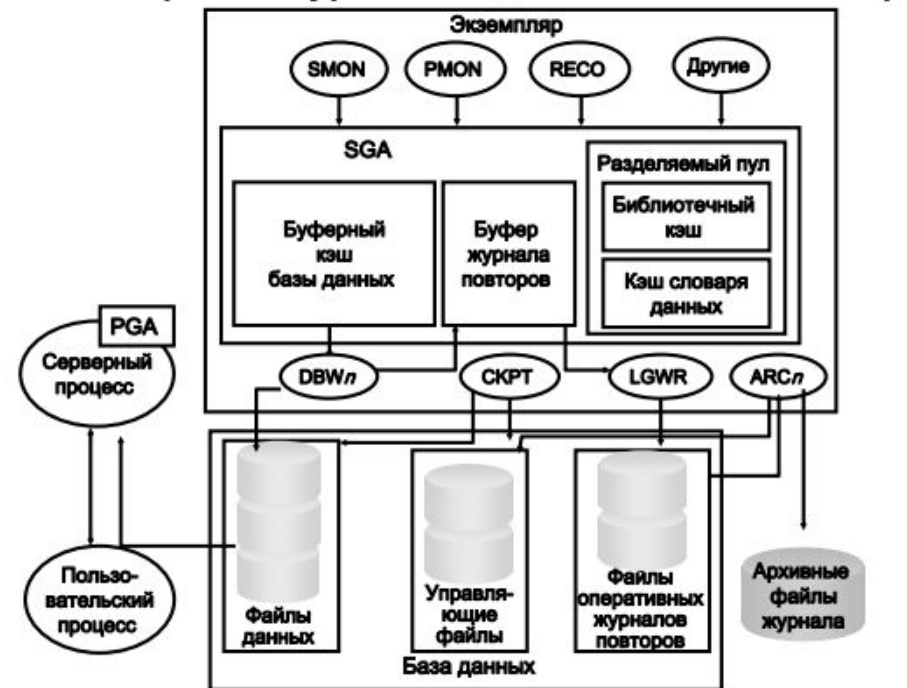
На внешнем уровне описываются различные подмножества элементов концептуального уровня для представлений данных различным пользовательским программам. Каждый **пользователь** получает в свое распоряжение часть представлений о данных, но полная концепция скрыта. Отделение внешнего уровня от концептуального обеспечивает **логическую независимость** данных.

2. СУБД Oracle. Архитектура. Подключение, взаимодействие с БД.

- Исторически первая и наиболее распространённая коммерческая СУБД на основе языка SQL.
- По классификации — объектно-реляционная распределённая клиент-серверная СУБД.
- Пример архитектуры Enterprise-level решения.
- Первая версия (v2) была выпущена в 1979 г.
- Актуальная версия — 12с («cloud» — «облако»).
- Начиная с версии 3 реализована поддержка транзакций.
- В версии 7 появилась поддержка PL/SQL.
- В версии 8 реализована поддержка ООП.
- В версии 9 реализована технология RAC (Real Application Cluster) — появилась возможность реализации кластерных БД.



Архитектура базы данных Oracle: обзор



База данных Oracle состоит из экземпляра и связанных с ним базами данных. Экземпляр состоит из структур памяти и фоновых процессов. При каждом запуске экземпляра выделяется совместно используемая область памяти, которая называется глобальной системной областью (System Global Area, SGA), а также запускаются фоновые процессы. База данных состоит как из физических, так и из логических структур. Так как физические и логические структуры разделены, физическими хранилищами данных можно управлять, не влияя на логические структуры хранения.

Подключение к базе данных

Подключения и сеансы тесно связаны с пользовательскими процессами, но имеют различные значения. *Подключение* – это канал связи между пользовательским процессом и экземпляром базы данных Oracle.

Сеанс представляет состояние подключения текущего пользователя к экземпляру базы данных. Сеанс длится с момента подключения пользователя и до момента отключения или выхода из приложения базы данных.

Для одного пользователя базы данных Oracle может быть создано множество параллельных сеансов с одним и тем же именем пользователя. Например, пользователь с именем/паролем HR/HR может подключиться к одному экземпляру базы данных Oracle несколько раз.

- **Подключение: взаимодействие пользовательского процесса и экземпляра**
- **Сеанс: конкретное подключение пользователя к экземпляру посредством пользовательского процесса**

Взаимодействие

1. Экземпляр, запущенный на узле, на котором установлена база данных Oracle, часто называется хостом или сервером базы данных.
2. Пользователь запускает приложение, порождающее пользовательский процесс. Приложение пытается установить соединение с сервером.
3. Сервер запускает прослушиватель с соответствующим обработчиком Oracle Net Services. Сервер обнаруживает запрос от приложения на соединение и создает выделенный серверный процесс по требованию пользовательского процесса.
4. Пользователь выполняет SQL-оператор DML-типа и подтверждает транзакцию.
5. Серверный процесс получает оператор и проверяет разделяемый пул (компонент SGA) на наличие любой разделяемой области SQL, которая содержит подобный SQL-оператор. Если разделяемая область SQL найдена, серверный процесс проверяет полномочия доступа пользователя к запрошенным данным, и оператор обрабатывается с помощью существующей разделяемой области SQL. Если же разделяемая область SQL не найдена, будет выделена новая разделяемая область SQL для синтаксического разбора и обработки оператора.
6. Серверный процесс извлекает все необходимые значения данных из файла фактических данных или из значений, хранящихся в SGA.
7. Серверный процесс изменяет данные в SGA. Поскольку транзакция была подтверждена, процесс LogWriter (LGWR) немедленно записывает ее в файл журнала повторов. Процесс Database Writer (DBWn) окончательно записывает измененные блоки на диск, когда это является целесообразным.
8. Если транзакция была обработана успешно, процесс сервера посылает по сети соответствующее сообщение приложению. В противном случае будет передано сообщение об ошибке.
9. На протяжении всей процедуры выполняются другие фоновые процессы, отслеживая условия, которые требуют вмешательства. Кроме того, сервер базы данных управляет транзакциями других пользователей и предотвращает конкуренцию между транзакциями за одни и те же данные.

3. Структура памяти БД Oracle.

База данных Oracle создает и использует структуры памяти для различных целей.

С экземпляром связаны две базовые структуры памяти:

- *Глобальная системная область (SGA)*: группа разделяемых структур памяти (известных как компоненты SGA), которые содержат данные и управляющую информацию для одного экземпляра базы данных Oracle. SGA является общей для всех фоновых и серверных процессов. Примерами данных, хранимых в SGA, являются кэшированные блоки данных и разделяемые области SQL.
- *Глобальная программная область (PGA)*: Области памяти, которые содержат данные и управляющую информацию для фонового или серверного процесса. PGA – это неразделяемая область памяти, создаваемая базой данных Oracle при запуске фонового или серверного процесса. К PGA может обращаться только серверный процесс. Каждый фоновый или серверный процесс имеет свою PGA.

SGA – это область памяти, которая содержит данные и управляющую информацию для экземпляра.

В SGA входят следующие структуры данных:

- Буферный кэш базы данных: кэширует блоки данных, извлеченные из базы данных (Если нужного блока данных нет в кэше, он читается с диска и помещается в кэш.; совместно используется всеми параллельно работающими пользователями.; управляется алгоритмом, основанным на LRU.)

- Буфер журнала повторов: кэширует данные для операций повторов (используемых для восстановления экземпляра), пока они не будут записаны в физические файлы журнала повторов, хранимые на диске(реду лог буффер. Фоновый процесс LGWR производит запись буфера журнала повторов на диск.)

- Разделяемый пул: кэширует различные построения, которые можно разделять между пользователями (Данные вытесняются из пула по алгоритму LRU.; Серверный процесс проверяет разделяемый пул на предмет наличия разделяемой области SQL для идентичного оператора.; Серверный процесс выделяет частную область SQL по запросу сеанса.; В некоторых случаях разделяемая область SQL сбрасывается целиком: ALTER SYSTEM FLUSH SHARED_POOL;)

- Большой пул: необязательная область, позволяющая выделять большие объемы памяти для определенных ресурсоемких процессов, таких как операции резервного копирования и восстановления, а также серверные процессы ввода/вывода (Выделяется вручную администратором БД. В отличие от разделяемого пула, нет автоматического освобождения памяти по LRU.; Может быть использован: • Для операций передачи большого объема данных между разными БД. • Для операций резервного копирования / восстановления. • Размер задаётся параметром инициализации LARGE_POOL_SIZE.)

- Пул Java: используется для всего сеансового Java-кода и данных в виртуальной машине Java (JVM)

- Пул потоков: используется в Oracle Streams для хранения данных, необходимых для операций захвата и применения

База данных Oracle использует параметры инициализации для создания и управления структурами памяти. Самый простой способ управления памятью – позволить базе данных автоматически осуществлять управление и настройку памяти. Для этого необходимо установить параметр инициализации планируемого объема памяти (MEMORY_TARGET) и параметр инициализации максимального объема памяти (MEMORY_MAX_TARGET).

Структуры памяти базы данных Oracle



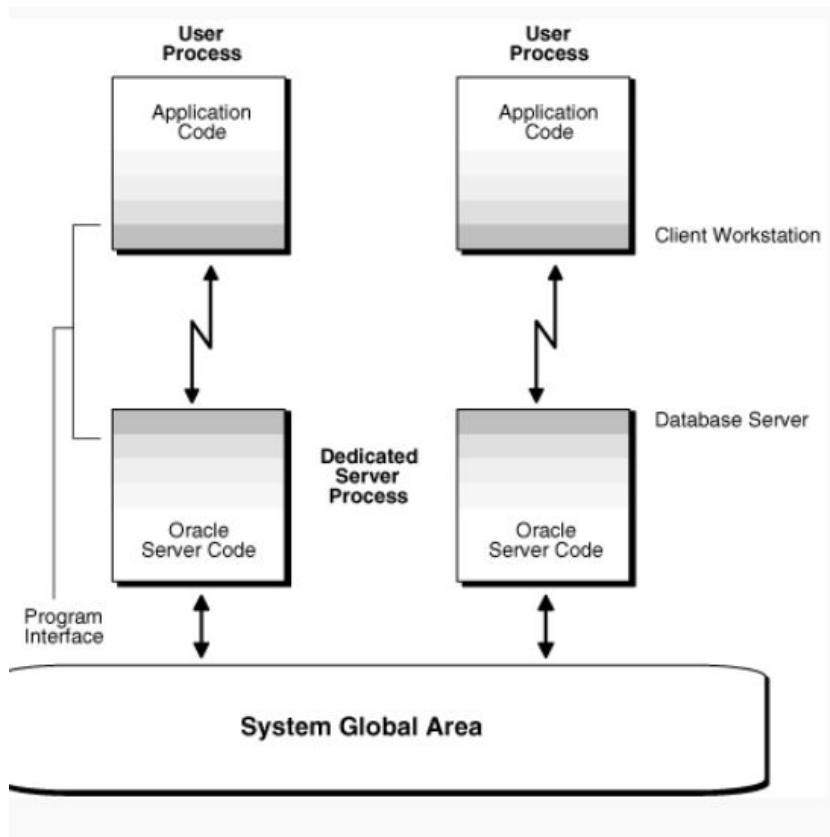
4. Архитектура процессов.

2 вида процессов:

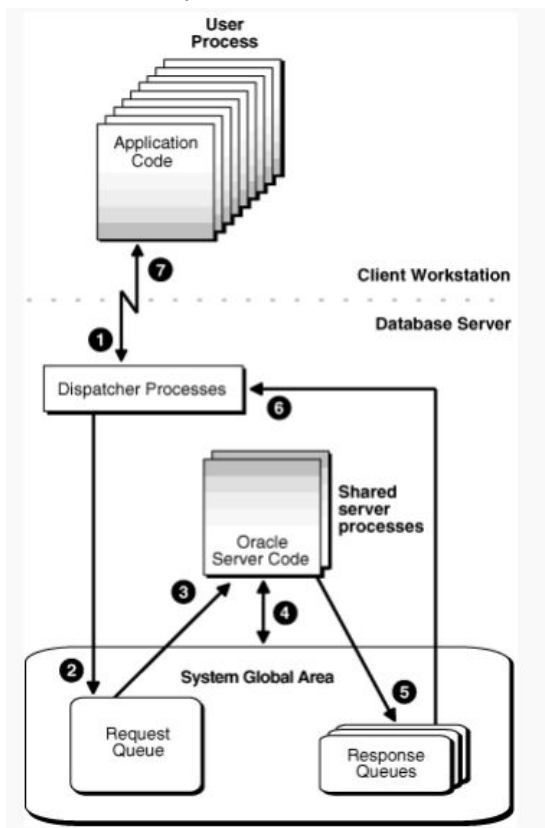
- Пользовательские процессы. Запускаются в момент подключения пользователя к БД. выполняют код приложения или инструмента Oracle
- Процессы базы данных (выполняют код сервера баз данных Oracle (включая серверные и фоновые процессы)):
 - Серверный процесс: подключается к экземпляру Oracle и запускается при установлении сеанса пользователем. Функции: выполнять синтаксический разбор и обрабатывать SQL-операторы, переданные через приложение, считывать нужные блоки данных из файлов данных на диске в разделяемые буферы базы данных SGA, возвращать результаты в такой форме, в которой приложение сможет обрабатывать данные
 - Фоновые процессы: запускаются при запуске экземпляра Oracle. Для получения дополнительных сведений о фоновых процессах см. представление V\$BGPROCESS. Примеры: Database writer – DBWn, Log writer – LGWR, Checkpoint – CKPT, System Monitor – SMON, Process monitor – PMON, Recoverer – RECO...

Для подключенных пользователей можно настроить как выделенный или разделяемый сервер.

- *Выделенный сервер*: для каждого пользователя приложение базы данных запускается пользовательским процессом, который обслуживается выделенным серверным процессом, исполняющим код сервера баз данных Oracle.



• *Разделяемый сервер*: устраняет необходимость в выделенном серверном процессе для каждого подключения. Диспетчер направляет множество входящих сетевых запросов на установку сеансов пулу процессов разделяемого сервера. Любой запрос клиента обслуживается процессом разделяемого сервера.



5. Запись данных в БД, логика работы процесса записи.

Процесс DBWn Записывает измененные (заполненные) буферы из буферного кэша базы данных на диск:

- асинхронно во время выполнения другой обработки;
- периодически для перехода к следующей контрольной точке.

Администратор может сконфигурировать СУБД на использование до 20 параллельных процессов DBW с помощью параметра инициализации DB_WRITER_PROCESSES. (бесполезно в однопроцессорных системах)

Если буфер в буферном кэше базы данных изменяется, он помечается как заполненный и добавляется в список LRUW заполненных буферов, отсортированный по номеру SCN. Таким образом, порядок буферов совпадает с порядком операций повтора, записанных в журналы повторов для таких измененных буферов. Когда количество доступных буферов в буферном кэше становится меньше внутреннего порогового значения, DBWn записывает заполненные буферы в файлы данных в порядке их изменения, следуя порядку в списке LRUW и одновременно «перемещает» указатель на контрольную точку в структуре памяти SGA, с которой будет начато восстановление в случае сбоя (инкрементальная установка контрольных точек)

Процесс DBWn записывает заполненные буферы на диск при таких условиях:

- Когда серверному процессу после сканирования порогового количества буферов не удастся найти свободный многократно используемый буфер, он подает процессу DBWn сигнал начинать запись. Процесс DBWn записывает заполненные буферы на диск в асинхронном режиме во время выполнения других вычислений.
- Процесс DBWn периодически записывает буферы, чтобы передвинуть контрольную точку, которая является позицией в потоке (журнале) повторов и с которой начинается восстановление экземпляра. Эта позиция определяется наиболее старым заполненным буфером в буферном кэше.

Во всех случаях процесс DBWn осуществляет пакетную (многоблочную) запись, что повышает продуктивность. Количество блоков, записываемых при многоблочной записи, зависит от используемой операционной системы.

6. Запись журнала повторов, логика работы процесса записи.

Процесс LGWR записывает буфер журнала повторов в файл журнала повторов на диске. Запись осуществляется:

- когда пользовательский процесс фиксирует транзакцию;
- когда буфер журнала повторов заполняется на одну треть;
- перед тем как процесс DBWn записывает измененные буферы на диск. После того, как данные из буфера журнала повторов записаны на диск, серверные процессы могут записать на их место новые данные.

- Данные в файл журнала повторов записываются сразу же после того, как пользователь вызвал оператор COMMIT. Т.е., данные в журнал повторов обычно записываются раньше, чем в файлы данных.
- Когда пользователь выдает оператор COMMIT, процесс LGWR помещает запись фиксации транзакции в буфер журнала повторов. После этого LGWR немедленно записывает на диск буфер журнала повторов и записи повторов из транзакции. Внесение соответствующих изменений в блоки данных задерживаются до момента наиболее эффективной записи. Это называется механизмом быстрой фиксации транзакции

Буфер журнала повторов является циклическим.

После того как процесс LGWR сохранит записи повторов из буфера в файл журнала повторов, серверные процессы смогут копировать новые записи на место записей в буфере журнала повторов, которые уже записаны на диск.

7. Создание контрольной точки, процесс архивирования журнала повторов.

Контрольная точка – это структура данных, определяющая системный номер изменения (SCN) в потоке повторов базы данных. Контрольные точки записываются в управляющий файл и в заголовок каждого из файлов данных. Они являются важнейшим элементом процесса восстановления.

При фиксации контрольной точки Oracle Database нужно обновить заголовки всех файлов данных, чтобы записать сведения о контрольной точке. Эту операцию выполняет процесс СКПТ. Процесс СКПТ не записывает блоки на диск, эту операцию выполняет процесс DBWn. Запись номеров SCN в заголовки файлов гарантирует, что все изменения, внесенные в блоки базы данных до фиксации данного номера SCN, уже записаны на диск.

Статические контрольные точки DBWR, которые отображает в Oracle Enterprise Manager монитор SYSTEM_STATISTICS, показывают количество выполненных запросов создания контрольной точки.

Процессы архиваторов (ARCn) копируют файлы журнала повторов на указанное устройство хранения после переключения журнала. Процессы ARCn функционируют, только если БД работает в режиме ARCHIVELOG и при этом включено автоматическое архивирование. Если ожидается, что архивирование вызовет интенсивную загрузку (например, при пакетной загрузке данных), можно увеличить максимальное количество процессов архиваторов при помощи параметра инициализации LOG_ARCHIVE_MAX_PROCESSES. Оператор ALTER SYSTEM позволяет изменять значение данного параметра динамически, увеличивая или уменьшая количество процессов ARCn.

8. Установка БД. Основные задачи администратора при установке в среде Unix-подобных систем.

Перед установкой задать значения ряду переменных окружения:

- ORACLE_BASE. Устанавливает путь к «корню» иерархии каталогов Oracle.

- ORACLE_HOME. Устанавливает путь к «корневому» каталогу БД. Этот путь свой для каждого экземпляра БД.
- ORACLE_SID. Задаёт имя экземпляра Oracle. Значение по умолчанию - ORCL
- NLS_LANG. Устанавливает язык и кодировку БД.

Установить БД:

- С помощью Oracle Universal Installer — в интерактивном режиме с помощью графической утилиты: `./runInstaller`
- «Silent Mode» — с помощью файла конфигурации, (Response File) заданного в ходе одной из предыдущих установок:
`./runInstaller -record -responseFile ./runInstaller -silent -responseFile responsefilename`

После завершения работы OUI необходимо выполнить ряд скриптов из-под суперпользователя:

```
# cd /u01/app/oracle/orainventory
# ./orainstRoot.sh
# cd /u01/app/oracle/product/11.1.0/db_1
# ./root.sh
```

При установке в ОС Linux или UNIX принять настройки по умолчанию для локального каталога bin. После завершения сценариев выйти из всех связанных с ними учетных записей и окон, чтобы завершить установку.

9. Архитектура хранения базы данных. Подход OFA.

OFA – метод конфигурирования БД. OFA использует преимущества ОС и дисковых подсистем и позволяет создать простую для администрирования конфигурацию, которая обеспечит максимальную гибкость растущих и высокопроизводительных баз данных.

OFA предназначена для:

- организации большого количества сложного ПО и данных на диске для избежания узких мест на устройствах и снижения производительности;
- упрощения повседневных задач администрирования (например, резервного копирования ПО и данных), которые нередко уязвимы к повреждению данных;
- упрощения переключения между несколькими БД Oracle;
- контроля и управления ростом базы данных;
- снижения фрагментации свободного пространства в словаре данных, изоляции прочих фрагментированных областей и сведения конкуренции за ресурсы к минимуму.

Оптимальная гибкая архитектура: схема именования

- Точки монтирования:
 - /u01
 - /disk01
- Каталоги:
 - /u01/app/oracle
 - /u01/app/applmgr
- Файлы:
 - Управляющие файлы: `controln.ctl`
 - Файлы журнала повторов: `redon.log`
 - Файлы данных: `tn.dbf`



10. Параметры инициализации экземпляра БД. Файлы параметров.

- Считываются экземпляром Oracle при его запуске.
- Существует 2 типа файлов параметров:
 - Файлы параметров сервера (SPFILE) — двоичный файл, чтение и запись в который осуществляет сервер БД. Самостоятельно изменять этот файл нельзя. Имя по умолчанию — `spfile.ora`.
 - Текстовый файл параметров инициализации — может быть только считан сервером, но не записан. Настройки параметров инициализации необходимо задавать и изменять вручную (с помощью текстового редактора). Имя по умолчанию — `init.ora`. При наличии SPFILE этот файл игнорируется.

Существует два типа файлов параметров инициализации:

- Статические параметры — могут быть изменены только путем редактирования файлов `init.ora` и `SPFILE`. Чтобы изменения вступили в силу, необходимо остановить и перезапустить БД.
- Динамические параметры — могут изменяться при работающей БД. Существует два типа динамических параметров:
 - Параметры уровня сеанса — оказывают влияние только на текущий сеанс. Пример — параметры поддержки национальных языков (NLS).
 - Параметры уровня системы — оказывают влияние на всю базу данных и все сеансы. Пример — параметры, отвечающие за сброс данных разделяемого пула или параметры расположения архивного журнала. Действие данных параметров зависит от настройки `SCOPE.x`.

Динамические параметры можно изменять с помощью команд `ALTER SESSION` и `ALTER SYSTEM`:
`SQL> ALTER SESSION SET NLS_DATE_FORMAT='mon dd yyyy';`

Некоторые дополнительные параметры инициализации БД:

- `DB_FILES` — максимальное количество файлов БД.

- PROCESSES — максимальное количество параллельных пользовательских процессов.
- DB_BLOCK_SIZE — размер блока данных БД (в байтах; по умолчанию — 8 КБ).
- DB_CACHE_SIZE — размер блока буферного кэша БД (в байтах; по умолчанию — 48 МБ для однопроцессорной системы).
- SGA_TARGET — общий размер SGA (в байтах).

11. Запуск и остановка экземпляра БД. Режимы запуска и остановки.

а) Запуск осуществляется командой startup.

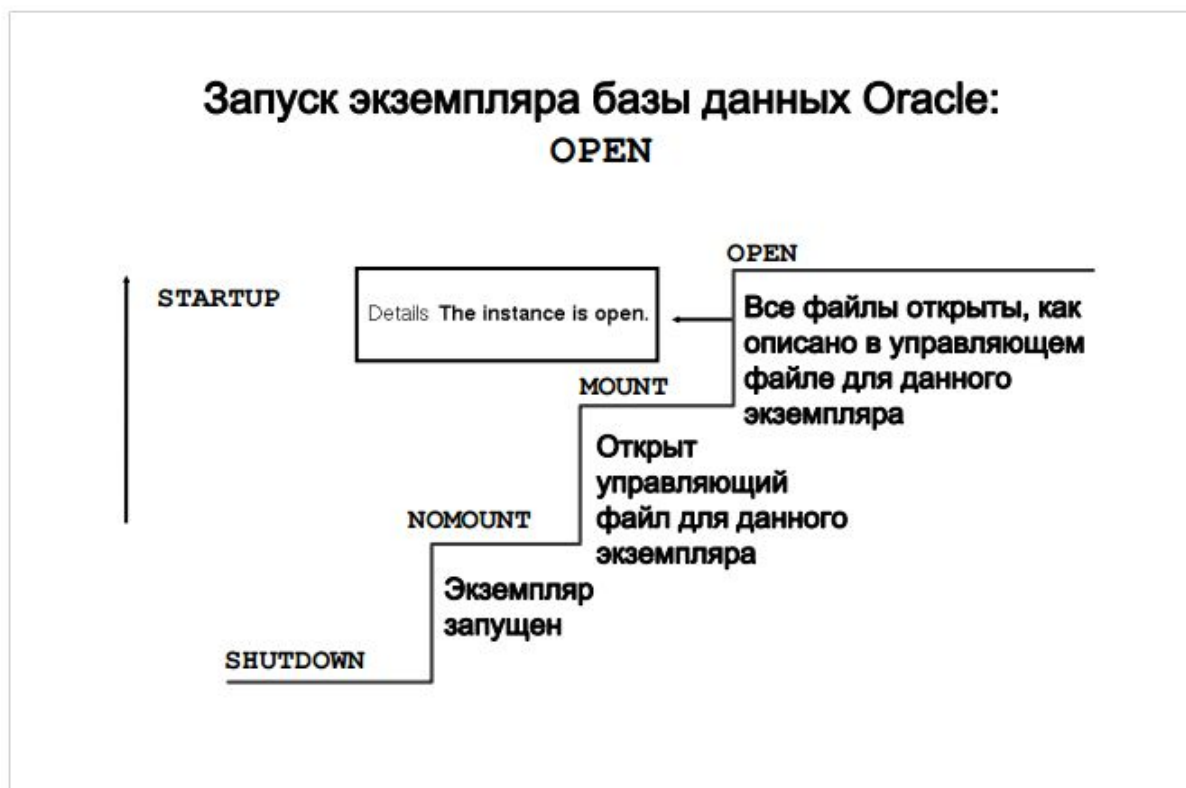
1) nomount - при запуске экземпляра выполняются следующие действия: поиск файлов параметров, выделение SGA, запуск фоновых процессов, открытие файла alert_.log и файлов трассировки

2) mount - при монтировании базы данных выполняются следующие операции:

- база данных ассоциируется с ранее запущенным экземпляром
- выполняется поиск и открытие управляющих файлов, указанных в файле параметров
- производится чтение управляющих файлов для получения сведений об именах и состоянии файлов данных и интерактивных файлах журналов повтора.

3) open - при открытии базы данных выполняются такие операции:

- открытие файлов данных
- открытие оперативных файлов журнала повторов



б) Остановка осуществляется командой shutdown.

- ABORT: выполняется минимальное количество операций до остановки. Так как этот режим требует выполнения восстановления перед запуском, используйте его только

при необходимости. Обычно используется в ситуациях, когда не срабатывает ни один другой режим остановки

- IMMEDIATE: наиболее часто используемый режим. Для всех неподтвержденных транзакций выполняется откат.
- TRANSACTIONAL: позволяет завершить транзакции.
- NORMAL: ожидает завершения сеансов.

Режимы остановки

| Режим остановки | A | I | T | N |
|---|-----|-----|-----|-----|
| Разрешены новые подключения | Нет | Нет | Нет | Нет |
| Ожидание завершения текущего сеанса | Нет | Нет | Нет | Да |
| Ожидание завершения текущей транзакции | Нет | Нет | Да | Да |
| Принудительное создание контрольной точки и закрытие файлов | Нет | Да | Да | Да |

Режимы остановки:

- A = ABORT
- I = IMMEDIATE
- T = TRANSACTIONAL
- N = NORMAL

12. Словарь данных и динамические представления V\$ и GV\$.

а) Словарь данных Oracle является описанием базы данных и содержит имена и атрибуты всех объектов базы данных. Все операции создания или изменения каких-либо объектов записываются в словарь данных для отражения изменений. Данные сведения хранятся в базовых таблицах, которые обслуживаются базой данных Oracle. Обращаться к данным таблицам следует через заранее определенные представления, а не напрямую.

Сведения словаря данных:

- информация о клиентах, объектах, ограничениях и хранилищах
- обновляются при обновлении объектных структур или описаний
- доступны любому пользователю для получения сведений о базе данных
- принадлежат пользователю SYS
- не должны изменяться непосредственно с помощью SQL

Глобальные представления с полными сведениями обо всем доступны только пользователям с полномочиями DBA - префикс DBA_. Следующий уровень полномочий - префикс ALL_(все объекты, которые может увидеть пользователь с заданными полномочиями независимо от того, является ли он владельцем данных

объектов). Префикс USER_ отражает только те объекты, которыми владеет пользователь.

б) База данных Oracle поддерживает более динамические наборы данных, которые отражают функционирование и производительность экземпляра базы данных. Некоторые из них могут отображать данные до того, как база данных будет смонтирована и открыта. Динамические представления производительности содержат сведения о: сеансах, состоянии файлов, ходе выполнения заданий и задач, блокировках, состоянии резервного копирования, использовании и выделении памяти, параметрах системы и сеансов, выполнении SQL, статистических данных и метриках. Префикс V\$ - представления производительности экземпляра БД.

Префикс GV\$ - глобальные представления узлов кластера RAC.

V\$SYSTEM_EVENT — содержит общесистемную информацию о ресурсах, которых ждет весь экземпляр.

V\$SESSION_EVENT — список событий, которые приходилось ждать в каждом сеансе.

V\$SESSION — информация о сеансе, в том числе о событии, которое ожидает сеанс в данный момент или ждал в последний раз

13. Методы разрешения имен, настройка псевдонимов.

Oracle Net поддерживает следующие методы разрешения имен:

- Разрешение имен Easy Connect: при данном методе клиенты могут соединяться с сервером БД Oracle с помощью строки соединения TCP/IP, которая состоит из имени хоста, после которого могут быть также указаны порт и имя службы: CONNECT <имя_пользователя>/<пароль>@<имя_хоста>:<порт прослушивателя>/<имя службы>. Данный метод разрешения имен не требует настройки.
- Локальное разрешение имен: при данном методе дескрипторы соединения (различаемые по именам сетевых служб) хранятся на стороне клиента в локальном файле конфигурации tnsnames.ora.
- Разрешение имен на сервере каталогов: для доступа к службе БД данный метод сохраняет идентификаторы соединения для обращения к службе БД на центральном сервере каталогов, который поддерживает протокол облегченного доступа к каталогам (LDAP);
- Внешнее разрешение имен: при данном методе имена служб хранятся в поддерживаемой сторонней службе разрешения имен. Поддерживаемые сторонние службы: - Служба внешнего разрешения имен Network Information Service (NIS); - Distributed Computing Environment (DCE) Cell Directory Services (CDS).

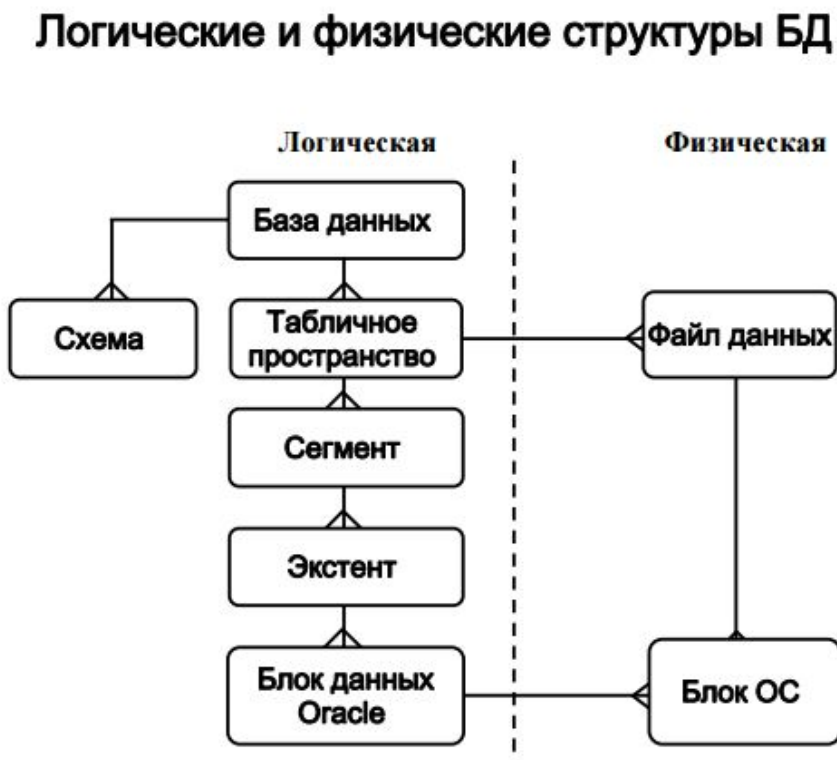
14. Выделенный и разделяемый режим работы сервера. Преимущества и недостатки.

а) При использовании выделенных серверных процессов соотношение пользовательских и серверных процессов равно один к одному. Каждый серверный процесс использует системные ресурсы, включая процессорное время и память. В системах с интенсивной нагрузкой выделение ресурсов памяти и процессора выделенным серверным процессам может оказаться недопустимым и негативно отразиться на масштабируемости таких систем. Если требования к ресурсам, негативно влияют на систему, можно выполнить следующие действия: увеличить объем системных ресурсов, увеличив объем памяти и мощность ЦП.

б) Для каждой службы в архитектуре разделяемого сервера используется как минимум один процесс диспетчера (обычно больше). При поступлении запроса соединения, он перенаправляется к наименее загруженному диспетчеру, обслуживающему службу с запрошенным именем. В течение сеанса поддерживается соединение с одним и тем же диспетчером. В отличие от выделенных серверных процессов, один диспетчер может обслуживать сотни сеансов пользователей.

На самом деле диспетчеры направляют запросы пользователей в общую очередь, которая размещена в области SGA, выделенной для разделяемого пула. Процессы разделяемого сервера выполняют большую часть операций выделенных серверных процессов. Они выбирают запросы из очереди и обрабатывают их. Поскольку запросы одного пользовательского процесса могут обрабатывать несколько процессов разделяемого сервера, большинство структур хранения, которые обычно находятся в PGA, следует разместить в области совместно используемой памяти (по умолчанию это разделяемый пул). Тем не менее, если настроен большой пул или параметр SGA_TARGET настроен для автоматического управления памятью, такие структуры хранения размещаются в области SGA, выделенной для большого пула.

15. Логические и физические структуры хранения. Представление табличных данных, блок базы данных.



Каждая база данных логически состоит из одного или нескольких табличных пространств. Для каждого табличного пространства явным образом создается один или несколько файлов данных, которых физически хранятся данные.

Схема – это набор объектов БД, принадлежащих пользователю БД. Объектами схемы - таблицы, представления, последовательности, хранимые процедуры, синонимы, индексы, кластеры и каналы связи с базой данных.

На низшем уровне логической структуры находятся блоки данных, в которых хранятся данные БД Oracle.

Экстент представляет собой определенное количество непрерывных блоков данных (полученных при одном размещении).

Сегмент представляет собой набор экстентов, выделенных для определенной логической структуры.

Сегменты бывают нескольких типов:

- сегменты данных: данные всех таблиц в кластере хранятся в сегменте данных кластера.
- индексные сегменты: любой индекс использует индексный сегмент, в котором хранятся все данные индекса.
- сегменты отмены: для каждого экземпляра БД создается одно табличное пространство отмены транзакций (UNDO).
- временные сегменты: Временные сегменты создаются базой данных Oracle, когда для выполнения оператора SQL необходима временная рабочая область.

Табличные данные

При создании таблицы создается сегмент для хранения ее данных. Табличное пространство содержит набор сегментов. С логической точки зрения таблица содержит строки, заполненные значениями столбцов. Строка хранится в блоке базы данных в виде неформатированной строки. Она называется неформатированной строкой потому, что ее фрагменты могут храниться в разных местах, так как вставленная строка может быть слишком большой, чтобы поместиться в одном блоке, или же в результате операции обновления существующая строка перестает помещаться в старое место.

Блок базы данных

- Заголовок блока: заголовок блока содержит сведения о типе сегмента (табличный или индексный), адресе блока данных, каталоге таблицы, каталоге строки и транзакционных слотах размером по 24 байта каждый, которые используются при внесении изменений в строки блока. Заголовок блока заполняется сверху вниз.
- Неформатированные данные: это данные для строк блока. Пространства неформатированных данных заполняются снизу вверх.
- Свободное пространство: свободное пространство внутри блока, за счет которого могут увеличиваться заголовок и пространство неформатированных данных.

16. Табличные пространства и файлы данных. Управление табличными пространствами.

База данных делится на табличные пространства, являющиеся логическими блоками хранения. Их можно использовать для объединения связанных между собой логических структур. Каждая база данных логически состоит из одного или более табличных пространств. Для каждого табличного пространства явным образом создается один или несколько файлов данных. В них физически хранятся данные всех логических структур табличного пространства.

Любая база данных Oracle должна содержать табличные пространства SYSTEM и SYSAUX, которые автоматически создаются вместе с базой данных. По умолчанию создаются табличные пространства типа SMALLFILE. Можно также создавать табличные пространства типа BIGFILE, что позволит БД Oracle оперировать

очень большими файлами (до 8 эксабайт). Табличное пространство может находиться в оперативном (доступном) и автономном (недоступном) режимах.

Табличное пространство SYSTEM всегда находится в оперативном режиме, пока база данных открыта. В нем хранятся таблицы, обеспечивающие основные функции базы данных, например, таблицы словаря данных.

Табличное пространство SYSAUX является вспомогательным. В нем хранится множество компонентов базы данных, поэтому оно должно находиться в оперативном режиме, чтобы обеспечивать правильную работу всех компонентов базы данных.

Для табличных пространств выделяется пространство в экстентах. Для отслеживания свободного и используемого пространства при создании табличного пространства можно использовать один из двух методов:

- Управляемые локально табличные пространства: управление экстентами в табличном пространстве осуществляется с помощью битовых образов. Каждый бит битового образа соответствует блоку или группе блоков. При выделении или освобождении экстента для повторного использования сервер Oracle изменяет значения битового образа для отражения нового состояния блоков.

- Управляемые словарем табличные пространства: экстенты управляются словарем данных. При операциях выделения и освобождения экстенентов сервер Oracle обновляет соответствующие таблицы в словаре базы данных. Это сделано для обеспечения обратной совместимости. Рекомендуется использовать управляемые локально табличные пространства.

17. Пользователи БД, учетные записи пользователей. Системные учетные записи.

Формуляр пользователя БД – это способ организации принадлежности и доступа к объектам БД.

Для доступа к БД пользователю необходимо указать соответствующий формуляр пользователя БД и успешно пройти аутентификацию для данного формуляра, это позволяет избежать потенциальных брешей в безопасности и предоставить подробные данные для определенных операций аудита. У каждого формуляра пользователя есть:

- Уникальное имя пользователя: не может содержать специальные символы и должно начинаться с буквы;
- Метод аутентификации: наиболее распространенным методом аутентификации является использование пароля, однако в Oracle Database 11g поддерживается несколько других методов аутентификации, включая биометрическую, аутентификацию с использованием сертификатов или маркеров.
- Табличное пространство по умолчанию: в этой области пользователь создает объекты, если не было указано другое табличное пространство.
- Временное табличное пространство: это область для хранения временных объектов, например временных таблиц, которые создаются экземпляром от имени пользователя.
- Профиль пользователя: набор ресурсов и ограничений с помощью паролей, присвоенных пользователю.
- Исходная группа потребителей: используется диспетчером ресурсов.

- Состояние формуляра: пользователям доступны только «открытые» формуляры. Параметр `account_status` может иметь различные значения, сочетающие «locked» (заблокированный) и «expired» (просроченный).

Формулярам SYS и SYSTEM по умолчанию предоставляется роль администратора БД.

Кроме того, формуляр SYS обладает всеми полномочиями с параметром ADMIN OPTION и является владельцем словаря данных. Для подключения к формуляру SYS необходимо использовать фразу AS SYSDBA для экземпляра БД и фразу AS SYSASM для экземпляра автоматического управления хранением (ASM). Пользователь с полномочиями SYSDBA может подключиться к формуляру SYS, воспользовавшись фразой AS SYSDBA. Только «уполномоченным» пользователям с полномочиями SYSDBA, SYSOPER или SYSASM разрешается запускать и останавливать экземпляры. Формуляру SYSTEM по умолчанию предоставляется роль DBA, но не предоставляются полномочия SYSDBA.

18. Системные и объектные полномочия. Назначение и удаление полномочий.

Полномочие – это право на выполнение определенного типа SQL-оператора или на доступ к объекту пользователя. С помощью БД Oracle можно контролировать списки операций в базе данных, которые разрешены и запрещены пользователю. Полномочия делятся на две категории:

- *Системные полномочия:* позволяют пользователю выполнять определенную операцию в базе данных. Системные полномочия предоставляются администратором или пользователем, который владеет явно заданным разрешением на управление полномочиями.

- SYSDBA и SYSOPER. Эти полномочия позволяют остановить, запустить, восстановить, а также выполнить любые другие административные задачи в базе данных.

- SYSASM- запуск, остановка экземпляра ASM и управление им.

- DROP ANY объект, удаление объекты других пользователей схемы.

- CREATE, MANAGE, DROP и ALTER TABLESPACE.

- CREATE LIBRARY - создавать и вызывать внешний код (например библиотеку C).

- CREATE ANY DIRECTORY - позволяет создать объект каталога (с доступом на чтение и запись) для любого каталога, доступного владельцу ПО Oracle. Это означает, что пользователь с таким полномочием имеет доступ к внешним процедурам этих каталогов.

- GRANT ANY OBJECT PRIVILEGE. Это полномочие позволяет предоставлять разрешения для объектов, которыми его обладатель не владеет.

- ALTER DATABASE и ALTER SYSTEM. Они позволяют изменять базу данных и экземпляр Oracle (например, можно переименовать файл данных или очистить кэш буфера).

- *Объектные полномочия:* объектное полномочие дают пользователю возможность выполнять определенную операцию над определенным объектом, например таблицей, представлением, последовательностью, процедурой, функцией или пакетом. Без специального разрешения пользователю доступны только собственные объекты. Объектные полномочия может предоставить владелец объекта,

администратор или пользователь, которому явным образом задано разрешение на предоставление таких полномочий.

- SELECT — возможность выполнять выборку данных из таблицы.
- INSERT — возможность добавлять новые строки в таблицу.
- UPDATE — возможность изменять данные в таблице.
- DELETE — возможность удалять строки из таблицы.
- REFERENCES — возможность создавать ограничения целостности для выбранной таблицы.
- ALTER — возможность выполнять оператор ALTER TABLE применительно к выбранной таблице.
- INDEX — возможность создавать индексы на столбцы выбранной таблицы.
- ALL — все перечисленные выше полномочия.

Присвоение и аннулирование:

С помощью SQL-операторов GRANT и REVOKE:

GRANT privileges ON object TO user;

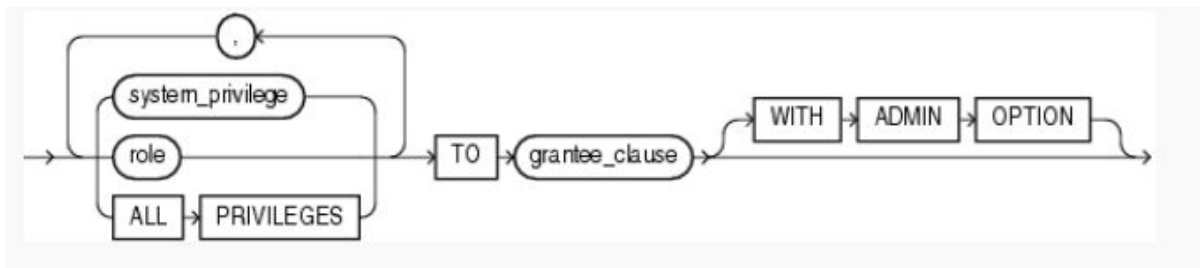
REVOKE privileges ON object FROM user;

Например:

SQL> GRANT SELECT, INSERT, UPDATE ON suppliers TO smithj;

SQL> REVOKE ALL ON suppliers FROM anderson;

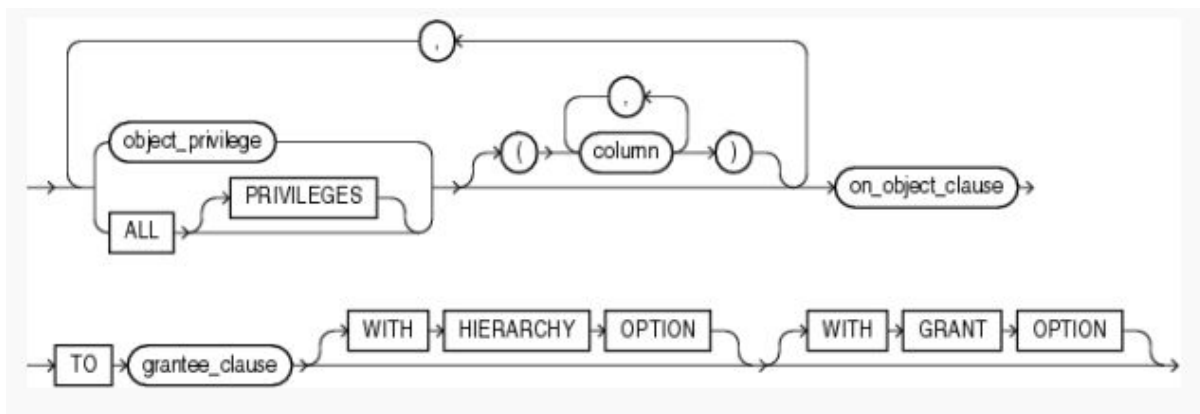
ADMIN OPTION:



Позволяет:

- Выдать эти системные полномочия другому формуляру или роли.
- Аннулировать эти полномочия у другого формуляра или роли.

GRANT OPTION:

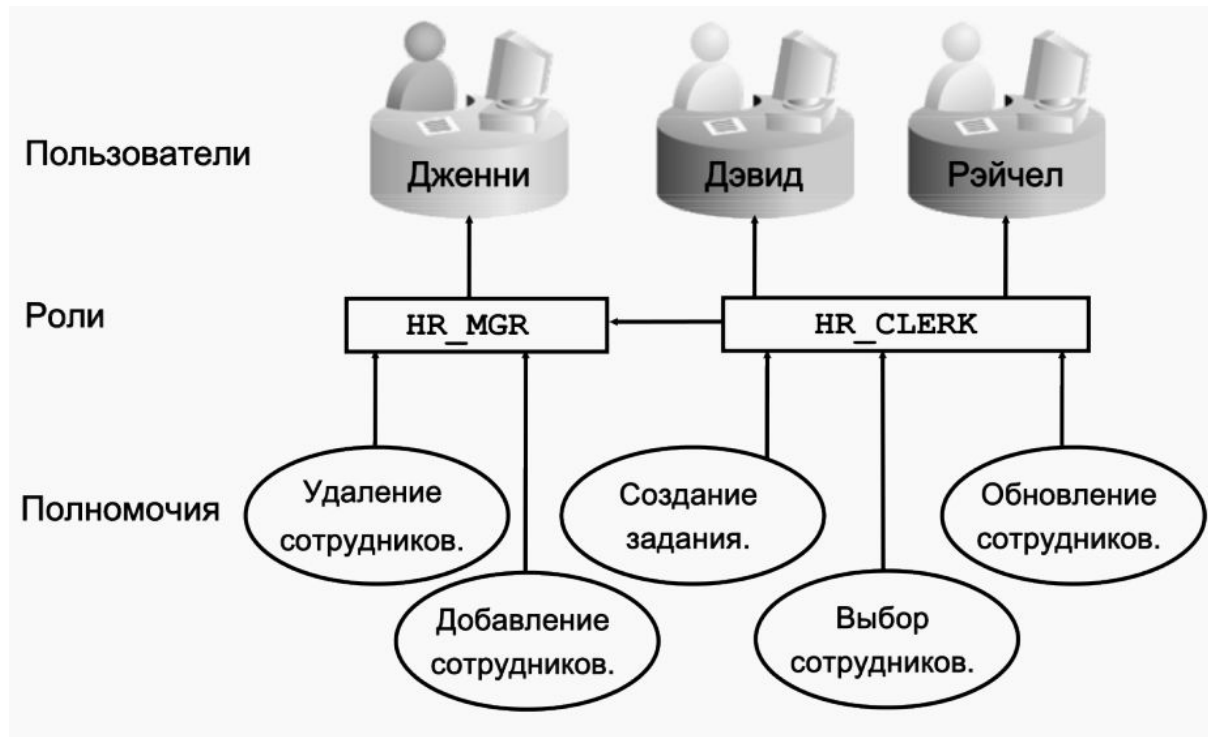


Позволяет:

- Выдать эти объектные полномочия другому формуляру или роли

19. Роли БД. Предопределенные роли. Профили пользователей.

Роль – это именованная группа связанных полномочий, которая предоставляется пользователям или другим ролям. С помощью ролей администратор БД управляет полномочиями.



Преимущества использования

- Упрощенное управление полномочиями. Вместо того чтобы предоставлять нескольким пользователям один и тот же набор полномочий, можно предоставить эти полномочия роли, а затем предоставить эту роль пользователям.
- Динамическое управление полномочиями. Если полномочия, связанные с ролью, изменяются, все пользователи, которым предоставлена эта роль, немедленно автоматически получают обновленные полномочия.
- Выбор доступных полномочий. Роли можно включать и отключать, чтобы временно активировать или деактивировать полномочия. Таким образом можно контролировать полномочия пользователя в той или иной ситуации.

Предопределенные роли

| | |
|----------------------|---|
| CONNECT | CREATE SESSION |
| RESOURCE | CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE |
| SCHEDULER_ ADMIN | CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER |
| DBA | Большая часть системных полномочий; несколько других ролей. Следует предоставлять только администраторам. |
| SELECT_ CATALOG_ROLE | Системных полномочий нет; HS_ADMIN_ROLE и более 1700 объектных полномочий в словаре данных |

Профили представляют собой именованные наборы ограничений на использование ресурсов БД и экземпляра. Также профили регулируют состояние формуляров и пароли пользователей (количество символов, срок действия и т.д.).

Каждому пользователю назначается профиль, одновременно только один. С помощью профилей администратор может контролировать следующие системные ресурсы:

- Процессор: ресурсы процессора можно ограничивать на основе сеансов или вызовов.
- Сеть/память: можно задать следующие параметры:
 - Время соединения: интервал времени в минутах, в течение которого пользователь может быть соединен до того, как он будет автоматически отключен;
 - Время простоя: указывает интервал времени в минутах, в течение которого сеанс пользователя может бездействовать до того, как он будет автоматически отключен.
- Дисковый ввод/вывод: ограничивает объем данных, которые пользователь может считать за сеанс или за вызов.

20. Целостность, ограничения, состояние ограничений.

Для недопущения введения в столбцы тех или иных значений можно использовать следующие ограничения:

- NOT NULL.
- Ключ UNIQUE.
- PRIMARY KEY.

Реализация ограничения целостности PRIMARY KEY в сервере Oracle обеспечивает выполнение двух следующих условий:

- Ни одна пара строк таблицы не содержит одинаковых значений в заданном столбце или группе столбцов.

- Столбец первичных ключей не может содержать неопределенных значений.

- Ссылочные ограничения целостности: разные таблицы в реляционной базе данных могут быть связаны общими столбцами, поэтому необходимо соблюдать правила отношения столбцов. Ссылочные правила целостности гарантируют сохранение этих отношений. Ссылочные ограничения целостности накладывают следующее

требование: для всех строк таблицы значение во внешнем ключе должно совпадать со значением в первичном ключе. При этом внешний ключ ссылается на родительский ключ в той же таблице.

- Ограничения CHECK: ограничение целостности CHECK для столбца или набора столбцов указывает, что для каждой строки таблицы должно выполняться или иметь неизвестное значение какое-либо заданное условие.

Нарушение ограничения происходит, если переданный DML-оператор не совместим с ограничением. Нарушения ограничения могут обуславливаться разными условиями. Вот некоторые из них:

- Неуникальность: попытка хранения одинаковых значений в столбце с ограничением UNIQUE (например, если столбец является первичным ключом или используется уникальный индекс);
- Целостность ссылочных данных: нарушение правила, согласно которому у каждой дочерней строки должна быть родительская строка;
- Проверка: попытка сохранить в столбце значение, противоречащее правилам, определенным для столбца.

Состояния ограничений

Чтобы проще разрешать ситуации, когда данные временно могут нарушать ограничение, можно использовать различные состояния ограничений. Ограничение целостности можно включить (ENABLE) или выключить (DISABLE). Если ограничение включено, проверка данных происходит при вводе данных или обновлении БД.

Данные, противоречащие правилам ограничения, не вводятся. Если ограничение отключено, в базу данных могут вводиться противоречащие ограничению данные.

Ограничение целостности может находиться в одном из следующих состояний:

- DISABLE NOVALIDATE: новые или уже существующие данные могут противоречить ограничению, если они не проверены.
- DISABLE VALIDATE: при данном состоянии ограничения изменение столбцов, на которых оно наложено, запрещено, так как это нарушит целостность данных – уже существующие данные будут проверены, а новые будут вводиться без проверки.
- ENABLE NOVALIDATE: новые данные соответствуют ограничению, однако уже существующие данные находятся в неизвестном состоянии.
- ENABLE VALIDATE: ограничению не должны противоречить ни новые, ни существующие данные.

//молодой Алек

21. Понятие блокировок, уровни блокировки и операторы их использующие

Блокировки — механизм предотвращающий изменение одних и тех же данных транзакциями разных пользователей. Различают блокировки данных (DML-блокировки) и DDL-блокировки, которые блокируют изменения в структуре БД. При попытке изменения данных в строке (операторами UPDATE, INSERT, DELETE, SELECT FOR UPDATE) блокировка накладывается непосредственно на изменяемую строку, так же блокировка накладывается и на изменения в БД. Блокировки удерживаются до тех пор пока не будет выполнен COMMIT или ROLLBACK.

22. Конфликты блокировок, устранение конфликтов блокировок. Взаимные блокировки.

Конфликты блокировок возникают при попытке изменения одних и тех же данных разными транзакциями. Заблокированная транзакция продолжит свое выполнение сразу после того как транзакция заблокировавшая объект будет закоммичена или откатена. Если блокировку пытаются получить несколько транзакций они все будут в последствии выполнены в порядке запроса блокировки. Возможно ручное устранение конфликтов блокировок, путем принудительного завершения сессии в которой выполняется блокирующая транзакция. Взаимные блокировки — возникают при попытке двух транзакций заблокировать одни и те же данные в разной последовательности, Oracle умеет детектировать такие ситуации и в этом случае одна из транзакций будет завершена с ошибкой.

23. Данные отмены операций(UNDO), изменение данных отмены операций и журнала повторов операций во время транзакции

Данные отмены операций используются для различных целей в их числе: откат транзакций, восстановление БД, предоставление консистентности чтения. UNDO данные хранятся в отдельном табличном пространстве UNDO в каждый момент времени в СУБД может существовать только одно активное табличное пространство UNDO. При выполнении операций изменяющих данные данные о такой операции заносятся в журнал повторов, так же данные о предшествующем состоянии заносятся в табличное пространство UNDO, следует отметить, что старые данные в данном табличном пространстве могут вытесняться новыми, существуют настройки для управления временем хранения данных в UNDO (параметр UNDO_RETENTION)

24. Автоматическое и ручное управление основной и разделяемой памятью.

Простейший способ управления памятью экземпляра – автоматическое управление, при таком методе администратор БД может установить только параметры MEMORY_TARGET и MEMORY_MAX_TARGET, Oracle автоматически выберет размеры всех буферов и кэшей. Кроме этого существует два метода ручного управления для каждой области памяти SGA и PGA: ручной и автоматический. В ручном режиме SGA есть возможность установить размеры отдельных буферов SGA, в автоматическом же можно устанавливать размеры всего буфера SGA. В случае PGA в автоматическом режиме так же можно устанавливать размеры всего PGA, а в случае ручного управления есть возможность установки максимального размера рабочей области для каждого типа SQL оператора.

25. Ошибки пользователя БД. Область мгновенного восстановления.

Под пользовательской ошибкой понимается выполнение пользователем запроса в рамках его привелегий, приводящего к неверному состоянию базы данных с точки

зрения пользователя. Область мгновенного восстановления предназначено для удержания вспомогательные файлы: файлы резервных копий, архивные логи, flashback логи, рекомендуется к использованию для упрощения процедур резервного копирования и восстановления, так же рекомендуется размещать её на отдельных физических устройствах.

26. Ошибки экземпляра БД. Процедура и этапы восстановления экземпляра.

Под ошибками экземпляра БД понимается ситуация в которой экземпляр БД «падает», что может происходить как из-за сбоев в критичных для экземпляра процессах, так и из-за отказа оборудования или сети электричества. После устранения причины сбоя при запуске экземпляра Oracle автоматически запустит процедуру восстановления, которая состоит из двух этапов: в первом к базе будут применены необходимые записи redo из online redo log, а во втором происходит откат всех незавершенных на момент сбоя транзакций.

27. Резервное копирование. Полная и инкрементальная резервные копии.

Резервное копирование — процесс создания копии данных на случай утери основных данных. В oracle резервные копии разделяются на консистентные и неконсистентные, в первом случае все файлы данных и control файлы имеют один и тот же SCN, для создания такой резервной копии необходима остановка БД, поэтому это иногда называют offline backup, во втором же случае резервное копирование осуществляется на работающей БД (online backup), что накладывает определённые ограничения, например БД обязательно должна работать в режиме архивирования логов (ARCHIVELOG).

Разделяют полные и инкрементальные резервные копии. В полных резервных копиях содержится вся информация, находившаяся в БД на момент создания копии, инкрементальные же копии содержат лишь изменения произошедшие с момента создания последней копии.

28. Менеджер резервного копирования RMAN

RMAN – клиент базы данных Oracle используемый для создания резервных копий. Использует собственный язык для задания команд, что позволяет создавать скрипты для выполнения бэкапов.

29. Data pump — архитектура. Утилиты export и import.

Data Pump – технология для загрузки и выгрузки данных из базы данных с большой скоростью. Основная реализация заключена в пакет DBMS_DATAPUMP в составе БД Oracle, в качестве клиентских утилит используются expdb и impdb для экспорта и импорта данных соответственно.

Утилиты же exp и imp включены в состав Oracle DB с ранних версий, имеют невысокую производительность так как поддерживают только однопоточную загрузку/выгрузку данных, в случае когда это возможно рекомендуется использовать более продвинутые инструменты технологии Data Pump.

30. Автоматическое управление хранением (ASM). Назначение, основные возможности.

Технология в составе СУБД Oracle, реализующая автоматическое управление хранением данных на уровне менеджера томов. Управление осуществляется отдельным экземпляром Oracle - Asm Instance. Один экземпляр может управлять хранением данных сразу нескольких БД. Одним из несомненных преимуществ является возможность расширять хранилище без остановки БД, кроме того ASM полностью поддерживает Oracle RAC. Внутри ASM файлы БД разделены на блоки определённого размера, для определения физического расположения блоков используются индексы.

//Олежя Топалов :D (Этот Олег-0)

31. Экземпляр ASM. Конфигурация, взаимодействие с экземпляром Oracle. Дополнительные процессы в составе экземпляра ASM и экземпляра Oracle.

Automatic Storage Management (ASM) — технология в составе СУБД Oracle, реализующая автоматическое управление хранением данных на уровне менеджера томов.

Особенности ASM:

- Управление хранением осуществляется с помощью специальных экземпляров Oracle — ASM Instances.
- ASM может управлять хранением данных как на отдельной машине, так и на уровне кластера RAC в целом.
- Один экземпляр ASM может управлять хранением данных сразу нескольких БД.
- Распределённым хранилищем управляет отдельный экземпляр Oracle (ASM Instance), независимый от «основного» экземпляра БД. Этот экземпляр должен быть запущен до запуска экземпляра БД.
- Экземпляр БД получает у экземпляра ASM информацию о расположении необходимых ему файлов, после чего работает с ними напрямую, без участия экземпляра ASM.
- Диски, на которых хранятся данные, объединяются в логические группы — disk groups.
- Экземпляр ASM содержит ряд дополнительных фоновых процессов, которых нет в «обычном» экземпляре БД:
 - RBAL (Rebalance) — управляет перераспределением ресурсов при изменениях в дисковых группах.
 - ARBn (Asm Rebalance Process) — пул процессов, непосредственно осуществляющих перемещение данных AU между дисками.
 - GMON (Group Monitor) — осуществляет мониторинг состояния дисков в группах.

- В случае использования ASM, в экземпляре БД также появляются «дополнительные» процессы:
 - RBAL — управляет доступом к дискам в группах.
 - ASMB (ASM Background Process) — осуществляет взаимодействие с экземпляром ASM.

При создании экземпляра ASM используется файл параметров — такой же, как и при создании «обычного» экземпляра Oracle. Тем не менее, он содержит ряд специфичных для ASM параметров:

- INSTANCE_TYPE — должен быть задан как ASM.
- DB_UNIQUE_NAME — имя сервиса ASM.
- ASM_POWER_LIMIT — определяет количество ресурсов, которые может использовать ASM при «перебалансировке» БД. Чем выше это значение, тем быстрее ASM выполняет перебалансировку, но тем больше он при этом потребляет ресурсов. Принимает значения от 1 до 11.
- ASM_DISKSTRING — параметр, определяющий набор дисков, которые «видит» ASM.
- ASM_DISKGROUPS — список имён дисковых групп, которые «видит» ASM в момент запуска

32. Дисковые группы. Назначение, особенности конфигурации. Добавление и удаление дисков. Allocation Units. Coarse- & Fine-Grained Striping.

- Дисковая группа — логическое объединение нескольких дисков ASM.
- Может хранить данные нескольких БД.
- Одна БД может хранить свои данные в нескольких дисковых группах.
- Диск может принадлежать только одной дисковой группе.
- Файл ASM может быть сохранён только на одной дисковой группе.
- Файлы хранятся распределённо — сразу на всех дисках, входящих в соответствующую группу.

Динамическая перебалансировка

- Автоматически осуществляется при каждом изменении конфигурации хранилища (например, при добавлении в него нового диска).
- Не требует перезапуска БД или какого-либо ограничения доступа к ней.
- Переразмещает данные в соответствии с новой ёмкостью хранилища.
- Осуществляется автоматически, никакой дополнительной конфигурации не требуется.
- Нагрузкой на систему можно управлять с помощью параметра ASM_POWER_LIMIT.

| Свойство | Create / Alter | Допустимые значения | Что определяет |
|------------------|----------------|--------------------------|-------------------|
| au_size | C | 1 2 4 8 16 32 64 MB P | Размер AU |
| compatible.rdbms | AC | Валидная версия СУБД | Формат сообщений, |

| | | | |
|---------------------------|----|-----------------------|--|
| | | | которыми обмениваются БД и ASM |
| compatible.asm | AC | Валидная версия ASM | Формат, в котором хранит свои данные ASM |
| disk_repair_time | A | От 0 М до 2^32 D | Время на восстановление, после которого диск переводится в состояние OFFLINE |
| template.tname.redundancy | A | UNPROTECT MIRROR HIGH | Способ организации зеркалирования |
| template.tname.stripe | A | COARSE FINE | Способ размещения данных на дисках |

Удаление диска из dgroupA:

```
ALTER DISKGROUP dgroupA DROP DISK A5;
```

Отмена операции удаления диска: ALTER DISKGROUP dgroupA UNDROP DISKS;

Добавление дисков в группы

```
ALTER DISKGROUP dgroupA ADD DISK
```

```
'/dev/rdisk/c0t4d0s2' NAME A5,
```

```
'/dev/rdisk/c0t5d0s2' NAME A6,
```

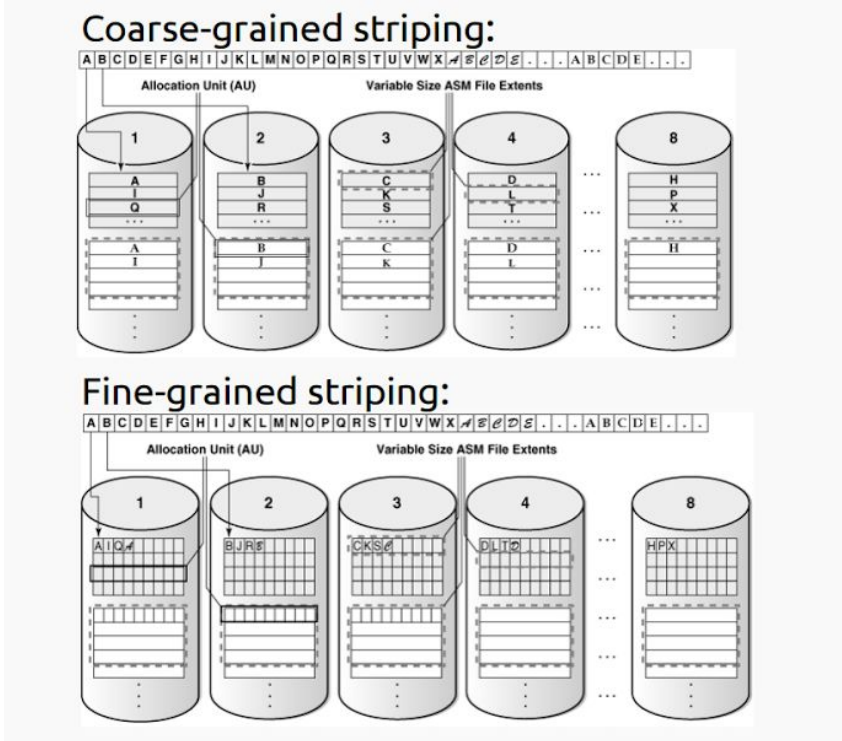
```
'/dev/rdisk/c0t6d0s2' NAME A7,
```

```
'/dev/rdisk/c0t7d0s2' NAME A8;
```

```
ALTER DISKGROUP dgroupA ADD DISK '/devices/A*';
```

Диски ASM разбиты на блоки распределения (allocation units — Aus):

Размер AU по умолчанию — 1 МБ. AU — минимальный объём дискового пространства, которым может оперировать ASM. Один блок данных может храниться только в одном конкретном AU.



ASM: Difference's between CORSE and FINE Striping

COARSE STRIPING

It is used for all voluminous input/output, e.g. input/output operations on datafiles.
The size of the coarse grained data stripes is large.

It manages the load balance across the disk groups.

The size of the coarse-grained stripe is always equal to the size of ASM Allocation Units (AU).

The size for coarse striping can be set using the `_asm_au_size` parameter.

FINE GRAINED STRIPING

It is used for all small input/output, e.g. input/output operations on online redolog files and control files.
The size of the fine grained data stripes is small.

It spreads the load on disk groups reducing latency for certain file types.

The size of the fine-grained stripe is always 128 KB.

The size for fine grained striping can be set using the `_asm_stripe_size` parameter.

33. Избыточность в Oracle ASM. Виды избыточности, конфигурация зеркалирования. Failure-группы.

ASM Failure Group - Набор дисков внутри конкретной группы, использующий общий ресурс (например, контроллер), от отказа которого должна быть обеспечена защита. По умолчанию ASM помещает каждый диск в свою собственную failure group. Если вариант «по умолчанию» не устраивает администратора, диски можно

перегруппировать. ASM автоматически располагает данные так, чтобы отказ разделяемого ресурса failure group не привёл к потере данных.

Зеркалирование дисковых групп Реализовано на уровне AU (а не на уровне дисков). На одном и том же диске могут храниться оригиналы и резервные копии различных AU. Если «оригинал» AU хранится на одном диске, то его «зеркало» будет храниться на другом диске в рамках той же группы.

3 режима организации избыточности данных:

- «Внешняя» (External Redundancy) — используется аппаратная реализация зеркалирования.
- «Стандартная» (Normal Redundancy): – Двукратное зеркалирование. – Как минимум, 2 failure groups.
- «Высокая» (High Redundancy): – Тройное зеркалирование. – Как минимум, 3 failure groups.

34. Роли ASM, отличия в уровне привилегий между ними.

Существует 3 роли, под которыми пользователь может подключиться к ASM:

- SYSASM и SYSDBA — административный доступ без каких-либо ограничений.
- SYSOPER — набор доступных SQL-команд ограничен минимально необходимым для обслуживания уже сконфигурированной системы:
 - STARTUP/SHUTDOWN
 - ALTER DISKGROUP MOUNT/DISMOUNT
 - ALTER DISKGROUP ONLINE/OFFLINE DISK
 - ALTER DISKGROUP REBALANCE
 - ALTER DISKGROUP CHECK
 - SELECT all V\$ASM_* views

Все остальные команды, в частности, CREATE DISKGROUP, ADD/DROP/RESIZE DISK и т. д., требуют наличия привилегий SYSASM или SYSDBA.

35. Архитектура быстрого восстановления с использованием сервера горячего резерва и кластерная архитектура RAC.

Физическая резервная база данных является поблочной копией первичной базы данных. Во время восстановления в аварийных ситуациях, резервная база данных в точности похожа на основную базу данных.

Для поддержания актуальности данных на standby сервере периодически должны применяться изменения их журнала повторов. Для этого с действующей БД должны копироваться архивы журналов повторов на сервер резервной БД. Для этого бд должна быть в режиме ARCHIVELOG

Архитектура Real Applications Clusters

В "обычной" некластерной конфигурации к базе данных эксклюзивно имеет доступ один экземпляр программного обеспечения СУБД Oracle.

С кластерной базой данных, расположенной на общих дисковых устройствах, одновременно могут работать множество экземпляров СУБД Oracle, запущенных на различных узлах кластера. При увеличении вычислительных потребностей в кластер без остановки его работы можно добавить дополнительные узлы и экземпляры. Новые ресурсы могут быть задействованы в работу сразу после подключения. Для объединения аппаратных компонентов в единую вычислительную систему необходимо кластерное программное обеспечение, управляющее членством узлов в кластере, осуществляющее мониторинг состояния и управление различными составляющими и служб кластера, предоставляющее механизм для взаимодействия между приложениями работающими на разных узлах и другие важные базовые функции.

36. Понятие кластера. Аппаратная и программная реализация кластера. Масштабируемость и отказоустойчивость кластерных решений.

Если приложение хорошо масштабируется на SMP, то оно будет хорошо масштабироваться и на кластере.

Аппаратная архитектура

Компьютер, входящий в состав кластера, называется узлом кластера. Каждый узел кластера должен иметь подключение к локальной сети, интерфейс для кластерного межсоединения и доступ к общему дисковому устройству хранения. Oracle Real Application Clusters строится по архитектуре с разделяемыми дисками - все сервера кластера должны иметь полный и равноправный доступ ко всем устройствам хранения, на которых размещается кластерная база данных Oracle.

В качестве таких дисковых устройств могут быть использованы дисковые устройства, подключаемые через сеть (NAS), специализированные сети устройств хранения (SAN) или SCSI-диски. На выбор типа устройства хранения часто влияет размер кластера, используемая платформа сервера и поддержка производителя аппаратного обеспечения. Устройство хранения должно обеспечить достаточную производительность операций ввода-вывода приложений и быть хорошо масштабируемым при добавлении в кластер дополнительных узлов.

Для работы кластера необходимо две изолированных друг от друга сети. Публичная сеть для связи между клиентами и серверами кластера. С использованием этой сети производится подключение клиентских сессий к базе данных, их балансировка между узлами и аварийное переключение в случае сбоя. Приватная или внутренняя сеть, обычно называемая межсоединением, необходима для передачи сообщений между узлами. В RAC межсоединение используется для реализации технологии "слияния" кэш (Cache Fusion) различных узлов кластера.

Начиная с СУБД Oracle версии 10g в комплект дистрибутива включено специально разработанное и интегрированное с СУБД программное обеспечение для объединения компьютеров в кластер - Oracle Clusterware. Теперь для того, что бы иметь кластерную базу данных RAC, больше нет необходимости приобретать кластерное программное обеспечение от других производителей. Установка Oracle Clusterware достаточно просто производится с помощью хорошо знакомой каждому администратору базы данных, универсальной программой инсталляции Oracle (Oracle Universal Installer). Тот факт, что у кластерного программного обеспечения и СУБД единый разработчик, упрощает настройку и техническую поддержку.

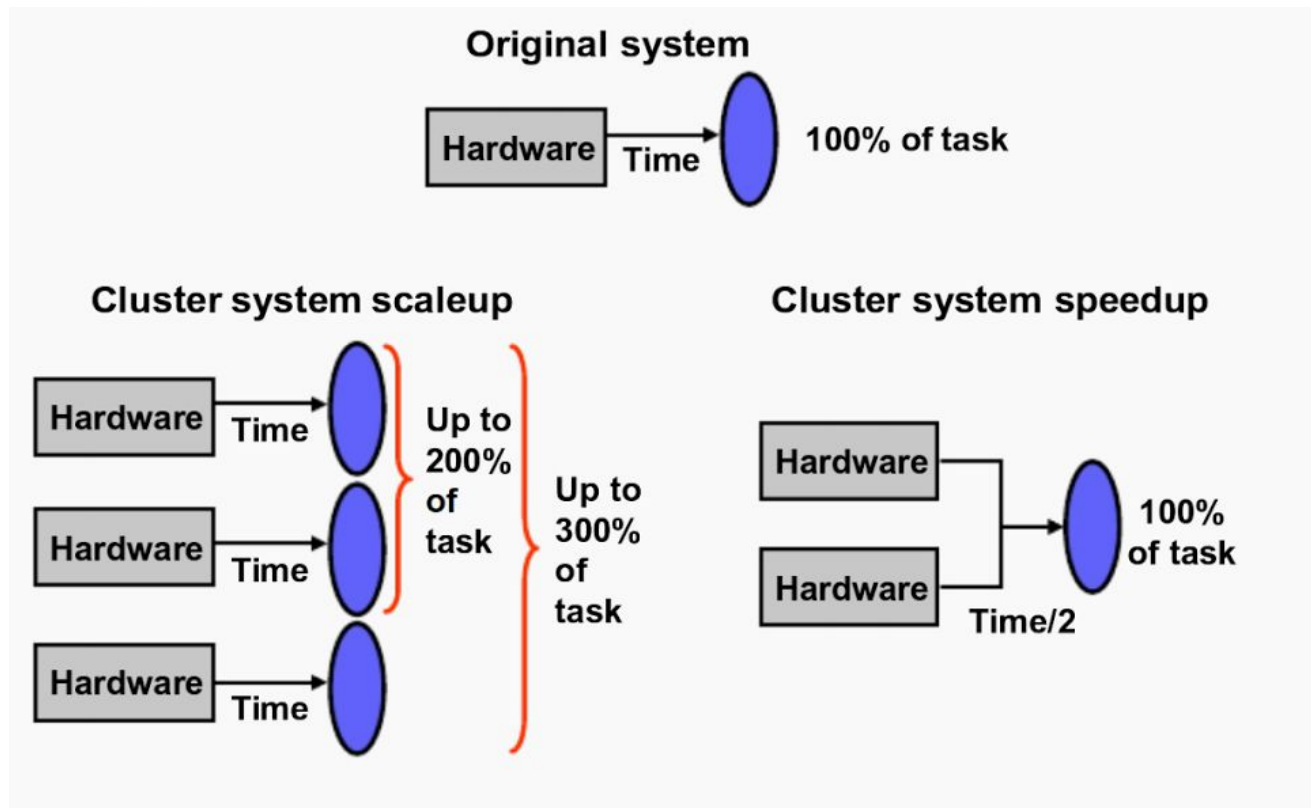
Множество экземпляров Oracle составляют единую БД. На каждой машине запущено по одному экземпляру Oracle. Файлы БД — «общие» для всех экземпляров Oracle внутри кластера. Доступом к данным и взаимодействием между экземплярами Oracle управляет специальное инфраструктурное ПО.

37. Основные принципы построения масштабируемых приложений. Speedup & Scaleup.

Для того, чтобы использование кластера было эффективным, требуется обеспечить масштабируемость на всех уровнях:

- Аппаратный — скорость чтения / записи на диски.
- Взаимодействие между узлами — пропускная способность сети и время отклика.
- Операционная система — возможность работы на многопроцессорных машинах.
- СУБД — синхронизация при параллельном доступе к данным.
- Приложение — особенности архитектуры.

ScaleUp and SpeedUp



38. Глобальные ресурсы Oracle RAC, особенности управления ими. Глобальные DPV.

- ✓ **Два и более экземпляра работают с одной базой данных**
- ✓ **Необходима координация действий экземпляров**
 - ✓ «Глобальная служба кэша» («Global Cache Services», **GCS**) обеспечивает скоординированный доступ к блокам БД в кешах буферов разных экземпляров
 - ✓ «Глобальная служба блокировок» («Global Enqueue Services», **GES**) управляет транзакционными блокировками (на уровне словаря данных) и другими общими ресурсами (например, SCN)
- ✓ **GCS и GES**
 - ✓ реализованы в виде дополнительных фоновых процессов экземпляра (LMSn, LMON, LMD, LCKn, DIAG)
 - ✓ работают с распределенной по экземплярам базой «глобальных ресурсов» посредством «глобальных блокировок»
 - ✓ В предыдущих версиях – DLM (Distributed Lock Manager)
- ✓ **GCS использует механизм «Cache Fusion»** для обмена содержимым кэша буферов между экземплярами через высокоскоростной Interconnect, что существенно расширяет возможности масштабирования приложений, которые невозможно было использовать в среде Oracle Parallel Server в предыдущих версиях ПО Oracle

Технология общего кэша в кластере называется Cache-fusion (синтез кэша). CRS на каждом узле порождает синхронные процессы LMSn, общее их название как сервиса — GCS (Global Cache Service). Эти процессы копируют прочитанные на этом экземпляре блоки (глобальные) из буферного кэша к экземпляру, который за ними обратился по сети, и также отвечают за откат неподтвержденных транзакций. За их координацию отвечает сервис GES (Global Enqueue Service), представленный на каждом узле процессами LMON и LMD. LMON отслеживает глобальные ресурсы всего кластера, обращается за блоками к соседним узлам, управляет восстановлением GCS. LMD управляет ресурсами узла, контролирует доступ к общим блоками и очередям, отвечает за блокировки запросов к GCS и управляет обслуживанием очереди запросов LMSn. Особая роль в координации ресурсов кластера и объединения кэша отведена таблице GRD (Global Resource Directory), в которой постоянно фиксируется, на каком экземпляре, какой блок (или его копия) доступен, режим, в котором экземпляр удерживает блок, роль блока, SCN. Таблица GRD распределена между узлами кластера. Каждый узел принимает участие в распределении ресурсов кластера, обновляя свою часть GRD. Часть таблицы GRD относится к ресурсам – объектам: таблицы, индексы и.т.п. Она постоянно синхронизируется (обновляется) между узлами. Когда узел прочел блок данных с диска, он становится master-ом этого ресурса и делает соответствующую отметку в своей части таблицы GRD. Блок помечается как локальный, т.к. узел пока использует его в одиночку. Если же этот блок потребовался другому узлу, то процесс GCS пометит этот блок в таблице как глобальный («опубликован» для кластера) и передаст затребовавшему узлу.

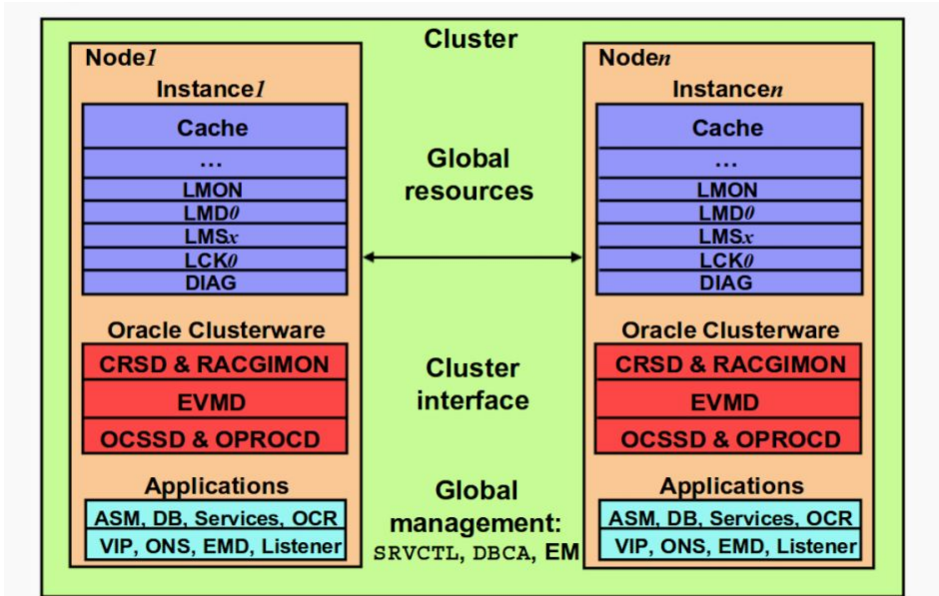
Глобальные DPV

- Содержат информацию обо всех запущенных экземплярах в составе кластера.
- У каждого локального представления (V\$) есть соответствующее ему глобальное представление (GV\$).

- Исполняются параллельно на всех узлах кластера — «ведущий» запрос на узле, к которому осуществляется обращение и «ведомые» запросы к V\$ на остальных узлах.
- Параллелизмом управляет специальный сервис — координатор параллельного исполнения (Parallel Execution Coordinator, PEC).

39. Архитектура Oracle RAC: процессы, конфигурационные файлы, файлы БД.

Процессы:

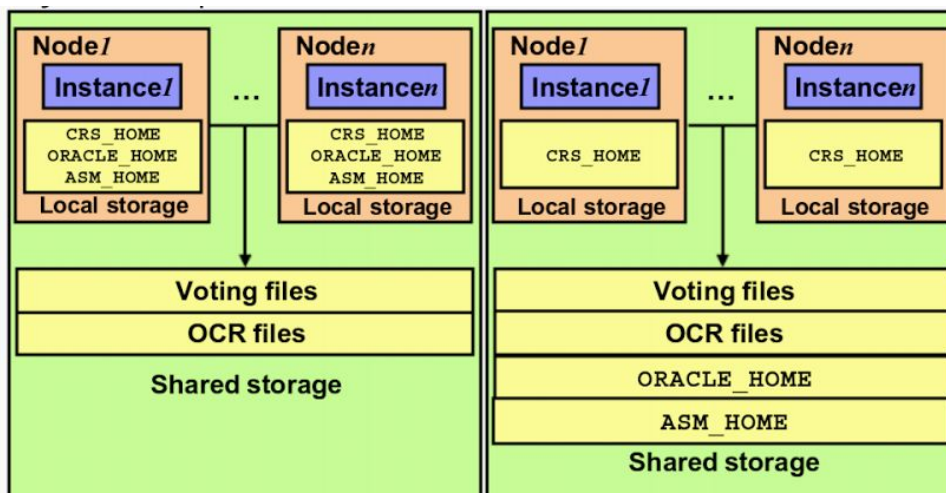


Конфигурационные файлы

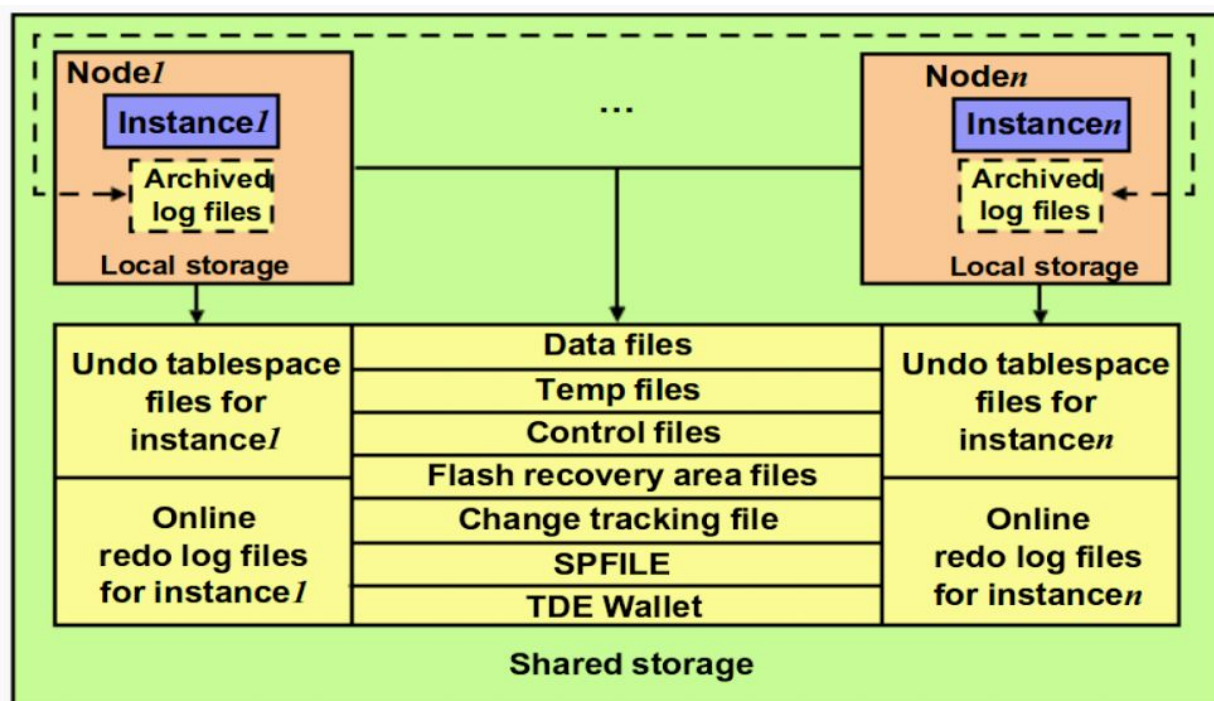
Появляются 2 новых категории файлов

- Файлы с результатами мониторинга состояния кластера.
- Файлы OCR — Oracle Cluster Registry.

Файлы OCR не могут храниться в ASM, т. к. они должны быть доступны ещё до запуска кластера.



Файлы БД:



40. Варианты построения системы хранения в Oracle RAC, их преимущества и недостатки.

CFS и RAW

CFS-Файловая система, специально разработанная для хранения разделяемых ресурсов RAC.

Особенности:

- Позволяет всем узлам кластера использовать общий ORACLE_HOME.
- Каждый том OCFS может размещаться на одном или нескольких физических дисках.

В файловом хранилище на базе OCFS можно разместить:

- Установленные бинарные файлы Oracle.
- Файлы экземпляра Oracle (конфигурационные, файлы данных и т. д.).
- Файлы параметров инициализации (spfile).
- Временные файлы, созданные экземпляром Oracle в время работы.
- Voting & OCR files.

Преимущества CFS:

- Проще администрировать.
- Ставится вместе с Oracle, не нужна дополнительная конфигурация.
 - Автоматически расширяется по мере возрастания объема данных.
- Можно использовать для хранения файлов архива журнала повторов.

Преимущества raw:

- Потенциально быстрее.
- Ниже требования к инфраструктуре.
- Можно использовать ASM для расширения возможностей.