

**Специальность 10.05.01 «Компьютерная безопасность»,
Специализация «Математические методы защиты информации»
Уровень высшего образования – специалитет**

Дисциплина: Основы построения защищенных баз данных.

**Лабораторная работа №1.
Администрирование Oracle. Управление экземпляром.**

1. Учебные цели:

- Отработать вопросы управления экземпляром Oracle в режиме командной строки SQL Plus и с использованием Web-интерфейса Enterprise Manager.
- Освоить приемы остановки и экземпляра БД, настройки параметров инициализации, просмотра журналов сигнальных сообщений, просмотра динамических представлений производительности.

2. Требования к результатам обучения основной образовательной программы, достигаемые при проведении лабораторной работы:

- Уметь использовать возможности современных систем для решения задач администрирования и защиты баз данных.
- Владеть средствами приложений СУБД Oracle для управления экземпляром Oracle в режиме командной строки SQL Plus и с использованием Web-интерфейса Enterprise Manager.

3. Перечень материально-технического обеспечения

ПЭВМ с проигрывателем виртуальных машин, виртуальная машина с установленной СУБД Oracle.

4. Краткие теоритические сведения и задания на исследование. Задания выделены рамками и синим шрифтом. Результаты лабораторной работы представляются в виде файла, содержащего копии экрана, показывающие этапы выполнения заданий.

Введение.

Архитектура базы данных Oracle

Сервер Oracle играет ключевую роль в управлении информацией. В общем случае сервер Oracle должен надежно и с высокой производительностью управлять большим объемом информации в многопользовательской среде, обеспечивая одновременный доступ пользователей к одним и тем же данным. Сервер Oracle должен препятствовать несанкционированному доступу и предоставлять эффективные решения для восстановления после сбоев.

Сервер Oracle:

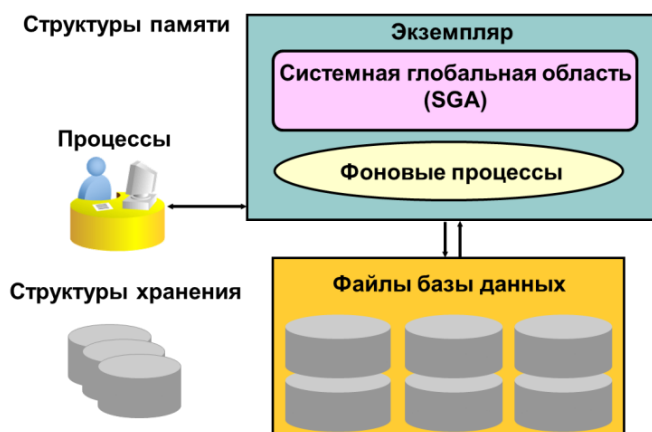
- система управления базой данных, предоставляющая открытый, полный и интегрированный подход к управлению информацией
- состоит из экземпляра Oracle и базы данных Oracle

Структуры базы данных.

Каждая работающая база данных Oracle связана с экземпляром (instance) Oracle. Когда на сервере баз данных стартует база данных, программное обеспечение Oracle выделяет память для системной глобальной области (System Global Area, SGA) и запускает несколько

фоновых процессов (background processes) Oracle. Такое объединение SGA и процессов Oracle называется экземпляром Oracle.

После запуска экземпляра программное обеспечение связывает экземпляр с определенной базой данных. Это называется монтированием базы данных. Вслед за этим база данных готова для открытия, после которого она становится доступной для зарегистрированных пользователей. На одном компьютере могут одновременно функционировать несколько экземпляров с доступом к своей собственной физической базе данных.



Архитектуру баз данных Oracle можно представлять как совокупность различных взаимосвязанных структурных компонентов.

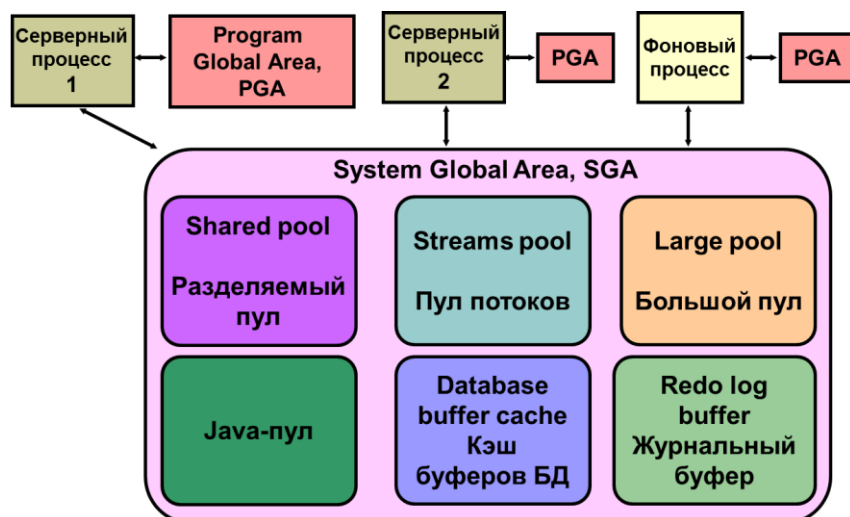
База данных Oracle использует структуры памяти и процессы для управления и доступа к базе данных. Все структуры памяти располагаются в основной памяти компьютеров, образующих сервер базы данных. Процессы представляют собой задания, работающие в памяти этих компьютеров. Процесс определяется как “поток управления” (“thread of control”) или как механизм (mechanism) в операционной системе, который может выполнять последовательность шагов.

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 (каждому процессу выделяется своя PGA).

SGA – это область разделяемой памяти, в которой содержатся данные и управляющая информация экземпляра.



Структуры данных SGA:

- **Кэш буферов БД (Database buffer cache);** содержит выбираемые из БД блоки данных.
- **Журнальный буфер (Redo log buffer);** кэширует информацию повторного выполнения (используемую при восстановлении экземпляра) до момента записи в журнальные файлы.
- **Разделяемый пул (Shared pool);** кэширует различные структуры, которые могут совместно использоваться пользователями.
- **Большой пул (Large pool)** – необязательная область, в которой отводится память для буферов определенных больших операций ввода/вывода (например, операций резервирования и восстановления), а также для серверных процессов ввода-вывода.
- **Java-пул;** используется для расположения Java-кода сеансов и данных внутри виртуальной Java-машины (Java Virtual Machine – JVM).
- **Пул потоков (Streams pool);** используется Oracle Streams.

При запуске экземпляра с помощью Enterprise Manager или SQL*Plus показывается размер памяти, выделенной для SGA.

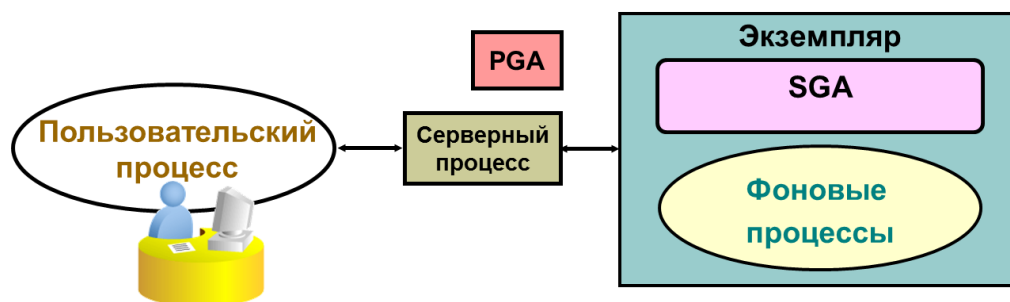
Программная глобальная область (PGA) – это область памяти, выделяемая для каждого серверного процесса, содержащая данные и управляющую информацию этого процесса. Серверный процесс – это процесс, который обрабатывает запросы клиента. Каждый серверный процесс имеет свою приватную область PGA, которая создается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

В рамках динамической инфраструктуры SGA можно без остановки экземпляра менять размеры кэша буферов БД, разделяемого пула, большого пула, Java-пула и пула потоков.

База данных Oracle использует параметры инициализации для создания и конфигурирования структур памяти. Например, параметр SGA_TARGET задает общий размер памяти, доступной для SGA. Когда параметр SGA_TARGET = 0, *автоматическое управление разделяемой памятью (Automatic Shared Memory Management)* выключено.

Процессы.

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

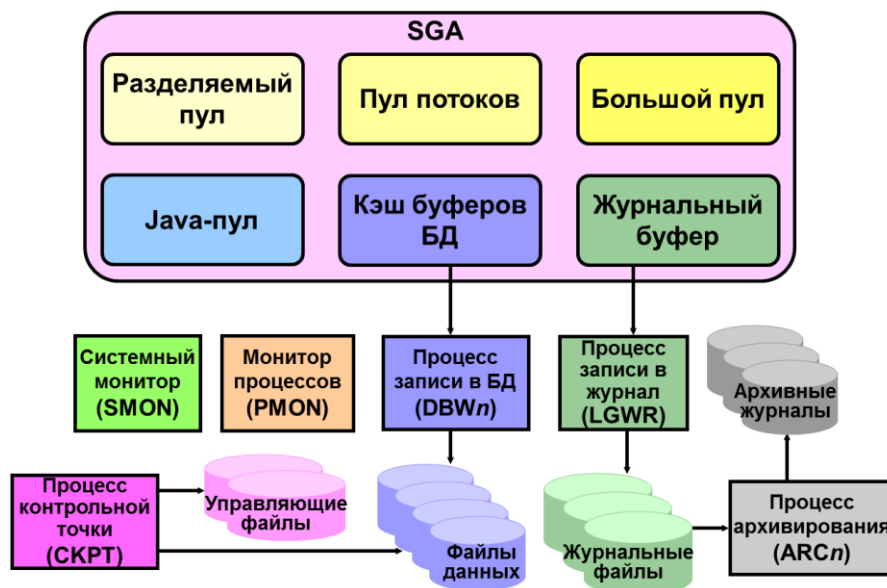


Кроме того, Oracle создает набор фоновых процессов для экземпляра. Эти процессы взаимодействуют друг с другом и с операционной системой. Они управляют структурами памяти, записывают информацию на диск в асинхронном режиме ввода/вывода и выполняют общесистемные служебные действия.

Состав работающих в текущий момент фоновых процессов зависит от используемых функциональных возможностей базы данных.

Управление экземпляром Oracle.

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

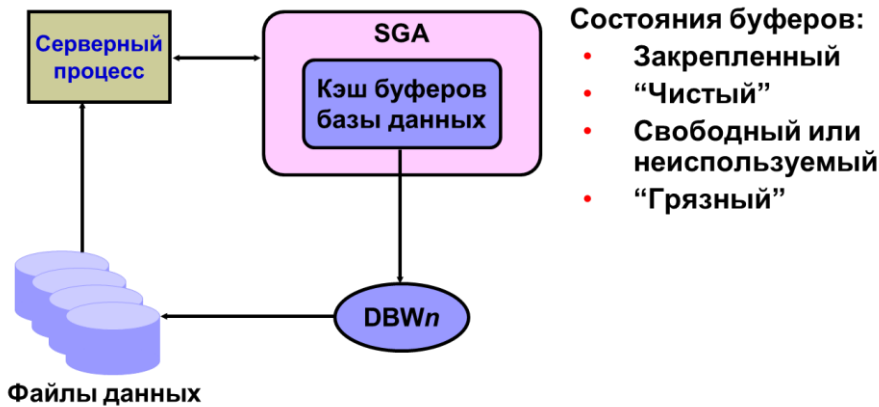


Наиболее общие фоновые процессы:

- **Системный монитор (System monitor – SMON)**; выполняет восстановление после отказа экземпляра при его последующем старте.
- **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)**; 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.

Серверный процесс и кэш буферов базы данных.

При обработке запроса серверный процесс Oracle ищет требуемые блоки в кэше буферов БД. Если блок не найден, серверный процесс читает блок из файла данных и размещает его копию в кэше буферов БД. Поскольку последующие запросы этого же блока обнаружат его в оперативной памяти, могут не понадобиться дополнительные физические чтения. Сервер Oracle использует алгоритм удаления *наиболее давно использовавшихся буферов* (*least recently used algorithm – LRU*) для их выгрузки с целью освобождения места и размещения новых блоков в кэше буферов.



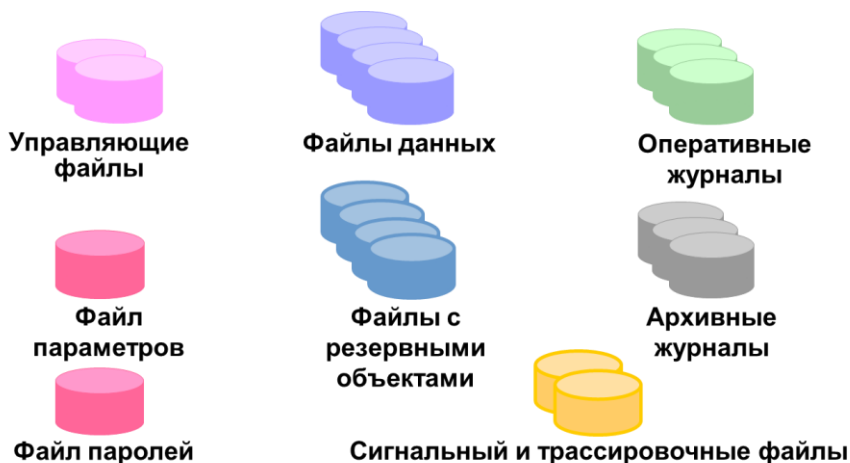
Буферы в кэше буферов могут быть в одном из четырех состояний:

- **Pinned** (закрепленный) – означает, что несколько сеансов не могут в один и тот же момент времени писать в один блок и вынуждены ждать доступа к блоку, находящемуся в буфере.
- **Clean** (“чистый”) – означает, что буфер в настоящее время не закреплен (unpinned) и является кандидатом на удаление из кэша, если на его содержимое не будет опять ссылок. Содержимое буфера либо синхронизировано с блоком на диске, либо буфер использовался для генерации и обработки старого *моментального снимка (snapshot)* блока в режиме *целостного чтения (consistent read – CR блок)*.
- **Free/unused** (свободный/неиспользуемый) – означает, что буфер пустой, т.к. экземпляр только что был запущен. Состояние очень похоже на состояние clean, за исключением того, что буфер еще не использовался.
- **Dirty** (“грязный”) – буфер больше не является закрепленным, но его содержимое (блок данных) было изменено и должно быть записано на диск процессом DBWn перед удалением из кэша.

Физическая структура базы данных.

В базе данных Oracle находятся перечисленные ниже виды файлов.

- **Управляющие файлы;** содержат метаданные о самой БД (например, данные о физической структуре). Эти файлы очень важны для БД. Без них не могут быть открыты файлы данных и поэтому не может быть открыт доступ к информации БД.
- **Файлы данных;** содержат данные пользователей и приложений, хранимые в БД.
- **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.



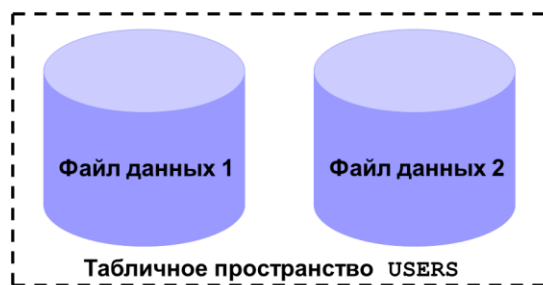
Большое значение для успешной работы БД имеют следующие файлы.

- **Файл параметров;** используется для описания стартовой конфигурации экземпляра.
- **Файл паролей;** позволяет пользователям удаленно подключиться к базе данных для выполнения административных задач.
- **Файлы с резервными объектами;** используются для восстановления базы данных. Эти файлы обычно *копируются из резерва (restore)* в случае *потери носителя (media failure)*, а также в случае пользовательской ошибки, приведшей к повреждению или удалению первоначального файла.
- **Архивные журналы (архивные файлы повторного выполнения);** хронологические копии созданных экземпляром оперативных журналов. Эти файлы позволяют восстановить базу данных. Для этого сначала из резервных объектов базы данных извлекаются потерянные файлы данных, которые затем восстанавливаются с помощью архивных и оперативных журналов.
- **Трассировочные файлы;** каждый серверный и фоновый процесс может писать в связанный с ним трассировочный файл. При обнаружении внутренней ошибки процесс выводит дамп с информацией об ошибке в трассировочный файл. Некоторая часть информации, которая пишется в трассировочный файл, предназначена для администратора базы данных, другая часть данных необходима службам технической поддержки Oracle (Oracle Support Services).
- **Сигнальные файлы;** являются специальными трассировочными файлами. Сигнальный файл базы данных представляет собой хронологический журнал сообщений и ошибок. Oracle рекомендует просматривать такие файлы.

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.

- **Табличные пространства состоят из одного или более файлов данных**
- **Файлы данных принадлежат только одному табличному пространству**



Примечание. Можно создать табличное пространства вида `bigfile`, которое представляет собой пространство с одним, но очень большим (до 4 миллиардов блоков) файлом данных. Традиционные табличные пространства вида `smallfile` (они создаются по умолчанию) могут содержать несколько файлов данных, которые не могут быть такими большими.

Табличные пространства SYSTEM и SYSAUX.

Каждая база данных Oracle содержит табличные пространства SYSTEM и SYSAUX. Они автоматически создаются, когда создается база данных. По умолчанию система создает

табличные пространства вида `smallfile`. Но можно создать табличные пространства вида `bigfile`, что позволит базе данных Oracle сопровождать сверхбольшие файлы (до 8 эксабайтов).

Табличное пространство может находиться в оперативном (доступном) или автономном (недоступном) состоянии.

`SYSTEM` всегда в оперативном состоянии, когда открыта база данных. В нем хранятся таблицы, поддерживающие функциональные возможности ядра базы данных (например, таблицы словаря данных).

`SYSAUX` – вспомогательное по отношению к `SYSTEM` табличное пространство. Оно хранит многие компоненты базы данных и должно быть в оперативном состоянии для корректного функционирования базы данных, используется дополнительными компонентами базы данных (например, репозиторием `Enterprise Manager`).

Сегменты, экстененты и блоки.

Такие объекты БД, как таблицы и индексы, хранятся в табличных пространствах в виде сегментов. Каждый сегмент состоит из одного или более *экстенентов*. Экстенент состоит из смежных блоков данных. Поэтому каждый экстенент 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).

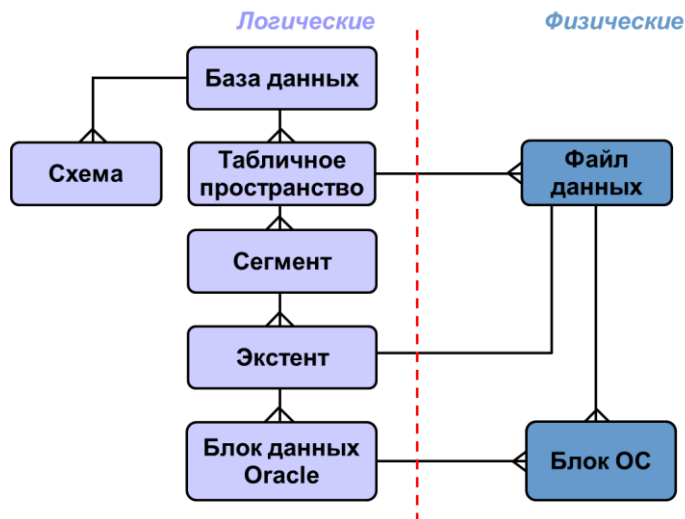
Можно создавать табличные пространства с различными размерами блоков. Обычно это вызвано только необходимостью поддержки переносимых табличных пространств.

- Сегмент существует в рамках табличного пространства.
- Сегмент состоит из набора экстенентов.
- Экстененты – это набор смежных блоков данных.
- Блоки данных отображаются в блоки ОС.



Логические и физические структуры базы данных

База данных Oracle – совокупность данных, обрабатываемых как единое целое. Общее назначение базы данных заключается в хранении и получении связанной информации. База данных состоит из логических и физических структур.



Табличные пространства

База данных делится на логические компоненты хранения, называемые табличными пространствами. В них объединяются связанные между собой логические структуры. Например, в табличном пространстве обычно группируют все объекты приложения для упрощения некоторых административных операций. Можно создать табличное пространство для данных приложения и дополнительно табличное пространство для индексов приложения.

Базы данных, табличные пространства и файлы данных

На слайде показаны взаимосвязи между базой данных, табличными пространствами и файлами данных. Каждая база данных логически делится на одно или несколько табличных пространств. Для каждого табличного пространства явно создается один или несколько файлов данных. В них хранятся данные всех логических структур, размещенных в табличном пространстве. Для временного (TEMPORARY) табличного пространства вместо файла данных создается файл временных данных (tempfile).

Схемы

Схема – совокупность объектов базы данных, принадлежащих пользователю БД. К объектам схемы, например, относятся таблицы, представления, последовательности, хранимые процедуры, синонимы, индексы, кластеры и связи базы данных. В общем случае в число объектов схем включается все, что приложение создает в базе данных.

Блоки данных

Данные в Oracle на самом низком структурном уровне хранятся в блоках данных. Один блок данных связан с конкретным количеством байтов в физическом пространстве базы данных на диске. Для каждого табличного пространства при его создании задается размер блока. База данных использует и распределяет свое свободное пространство в блоках данных Oracle.

Экстенды

Следующий более высокий уровень логического пространства базы данных называется экстендом. Он содержит конкретное количество смежных блоков данных (полученных при одном выделении), используемых для хранения информации определенного вида.

Сегменты

В логической структуре БД сегмент располагается на один уровень выше экстенда. Сегмент – набор экстендов, выделенных определенной логической структуре. Ниже перечислены некоторые типы сегментов.

- **Сегменты данных;** каждая таблица, отличная от кластеризованной и индекс-таблицы, имеет сегмент данных. Все табличные данные хранятся в экстендах своего сегмента данных. В секционированной таблице у каждой секции свой сегмент данных. У каждого кластера также имеется свой сегмент данных. Данные всех таблиц кластера располагаются в кластерном сегменте данных.

- **Индексные сегменты;** у каждого индекса имеется индексный сегмент, в котором находятся все его данные. В секционированном индексе каждая секция имеет свой индексный сегмент.

- **Сегменты отмены;** администратор базы данных создает одно табличное пространство типа UNDO для временного хранения информации *отмены (undo)*. Эта информация используется для генерации данных при согласованном чтении из базы данных, а также при восстановлении базы данных и откате незафиксированных транзакций пользователей.

- **Временные сегменты;** создаются Oracle, когда командам SQL требуется рабочая область для завершения выполнения. После окончания выполнения экстенды временного сегмента возвращаются экземпляру для последующего использования. Задавайте временное табличное пространство для каждого пользователя или же временное табличное пространство, используемое по умолчанию для всей базы данных в целом.

База данных Oracle выделяет пространство динамически. Дополнительные экстенды выделяются, когда заполнены все экстенды сегмента. Так как выделение экстендов производится, когда это необходимо, на диске экстенды одного и того же сегмента могут быть смежными и несмежными.

Задачи администратора базы данных Oracle.

Задачи проектирования, реализации и сопровождение базы данных можно расположить в следующем приоритетном порядке:

1. Оценка аппаратного обеспечения сервера БД
2. Установка программного обеспечения Oracle
3. Планирование базы данных и стратегия безопасности
4. Создание, миграция и открытие базы данных
5. Резервирование базы данных
6. Регистрация пользователей системы и планирование сетевого доступа к Oracle
7. Реализация проекта базы данных
8. Восстановление после сбоев БД
9. Мониторинг базы данных

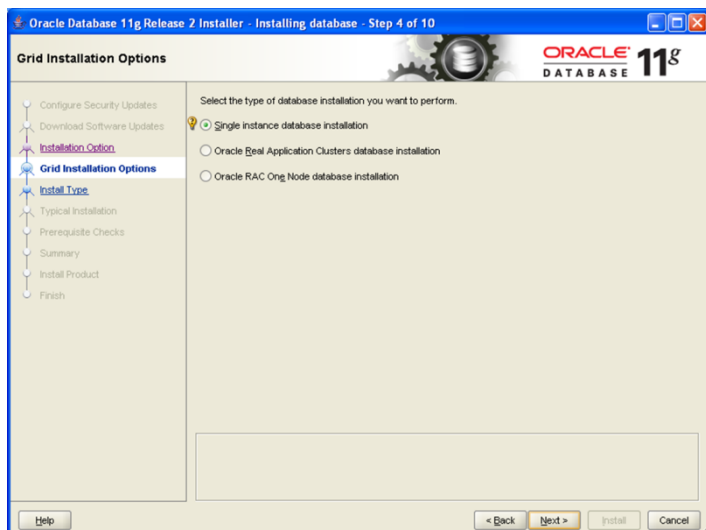
Инструментальные средства, используемые для администрирования базы данных Oracle

Следующие инструментальные средства используются для инсталляции и обновления версии (upgrade).

- **Oracle Universal Installer (OUI);** устанавливает программное обеспечение и опции Oracle. OUI может автоматически запустить утилиту Database Configuration Assistant (DBCA) для создания базы данных.
- **Database Configuration Assistant (DBCA);** создает базу данных, используя шаблоны, поставляемые Oracle. В этом случае копируется преконфигурированная начальная база данных. Предоставляются и другие варианты использования этой утилиты, позволяющие создать свою базу данных и шаблоны.
- **Database Upgrade Assistant (DBUA);** этот инструмент используется для обновления версии существующей базы данных в соответствии с новым релизом Oracle.
- **Oracle Net Manager;** применяется для конфигурирования сетевых соединений баз данных и приложений Oracle.

Oracle Universal Installer (OUI).

Oracle Universal Installer (OUI) – это приложение на языке Java, позволяющее устанавливать компоненты и комплексы на различных уровнях, пакеты целиком со всеми возможностями, а также использовать Web-доступ к дистрибутиву. Это средство установки легко переносится на платформы, поддерживающие Java, и проблемы, зависящие от платформ, могут быть отделены от общего процесса инсталляции.



Oracle Universal Installer обеспечивает следующие возможности:

- автоматическое разрешение зависимостей и обработка сложной логики;
- установка с использованием Web;
- установка компонентов и комплексов;
- неявная деинсталляция;
- поддержка нескольких домашних директорий Oracle;
- национальная языковая поддержка (national language support – NLS)/поддержка глобализации;
- поддержка распределенной установки;
- автоматическая (“тихая” – “silent”) установка с использованием файлов ответов.

Следующие инструментальные средства используются для управления экземпляром и базой данных Oracle.

- **Oracle Enterprise Manager (EM)**; EM объединяет графическую консоль, агенты, общие службы и инструменты для интегрированного и всестороннего управления системной платформой и функционирующими на ней продуктами Oracle. После установки программного обеспечения Oracle, создания и обновления версии базы данных и конфигурирования сети Enterprise Manager можно использовать в качестве единственного интерфейса управления вашей БД. Дополнительно к основанному на Web-доступе пользовательскому интерфейсу для выполнения команд SQL Enterprise Manager предоставляет бесшовный интерфейс к другим компонентам Oracle, используемым для администрирования базы данных (например, Recovery Manager и планировщику заданий).

Три основных инструмента Enterprise Manager для администрирования БД Oracle:

- Enterprise Manager Database Console; используется для администрирования одной базы данных;
- Enterprise Manager Grid Control; используется для одновременного администрирования многих баз данных;
- Enterprise Manager Java Console; используется для доступа к инструментам, недоступным через Web-интерфейс.
- **SQL*Plus**; стандартный интерфейс командной строки для управления базой данных
- **iSQL*Plus**; интерфейс к базе данных Oracle из браузера.
- **Recovery Manager (RMAN)**; инструмент Oracle, предоставляющей полное решение задач резервирования, копирования из резерва и восстановления для всей базы данных и заданных файлов БД.

- **Oracle Secure Backup** позволяет в экосистеме Oracle управлять задачами резервирования на ленту:

- защита доступа к лентам, обеспечиваемая БД Oracle на основе интеграции с Recovery Manager

- бесшовная поддержка Oracle Real Application Clusters (RAC)
- централизованное администрирование распределенных клиентов и серверов среднего уровня (Oracle Application Servers, Oracle Collaboration Suites, домашних каталогов Oracle и двоичных файлов).
- **Data Pump**; утилита высокоскоростной передачи данных из одной базы данных в другую. Например, можно экспортировать таблицу и импортировать ее в другую БД.
- **SQL*Loader**; позволяет загрузить данные из внешнего файла в базу данных Oracle. Это одна из утилит Oracle, которой можно воспользоваться для загрузки данных в таблицы БД.
- **Инструменты командной строки:**
 - Для управления Enterprise Manager используется команда:
emctl start | status | set | stop
 - Для запуска и остановки iSQL*Plus используется команда:
isqlplusctl start | stop
 - Для управления процессом прослушивания используется команда:
lsnrctl help | start | status | stop

Установка переменных среды операционной системы.

ORACLE_BASE: корневой каталог дерева, в котором устанавливается программное обеспечение Oracle.

ORACLE_HOME: каталог для программного обеспечения версии сервера Oracle.

ORACLE_SID: имя экземпляра, с которым устанавливается соединение по умолчанию (обычно ORCL).

NLS_LANG: определяет начальные установки национальной поддержки сеанса. Задается в виде: язык_территория.кодировка.

Пример: RUSSIAN_CIS.CL8MSWIN1251

Параметр, приведенный в примере, задает русский язык (RUSSIAN) для сообщений Oracle, сортировок, наименований дней и месяцев. В качестве территории установлено значение CIS (Commonwealth of Independent States, Содружество Независимых государств, СНГ), определяющее формат времени, даты, числовых и стоимостных данных. Oracle Net использует для преобразования символьных данных клиента кодировку CL8MSWIN1251. Переменная NLS_LANG устанавливается в среде UNIX и в реестре Windows. Действующие в сеансе параметры национальной поддержки можно получить, используя запрос:

```
select * from nls_session_parameters;
```

1) Компоненты управления.

Три основных компонента управления Oracle Database:

- Экземпляр базы данных.
- *Процесс прослушивания (listener)*, обеспечивающий установление соединения с БД
- Интерфейс управления - консоль Oracle Enterprise Manager Database Control.

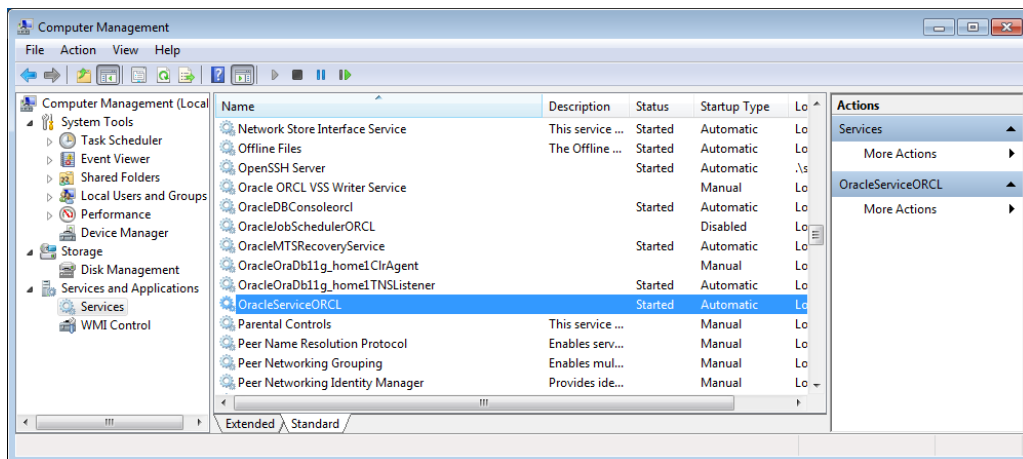
В Windows – системах эти компоненты функционируют как службы: OracleServiceORCL, OracleOraDB11g_home1TNSListener, OracleDBConsoleorcl соответственно.

Каждый из этих компонентов должен быть явно запущен перед тем, как можно будет использовать службы этих компонентов. При остановке сервера, на котором выполняется база данных, эти компоненты следует чисто останавливать.

Первым должен быть запущен интерфейс управления (в случае Database Control до него следует запустить процесс прослушивания). После этого могут быть запущены другие компоненты.

Запустите оснастку Windows «Службы».

Посмотрите, запущены ли службы OracleServiceORCL, OracleOraDB11g_home1TNSListener, OracleDBConsoleorcl.



Остановите службу OracleServiceORCL или OracleOraDB11g_home1TNSListener. Запустите Sqldeveloper, попробуйте подключиться к базе данных. Запустите службу, проверьте подключение.

Служба OracleDBConsoleorcl позволяет осуществлять администрирование базы данных с использованием Web-интерфейса Oracle Enterprise Manager (далее - **ЕМ**).

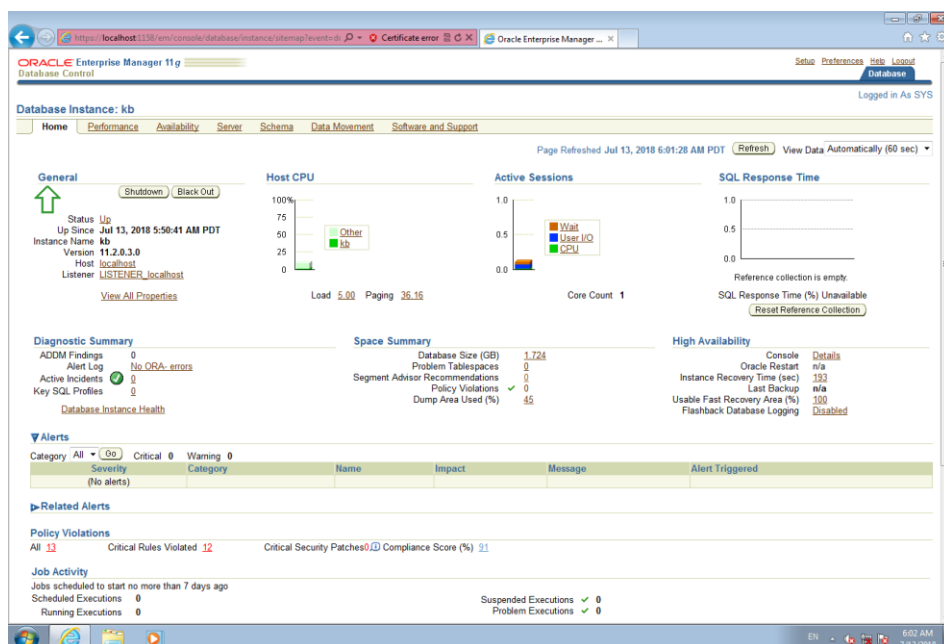
Для входа в ЕМ запустите Internet Explorer. В адресной строке укажите: <https://localhost:1158/em/>

Во всех работах второй части используем пароль администратора, заданный при установке БД. В виртуальной машине, которую я делала это учетная запись SYS (режим входа SYSDBA) пароль Kb_2021.

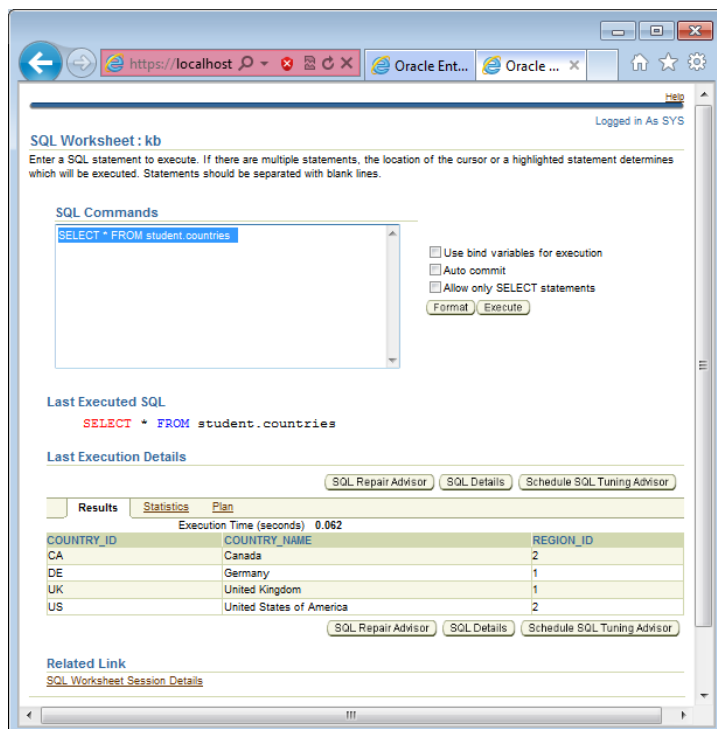
* User Name

* Password

Connect As



Найдите на странице «SQL Worksheet», запустите.
Попробуйте выполнить любую SQL-команду, например: `select * from ora1.countries` (префикс перед таблицей указывает ее владельца)

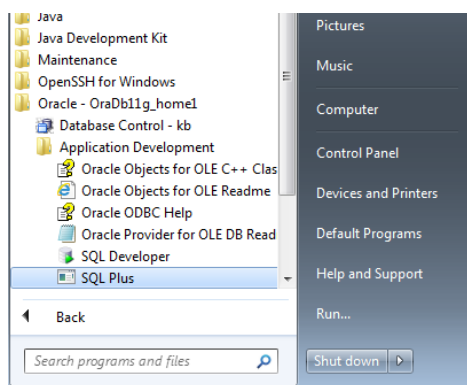


Доступ к базе данных для выполнения SQL-команд возможен также через «SQL Plus». SQL*Plus позволяет использовать интерфейс командной строки для написания команд «SQL Plus», SQL и PL/SQL. При этом можно:

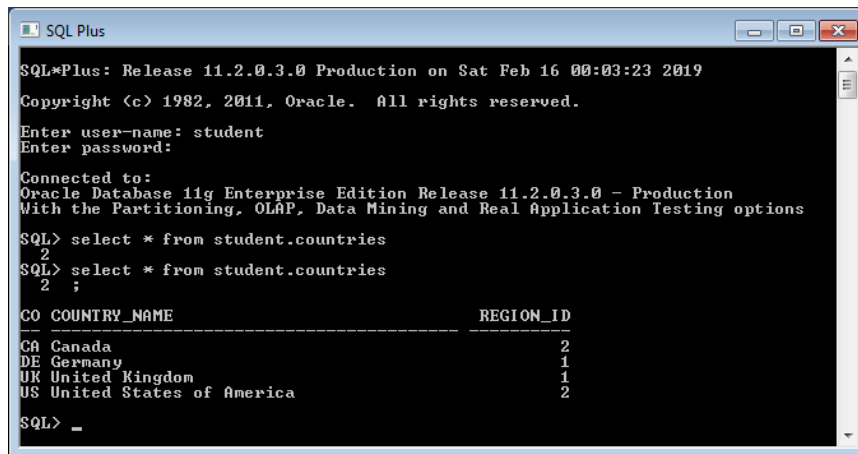
- вводить, редактировать, выполнять, хранить, выбирать и сохранять команды SQL и блоки PL/SQL;
- форматировать, подсчитывать, хранить и выводить результаты запросов;
- просматривать определения столбцов любой таблицы;
- посылать сообщения и принимать ответы от конечных пользователей;
- выполнять работы по администрированию базы данных.

Обычно программа используется для запуска готовых SQL-скриптов, присылаемых разработчиками для настройки базы данных, внесения изменений в схему и т.п.

Запустите «SQL Plus»:



Введите логин и пароль. Попробуйте выполнить любой запрос к базе данных.



```
SQL*Plus: Release 11.2.0.3.0 Production on Sat Feb 16 00:03:23 2019
Copyright (c) 1982, 2011, Oracle. All rights reserved.

Enter user-name: student
Enter password:

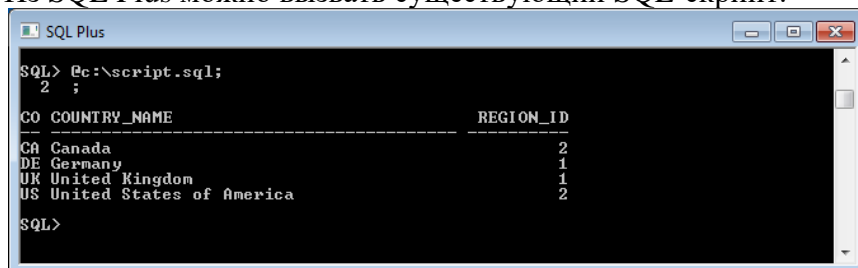
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select * from student.countries
2
SQL> select * from student.countries
2
;
```

CO	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

```
SQL> _
```

Из SQL Plus можно вызвать существующий SQL-скрипт.



```
SQL> @c:\script.sql;
2
;
```

CO	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

```
SQL>
```

Создайте скрипт-файл (расширение *.sql) с любым SQL-запросом и выполните его из SQL Plus.

SQL Plus может подключаться к базе данных без ввода имени учетной записи и пароля в случае локального подключения (с того же компьютера), при условии что пользователь ОС входит в группу OSDBA (для linux-систем) или группу ora_dba для Windows-систем.

Формат в режиме командной строки :

`sqlplus "/ as sysdba" ($ sqlplus / as sysdba для Linux)`

2) Файлы параметров инициализации.

При старте сервер Oracle читает файл параметров. Существует два типа файлов параметров инициализации:

- **Файл серверных параметров.** Этот вид файла предпочтительнее всего использовать. Он представляет собой двоичный файл, в который может писать и читать сервер базы данных. Его *нельзя редактировать вручную*. Файл размещается на сервере, на котором функционирует база данных Oracle. Его содержимое сохраняется между остановкой и запуском экземпляра БД. Такой файл часто называют *файлом серверных параметров (server parameter file, SPFILE)*. По умолчанию при старте экземпляра автоматически ищется файл с именем spfile<SID>.ora.

- **Текстовый файл параметров инициализации.** Сервер базы данных читает, но не пишет в файл такого вида. Значения параметров инициализации в этом файле должны задаваться и изменяться вручную с помощью текстового редактора. Только после таких изменений, сделанных вручную, модифицированные параметры инициализации сохраняют свое значение после остановки и повторного запуска экземпляра БД. По умолчанию при старте экземпляра, если не найден SPFILE, автоматически ищется файл с именем init<SID>.ora.

Рекомендуется создавать SPFILE, использование которого позволяет осуществлять динамическое управление параметрами инициализации. С помощью SPFILE можно постоянно хранить и сопровождать параметры инициализации, располагая их в файле на диске на стороне сервера.

Упрощение сопровождения параметров инициализации.

Параметры инициализации разделены на две группы: *базовые (basic)* и *дополнительные (advanced)*.

В большинстве случаев для достижения рациональной производительности требуется устанавливать и настраивать только 32 базовых параметра. В редких ситуациях может понадобиться изменить дополнительные параметры для достижения оптимальной производительности.

Параметр определяется как базовый исходя из того, что он с большой вероятностью будет устанавливаться для сохранения хорошей производительности функционирующей базы данных. Все остальные параметры рассматриваются как дополнительные.

В перечень базовых параметров, например, входят “месторасположения” или имена каталогов для хранения файлов определенного типа.

Примеры таких параметров: AUDIT_FILE_DEST, BACKGROUND_DUMP_DEST, CORE_DUMP_DEST, DB_CREATE_FILE_DEST, DB_CREATE_ONLINE_LOG_DEST_n, DB_RECOVERY_FILE_DEST и USER_DUMP_DEST.

Примеры параметров инициализации.

Параметр CONTROL_FILES задает одно или несколько имен управляющих файлов. Oracle настоятельно рекомендует мультиплексировать и зеркалировать управляющие файлы. Этот параметр может включать список, содержащий от 1 до 8 имен файлов (с именами путей). По умолчанию число файлов в списке зависит от ОС.

Параметр DB_BLOCK_SIZE задает размер (в байтах) блока базы данных Oracle. Его значение определяется при создании базы данных и не может быть впоследствии изменено. Диапазон значений: 1024 – 65536 (в зависимости от ОС). Значение по умолчанию: 8Кб (зависит от ОС).

Параметр DB_CACHE_SIZE определяет размер кэша буферов для блоков стандартного размера. Диапазон значений: по меньшей мере 16 Мб. Значение по умолчанию: 48 Мб.

Параметр DB_FILE_MULTIBLOCK_READ_COUNT задает максимальное количество блоков, читаемых за одну операцию ввода-вывода при полном последовательном просмотре. Диапазон значений зависит от ОС. Значение по умолчанию: 8.

Параметр DB_FILES определяет максимальное количество файлов базы данных, которые могут быть открыты для данной базы данных. Диапазон значений: максимальное значение зависит от ОС. Значение по умолчанию: зависит от ОС (200 для Solaris).

Параметр PGA_AGGREGATE_TARGET определяет совокупный размер памяти, выделяемой для *программных глобальных областей (Program Global Area, PGA)* всех серверных процессов, связанных с экземпляром. Перед включением автоматической настройки рабочих областей (параметр workarea_size_policy=auto) для этого параметра задается положительное значение. Выделяемое пространство не располагается в *системной глобальной области (System Global Area, SGA)*. База данных использует этот параметр в качестве целевого размера памяти, используемой для всех PGA. Для того, чтобы правильно установить этот параметр, сначала вычтете размер SGA из общего размера памяти, доступной в системе для экземпляра Oracle. Размер оставшегося пространства может быть назначен в качестве значения параметра PGA_AGGREGATE_TARGET. Диапазон значений: от 10 Мб до 400 Гб. Значение задается целым числом, за которым следует символ К, М или Г (килобайты, мегабайты или гигабайты). Значение по умолчанию не определено, что означает полное отключение автоматической настройки рабочих областей.

Параметр PROCESSES задает максимальное количество процессов в ОС, выполняющих действия пользователей, которые могут быть одновременно подсоединены к серверу Oracle. Это значение должно учитывать все фоновые процессы. Диапазон значений: от 6 до величины, зависящей от ОС. Значение по умолчанию: 40 или более в зависимости от ОС.

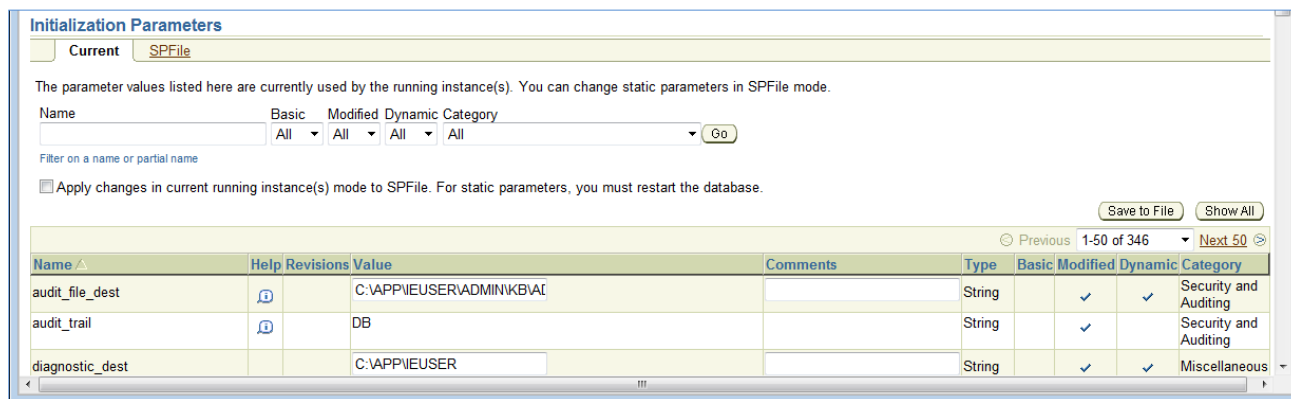
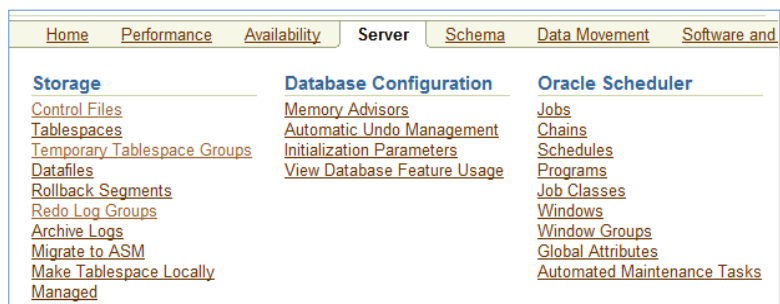
Параметр SHARED_POOL_SIZE задает размер разделяемого пула в байтах. Разделяемый пул содержит, например, следующие объекты: разделяемые курсоры, хранимые процедуры, структуры управления, буферы сообщений, используемые при параллельном

выполнении. Увеличение размера этого буфера может повысить производительность многопользовательских систем. Диапазон значений: минимальный размер – гранула (см. V\$SGAINFO) . Значение по умолчанию: для 64-битных систем 84 Мб, для 32-битных систем 32 Мб.

Параметр UNDO_MANAGEMENT определяет режим управления пространством отмены. Если значение AUTO, тогда экземпляр запускается в *системном режиме управления информацией отмены (System Managed Undo (SMU) mode)*. В противном случае экземпляр стартует в ручном режиме использования сегментов отката (*Rollback Undo (RBU) mode*). Внешне пространство для информации отмены выделяется в режиме RBU в виде сегментов отката, а в режиме SMU – в виде табличных пространств типа UNDO. Допустимые значения: AUTO или MANUAL. По умолчанию, если параметр UNDO_MANAGEMENT не задан при запуске первого экземпляра, устанавливается значение MANUAL, и экземпляр стартует в режиме RBU. Не первый экземпляр запускается в том же режиме, в котором работают другие существующие экземпляры.

Просмотр и изменение параметров инициализации.

Параметры инициализации можно просмотреть и изменить с помощью Enterprise Manager. Для этого на странице Database Administration щелкните на ссылке Initialization Parameters, расположенной в секции Database Configuration.



В ЕМ найдите параметр, указывающий место расположения контрольных файлов учебной базы данных.

В SQL Plus отобразите значения параметров SGA_MAX_SIZE, DB_CACHE_SIZE и SHARED_POOL_SIZE. Используйте команду SHOW PARAMETER.

Измените значение параметра JOB_QUEUE_PROCESSES на 1000 и в памяти экземпляра (MEMORY) и в файле параметров (SPFILE).

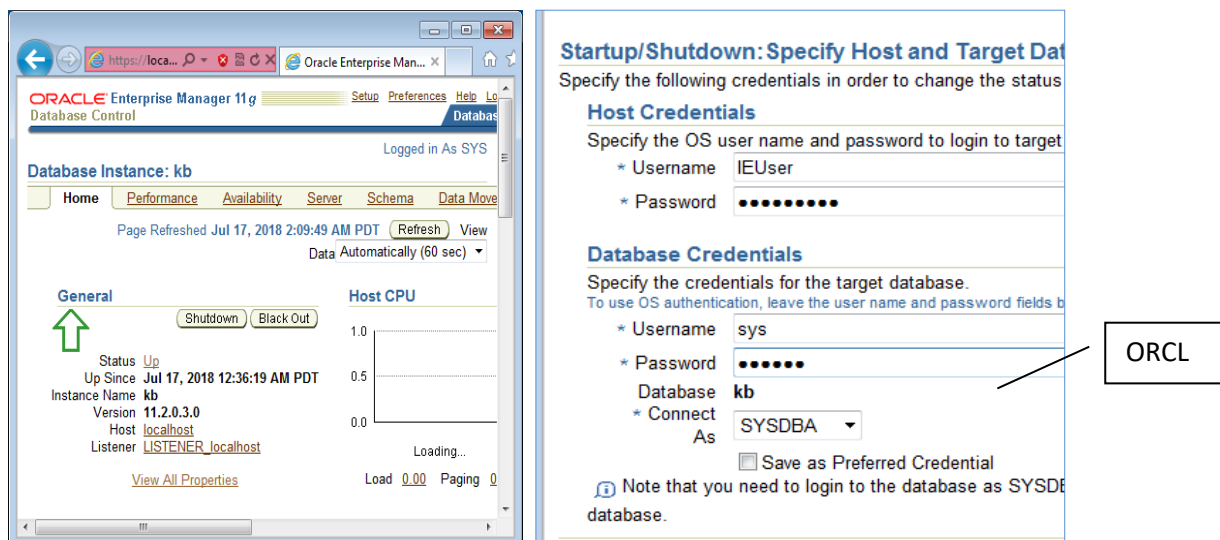
Подсказка: ALTER SYSTEM SET job_queue_processes=1000 SCOPE=BOTH;

3) Запуск и остановка базы данных.

Запуск и остановку базы данных можно также выполнить из Oracle Enterprise Manager.

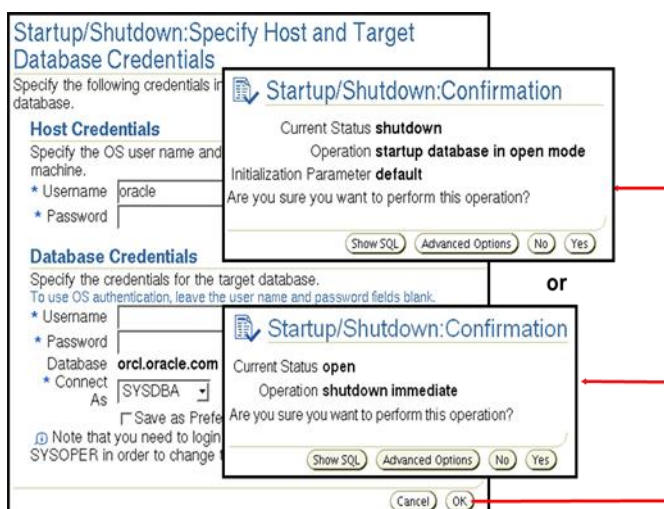
После щелчка на кнопке Startup или Shutdown у вас запрашиваются имена и пароли для соединения с хостом (компьютером, на котором находится БД) и с самой базой данных.

После этого можно щелкнуть на кнопке Advanced Options для того, чтобы изменить любые опции запуска или режим остановки. Кроме того, по кнопке Show SQL можно вывести команду SQL, используемую для старта или остановки.

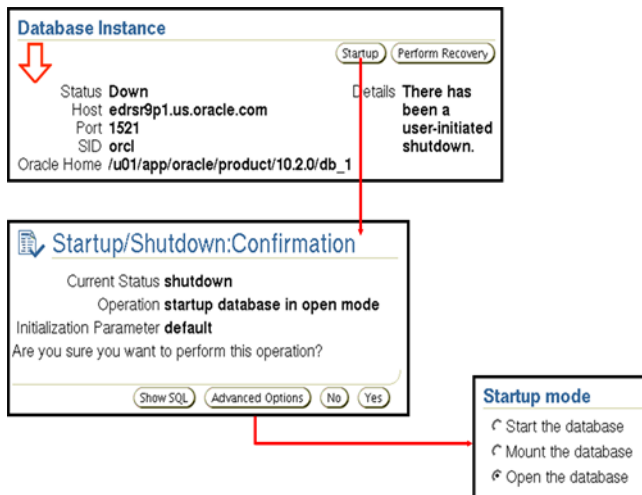


После этого можно щелкнуть на кнопке Advanced Options для того, чтобы изменить любые опции запуска или режим остановки. Кроме того, по кнопке Show SQL можно вывести команду SQL, используемую для старта или остановки.

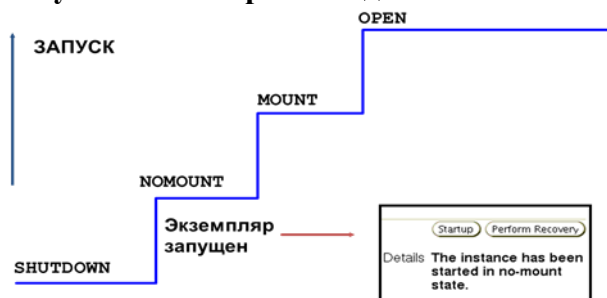
После этого можно щелкнуть на кнопке Advanced Options для того, чтобы изменить любые опции запуска или режим остановки. Кроме того, по кнопке Show SQL можно вывести команду SQL, используемую для старта или остановки.



Если база данных еще не запущена, это отображается на странице, выводимой после подсоединения к Enterprise Manager Database Control. Чтобы запустить экземпляр БД, щелкните на кнопке Startup. Введите имена и пароли для соединения с хостом и базой данных. Дополнительно можете выбрать режим запуска.



Запуск экземпляра базы данных: состояние NOMOUNT



Во время запуска экземпляра базы данных необходимо выбрать состояние, в которое в результате перейдет экземпляр. Следующие сценарии описывают различные этапы запуска экземпляра.

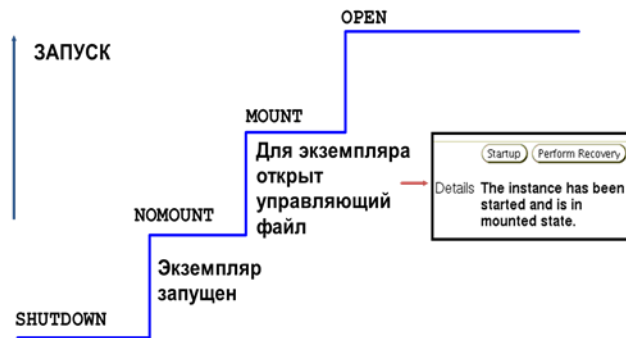
Обычно экземпляр запускается в режиме NOMOUNT только во время создания базы данных или для пересоздания управляющих файлов, а также при выполнении определенных сценариев резервирования и восстановления.

Запуск экземпляра подразумевает выполнение следующих задач:

- поиск в директории \$ORACLE_HOME/dbs файла параметров, осуществляемый в следующем порядке:
 - ищется файл spfileSID.ora
 - если он не найден, тогда производится поиск файла spfile.ora ;
 - если он не найден, тогда поиск файла initSID.ora, который содержит параметры экземпляра базы данных. Задание параметра PFILE в команде STARTUP переопределяет установленный по умолчанию порядок выбора файла параметров;
- выделение SGA;
- запуск фоновых процессов;
- открытие сигнального файла alertSID.log и файлов трассировки.

Примечание: SID – системный идентификатор экземпляра (например, на учебной виртуальной машине - ORCL).

Запуск экземпляра базы данных: состояние MOUNT



Монтирование базы данных включает следующие задачи:

- ассоциация базы данных с предварительно запущенным экземпляром;
- определение местоположения управляющих файлов, которые указаны в файле параметров;
- чтение управляющих файлов с целью получения имен и статуса файлов данных и журнальных файлов. Однако на данный момент не проверяется фактическое существование файлов данных и журнальных файлов.

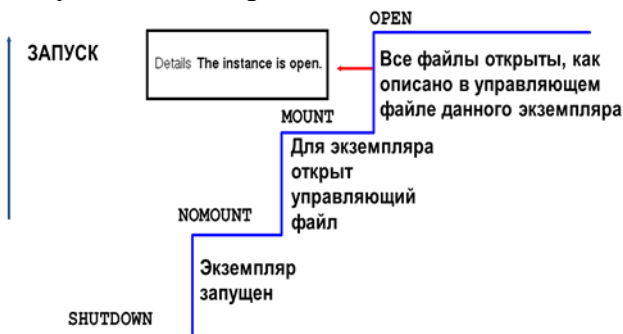
Запуск экземпляра и монтирование базы данных без ее открытия выполняется в целях осуществления особых эксплуатационных операций.

Например, база данных должна быть смонтирована, но не открыта, при выполнении следующих задач:

- переименование файлов данных (файлы данных автономного табличного пространства могут быть переименованы, когда база данных открыта);
- включение и отключение режима архивирования журнальных файлов;
- полное восстановление базы данных.

Примечание. База данных может остаться в состоянии MOUNT, даже если была выполнена попытка ее перевода в состояние OPEN. Это может произойти, когда требуется восстановление базы данных (его можно выполнить разными способами).

Запуск экземпляра базы данных: состояние OPEN



Нормальное функционирование базы данных подразумевает то, что экземпляр запущен, а база данных смонтирована и открыта. В этом состоянии любой зарегистрированный пользователь может соединиться с БД для выполнения операций с данными.

Открытие базы данных подразумевает выполнение следующих задач:

- открытие оперативных файлов данных;
- открытие оперативных журнальных файлов.

Если какие-либо из файлов данных или журнальных файлов недоступны в момент открытия базы данных, сервер Oracle возвращает ошибку.

При выполнении окончательного этапа открытия базы данных Oracle проверяет доступность всех файлов данных и журнальных файлов, а также проверяет целостность базы

данных. Если необходимо, фоновый процесс системный монитор (SMON) инициирует восстановление экземпляра.

Экземпляр БД можно запустить в режиме ограниченного доступа. В этом случае база данных доступна только для пользователей с административными привилегиями. Чтобы запустить экземпляр в таком режиме, выберите опцию “Restrict access to database” на странице Advanced Startup Options.

Остановка экземпляра базы данных Oracle

В случае, когда экземпляр уже запущен и его необходимо остановить, подключитесь к Enterprise Manager Database Control и щелкните на кнопке Shutdown. После этого, если щелкнуть на кнопке Advanced Options, можно выбрать режим остановки (Normal, Transactional, Immediate или Abort).



Режим остановки	A = ABORT (аварийный)	I = IMMEDIATE (немедленный)	T = TRANSACTIONAL (транзакционный)	N = NORMAL (нормальный)
Разрешить новые соединения	нет	нет	нет	нет
Ждать конца текущего сеанса	нет	нет	нет	да
Ждать конца текущей транзакции	нет	нет	да	да
Контрольная точка, закрытие файлов	нет	да	да	да

Режимы остановки можно расположить в следующем порядке, отражающем постепенное изменение выполняемых действий, необходимость которых зависит от текущего функционирования.

- ABORT

Перед остановкой производится наименьшее число действий. После этого при запуске система должна выполнить восстановление. Поэтому используйте этот режим только, когда это необходимо. Обычно он применяется, когда другие варианты остановки не отрабатывают, когда это вызвано проблемами, возникающими при запуске или когда требуется немедленно оставить экземпляр перед проблемной ситуацией, например, при получении сообщения о том, что через несколько секунд будет выключение питания.

- IMMEDIATE

Обычно используемая опция. При этом незафиксированные транзакции откатываются.

- TRANSACTIONAL

Предоставляется возможность завершить транзакции.

- NORMAL

Экземпляр не останавливается, пока не отсоединятся сеансы.

Наименьшее время остановки в режиме ABORT и наибольшее – в NORMAL.

Нормальная остановка (SHUTDOWN NORMAL).

Нормальный режим остановки базы данных используется по умолчанию. Нормальная остановка базы данных происходит следующим образом:

- Новые соединения не разрешаются.
- Сервер Oracle ожидает отсоединения всех пользователей и только после этого продолжает остановку базы данных.
- Буферы из кэша базы данных и журнала записываются на диск.
- Фоновые процессы завершаются и SGA удаляется из памяти.
- Перед остановкой экземпляра Oracle закрывает и демонтирует базу данных.
- При следующем запуске не потребуется восстановления экземпляра.

Транзакционная остановка (SHUTDOWN TRANSACTIONAL)

Транзакционная остановка обеспечивает сохранность данных клиентов, включая результаты текущих действий. Остановка базы данных в транзакционном режиме происходит следующим образом:

- Ни один клиент не может запустить новую транзакцию в этом экземпляре.
- Клиент принудительно отсоединяется, как только завершается текущая транзакция.
- Как только все транзакции завершены, немедленно выполняется остановка.
- При следующем запуске не потребуется восстановления экземпляра.

Немедленная остановка (SHUTDOWN IMMEDIATE)

Немедленная остановка базы данных выполняется следующим образом:

- Обработка команд SQL, выполняемых Oracle в данный момент, не завершается.
- Сервер Oracle не ожидает отсоединения пользователей, работающих с базой данных в текущий момент.
- Oracle выполняет откат всех активных транзакций и принудительно отсоединяет всех пользователей.
- Oracle закрывает и демонтирует базу данных перед остановкой экземпляра.
- При следующем запуске не потребуется восстановления экземпляра.

Аварийная остановка (SHUTDOWN ABORT)

Если режимы нормальной и немедленной остановки не срабатывают, может быть выполнена аварийная остановка базы данных. Аварийное завершение работы экземпляра выполняется следующим образом:

- Немедленно отменяются все команды SQL, обрабатываемые сервером Oracle.
- Сервер Oracle не ожидает отсоединения пользователей, работающих с базой данных в текущий момент.
- Буферы из кэша базы данных и журнала не записываются на диск.
- Не выполняется откат незафиксированных транзакций.
- База данных не закрывается и не демонтируется
- Экземпляр удаляется без закрытия файлов.
- При следующем запуске потребуется восстановление экземпляра, которое произойдет автоматически.

Примечание: не рекомендуется резервировать базу данных, остановленную в несогласованном состоянии.

В EM выполните остановку базы данных в режиме «Normal», затем запустите базу данных. Просмотрите SQL-команду для выполнения запуска и остановки.

Выполните те же действия из SQL Plus (или SQL-developer).

Формат SQL-команд:

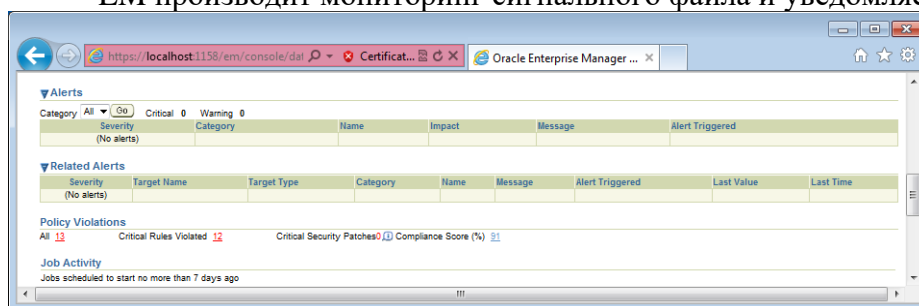
```
SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]
STARTUP [FORCE] [RESTRICT] [MOUNT | OPEN | NOMOUNT]
```

4) Просмотр сигнального файла.

У каждой базы данных есть файл `alert_sid.log`. Он располагается на сервере базы данных и хранится в каталоге, задаваемом параметром инициализации `background_dump_dest`. В сигнальном файле БД содержится хронологический журнал сообщений и ошибок, включая следующее:

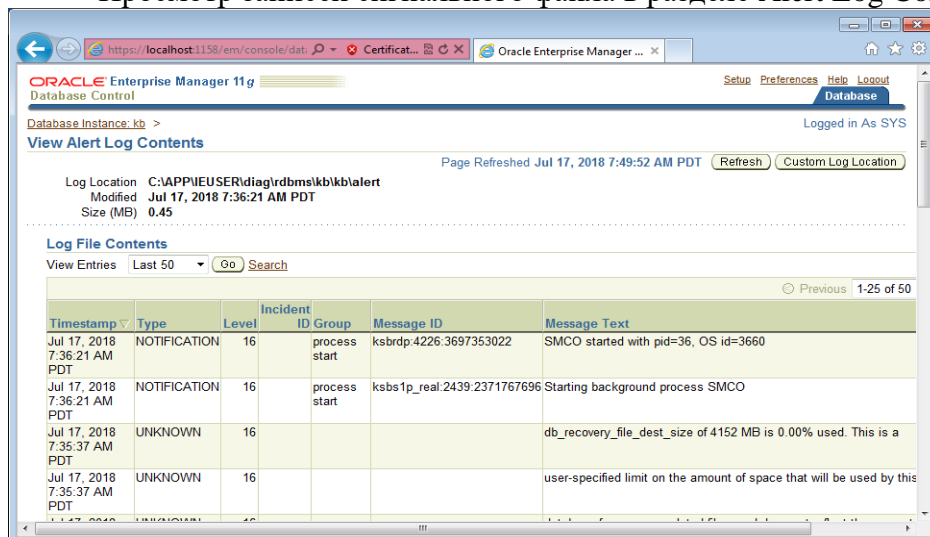
- значения параметров инициализации, отличные от установленных по умолчанию, которые использовались при запуске экземпляра БД;
- все внутренние ошибки (ORA-600), ошибки повреждения блоков (ORA-1578), и ошибки взаимоблокировки (ORA-60);
- сообщения об административных операциях, выполняемых по командам SQL CREATE, ALTER, DROP DATABASE, TABLESPACE и с помощью Enterprise Manager; также содержатся записи о выполнении команд SQL Plus STARTUP, SHUTDOWN, ARCHIVE LOG и RECOVER;
- некоторые сообщения и ошибки, связанные с работой разделяемых серверных и диспетчерских процессов;
- ошибки, возникающие при автоматическом обновлении материализованного представления.

ЕМ производит мониторинг сигнального файла и уведомляет о критических ошибках.



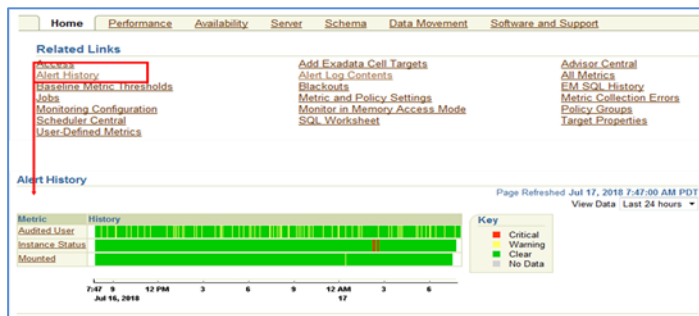
Сигнальный файл можно просматривать для выявления других ошибок и информационных сообщений. Этот файл может вырасти до больших размеров. Поэтому его следует время от времени резервировать и удалять. Если во время попытки записи БД не находит сигнальный файл, она его пересоздает.

Просмотр записей сигнального файла в разделе Alert Log Contents.



Просмотр хронологии сигнальных сообщений.

На странице Alert History выводится диаграмма, отражающая хронологию сигнальных сообщений базы данных в выбранном отрезке времени. Сигнальные сообщения указывают на потенциальные проблемы, которые проявляются либо при выходе значений наблюдаемых метрик за допустимые диапазоны, либо, когда наблюдаемая цель больше недоступна.



На странице Alert History выводится диаграмма, отражающая хронологию сигнальных сообщений базы данных в выбранном вами отрезке времени. Сигнальные сообщения указывают на потенциальные проблемы, которые проявляются либо при выходе значений наблюдаемых метрик за допустимые диапазоны, либо, когда наблюдаемая цель больше недоступна.

В ЕМ просмотрите записи Alert Log Contents, а также диаграммы Alert History, уточните сведения о каком-нибудь критическом событии (выбрав мышкой соответствующую метку).

5) Динамические представления производительности

Oracle хранит и поддерживает большое множество динамических данных об операциях и производительности экземпляра БД. Эти динамические представления производительности основываются на виртуальных таблицах, которые строятся из структур памяти внутри сервера БД. Такие таблицы не являются обычными таблицами, располагаемыми внутри базы данных. Поэтому некоторые из них можно просмотреть перед тем, как база данных будет смонтирована или открыта.

Динамические представления производительности содержат информацию о:

- сеансах;
- состоянии файлов;
- продвижении заданий и задач;
- блокировках;
- статусе резервных объектов;
- распределении и использовании памяти;
- параметрах системы и сеанса;
- выполнении кода SQL;
- статистиках и метриках.

Примечание: представления DICTIONARY и DICT_COLUMNS содержат имена динамических представлений производительности и их столбцов.

Наиболее часто эти представления использует Enterprise Manager. Однако пользователи также могут выполнять запросы к этим представлениям, когда это необходимо.

Примеры:

а) Какие команды SQL потребляли при своем выполнении более 200000 микросекунд времени ЦП и сколько раз они выполнялись?

```
SQL> SELECT sql_text, executions FROM v_$sql WHERE cpu_time > 200000;
```

б) Какие из существующих сеансов были установлены в течение прошедшего дня с компьютера EDRSR9P1?

```
SQL> SELECT * FROM v_$session WHERE machine = 'EDRSR9P1' and logon_time > SYSDATE - 1;
```

в) Какие сеансы и как долго удерживают блокировку, требуемую другому сеансу?

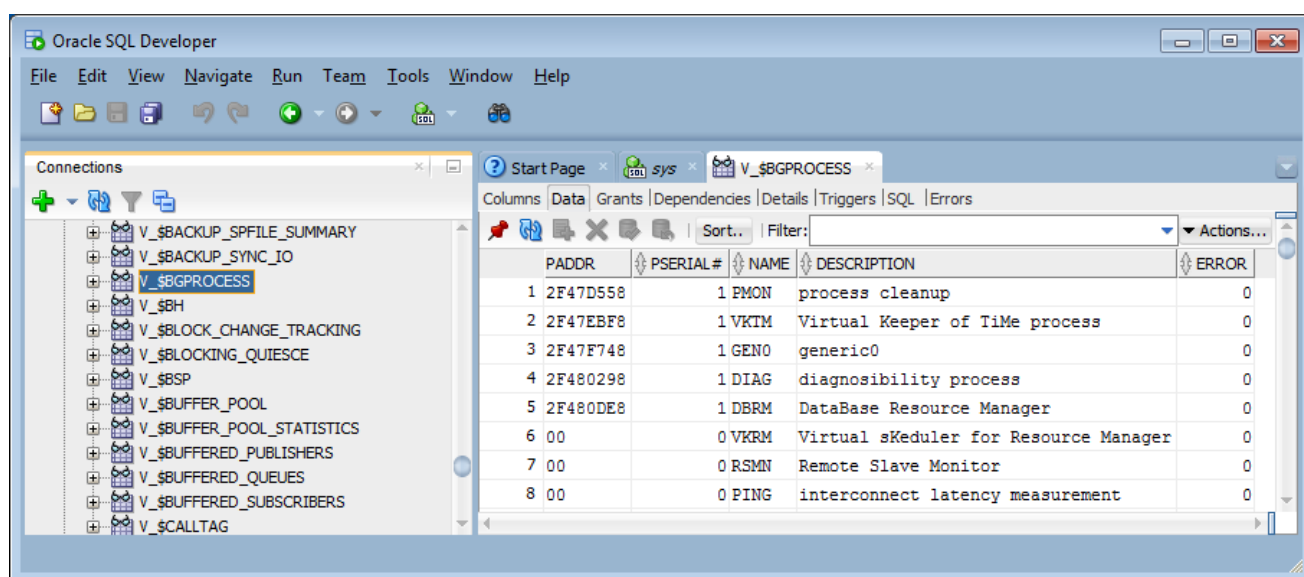
```
SQL> SELECT sid, ctime FROM v_$lock WHERE block > 0;
```

Общая характеристика динамических представлений производительности:

- Владелец этих представлений – пользователь SYS.
- Различные представления доступны после:
 - запуска экземпляра
 - монтирования базы данных
 - открытия базы данных
- Для просмотра имен представлений можно выполнить запрос к V_\$FIXED_TABLE.
- Эти представления часто называют «V_\$-представлениями»
- Данные таких представлений меняются динамически, поэтому целостность чтения не гарантируется.

Некоторые представления содержат данные, которые недоступны во всех состояниях экземпляра и базы данных. Например, когда экземпляр запущен, но база данных еще не смонтирована, можно запросить из представления V_\$BGPROCESS сведения о работающих фоновых процессах. Однако получить данные о статусе файлов из представления базы данных V_\$DATAFILE невозможно, поскольку нельзя прочитать эти сведения из управляющего файла, который становится доступным только после монтирования базы данных.

В SQL Developer найдите и просмотрите представления V_\$SQL, V_\$LOCK, V_\$SESSION, V_\$BGPROCESS, V_\$DATAFILE, V_\$FIXED_TABLE. Просмотрите их описание и содержимое.



Время на выполнение лабораторной работы – 2 часа.