Вопросы по дисциплине СУБД Oracle (15/16) уч. год.

<http://bourabai.ru/dbt/servers/oracle2.htm> хорошая ссылка

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

(C) Wikipedia:

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

Системы управления базами данных позволяют объединять большие объемы информации и обрабатывать их, сортировать, делать выборки по определённым критериям и т.п.

Современные СУБД дают возможность включать в них не только текстовую и графическую информацию, но и звуковые фрагменты и даже видеоклипы.

Простота использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями.

СУБД обеспечивают правильность, полноту и непротиворечивость данных, а также удобный доступ к ним.

**Основные функции СУБД:**

● Управление данными на дисках.

● Управление данными в ОП(оперативной памяти) (в т.ч., кэширование).

● Журналирование изменений, резервное копирование, восстановление после сбоев.

● Поддержка языков БД (DML + DDL).

DDL – Data Definition Language (язык описания данных)

DML – Data Manipulation Language (язык манипулирования данными)

DDL и DML — подмножество языка SQL:

Язык DDL служит для создания и модификации структуры БД, т.е. для создания/изменения/удаления таблиц и связей.

Функции языков DDL определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL это глаголы — «create» («создать»), «alter» («изменить»), «drop» («удалить»). Эти запросы или команды часто смешиваются с другими командами SQL, в связи с чем DDL не является отдельным компьютерным языком.

Язык DML позволяет осуществлять манипуляции с данными таблиц, т.е. с ее строками. Он позволяет делать выборку данных из таблиц, добавлять новые данные в таблицы, а так же обновлять и удалять существующие данные.

Язык DML содержит следующие конструкции:

SELECT – выборка данных

INSERT – вставка новых данных

UPDATE – обновление данных

DELETE – удаление данных

MERGE – слияние данных

<http://citforum.ru/database/osbd/glava_6.shtml> (ссылка про функции СУБД)

**Состав СУБД**

● Ядро — отвечает за управление данными и журнализацию.

● Процессор языка БД — транслирует запросы с высокоуровневого языка на низкоуровневый.

● Подсистема поддержки времени исполнения - интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД

● Сервисные программы (внешние утилиты)

**Категории 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; (все части локальной СУБД размещаются на одном компьютере)

● распределённые. (части СУБД могут размещаться на двух и более компьютерах).

*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 ANSI-SPARC**

Предложена в 1975 г. подкомитетом SPARC (Standards Planning And Requirements Commitee) ANSI.

*Архитектура СУБД включает в себя 3 уровня:*

● Внешний (пользовательский).

● Промежуточный (концептуальный).

● Внутренний (физический).

В основе архитектуры ANSI-SPARC лежит концептуальный уровень. В современных СУБД он может быть реализован при помощи представления. 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.

Почти все современные СУБД соответствуют принципам ANSI-SPARC.

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

Oracle по классификации — объектно-реляционная распределённая клиент-серверная СУБД.

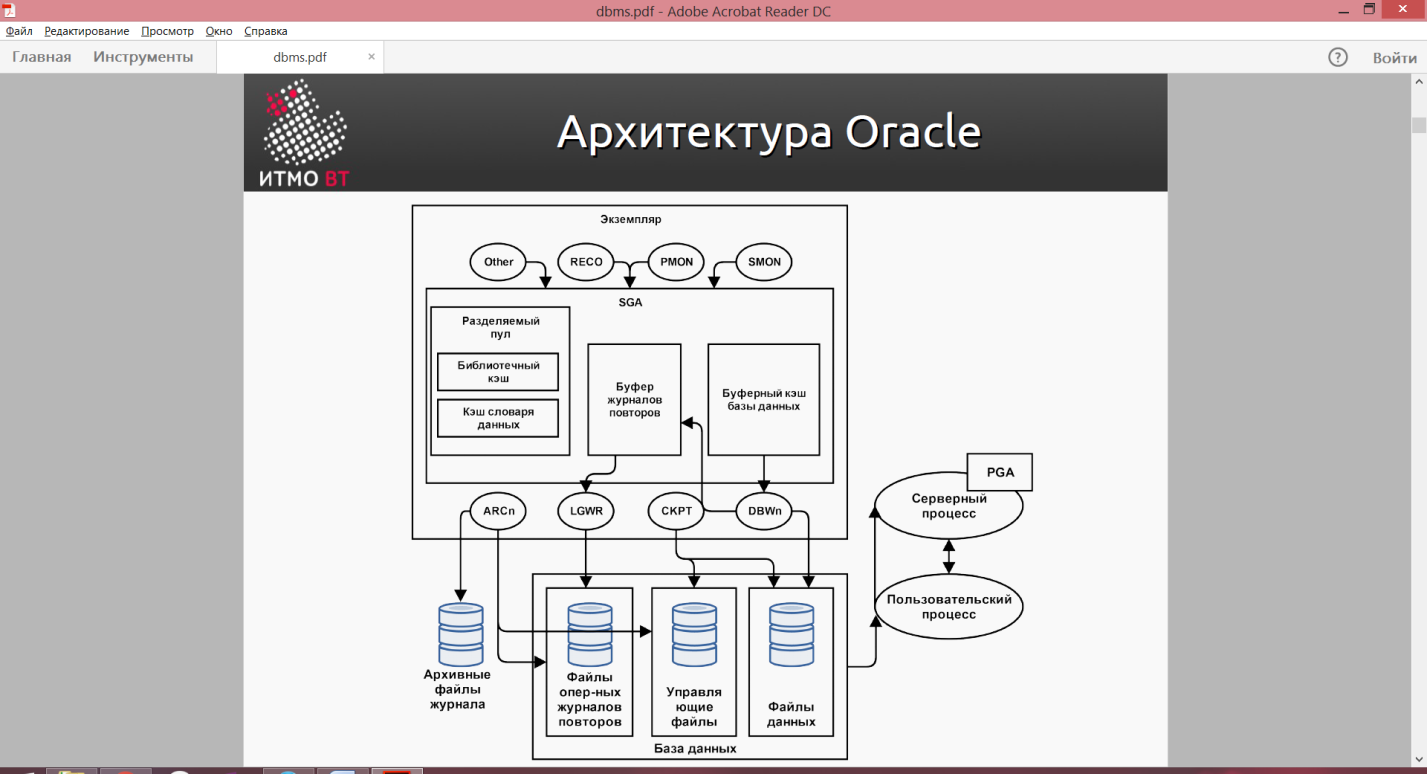
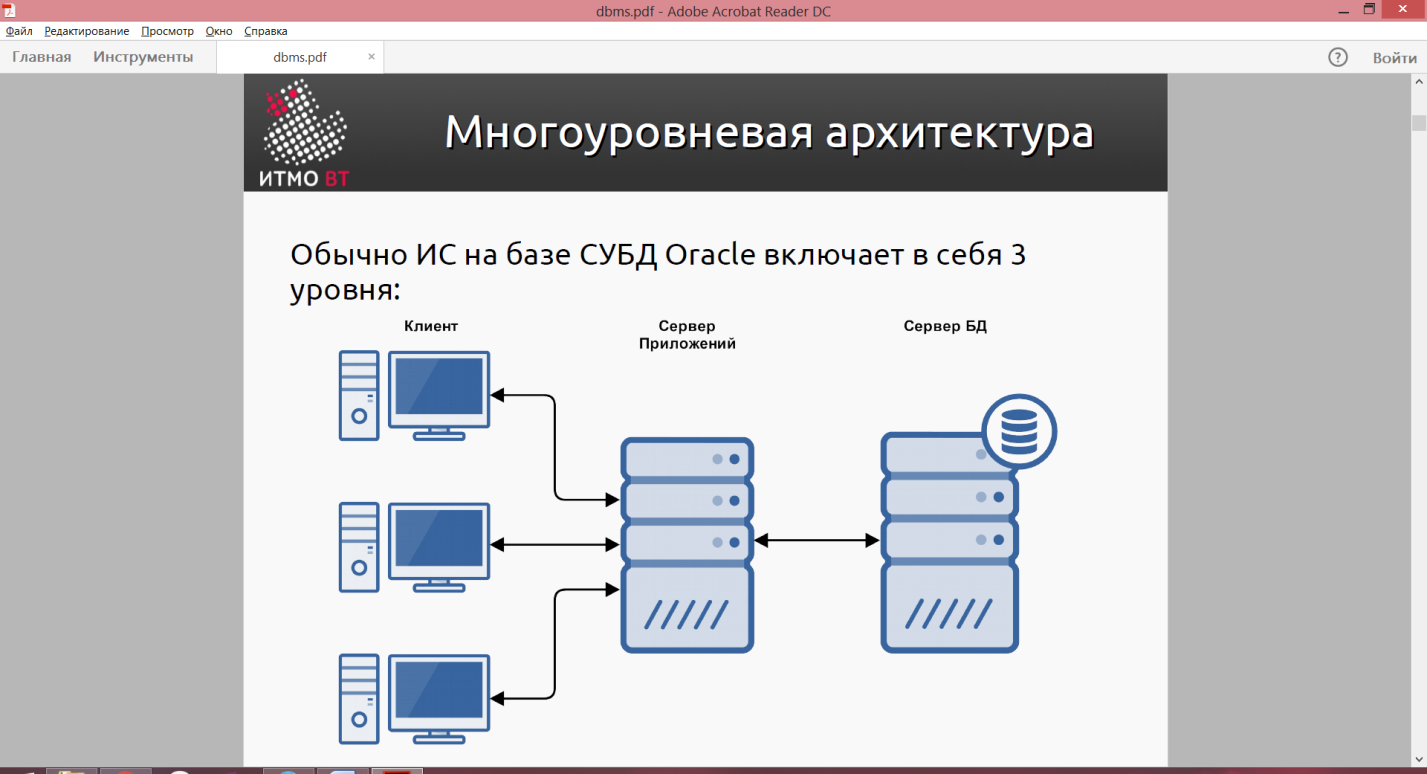
База данных Oracle состоит из экземпляра и связанных с ним базами данных. Экземпляр состоит из структур памяти и фоновых процессов. При каждом запуске экземпляра выделяется совместно используемая область памяти, которая называется глобальной системной областью (System Global Area, SGA), а так же запускаются 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.

Подключения и сеансы тесно связаны с пользовательскими процессами, но при этом они имеют весьма различные значения.

Подключение–это канал связи между пользовательским процессом и экземпляром базы данных Oracle. Канал связи устанавливается с помощью доступного механизма межпроцессного взаимодействия (на компьютере, на котором запущены как пользовательский процесс, так и база данных Oracle) или же с помощью сетевого программного обеспечения (когда на разных компьютерах, взаимодействующих через сеть, запущено приложение базы данных и база данных Oracle).

Сеанс представляет состояние подключения текущего пользователя к экземпляру базы данных. Например, когда пользователь запускает приложение SQL\*Plus, он должен предоставить действительное имя пользователя и пароль, после чего для данного пользователя будет открыт сеанс. Сеанс длится с момента подключения пользователя и до момента отключения или выхода из приложения базы данных.



**3. Структура памяти БД Oracle.**

База данных Oracle создает и использует структуры памяти для различных целей. Например, в памяти хранится код выполняемой программы, данные, разделяемые среди пользователей, и частные области данных для каждого подключенного пользователя.

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.

**- Глобальная программная область(PGA):** Области памяти, которые содержат данные и управляющую информацию для фонового или серверного процесса.

**Буферный кэш БД (Cache Buffer)**

● Является частью SGA.

● Хранит копии блоков данных, считанных их файлов данных.

● Если нужного блока данных нет в кэше, он читается с диска и помещается в кэш.

● Совместно используется всеми параллельно работающими пользователями.

● Управляется сложным алгоритмом, основанным на LRU.

Least recently used (LRU): в первую очередь, вытесняется неиспользованный дольше всех.

*Каждый буфер может находиться в одном из четырёх возможных*

*состояний:*

● Закрепленный (за сеансом). Другим сеансам запрещено одновременно выполнять запись в один и тот же блок — они должны ждать, пока он освободится.

● Очищенный. Буфер является незакрепленным и становится кандидатом на немедленное устаревание, если его текущее содержимое (блок данных) не будет запрошено еще раз.

● Свободный / неиспользуемый. Буфер пуст, так как экземпляр был только что запущен. Отличие от очищенного в том, что буфер еще не использовался.

● Заполненный. Буфер больше не закреплен, однако его содержимое (блок данных) изменилось, и процессу DBWn необходимо сбросить его на диск, прежде буфер можно будет объявить устаревшим.

**Буфер журнала повторов (Redo Log Buffer)**

Циклический буфер в SGA.

● Хранит информацию об изменениях в БД.

● Содержит записи повторов, в которых хранится информация для повторного применения изменений, внесенных операциями DML и DDL.

● Записи повторов используются для восстановления базы данных в случае необходимости.

● Фоновый процесс LGWR производит запись буфера журнала повторов на диск.

**Разделяемый пул (Shared Pool)**

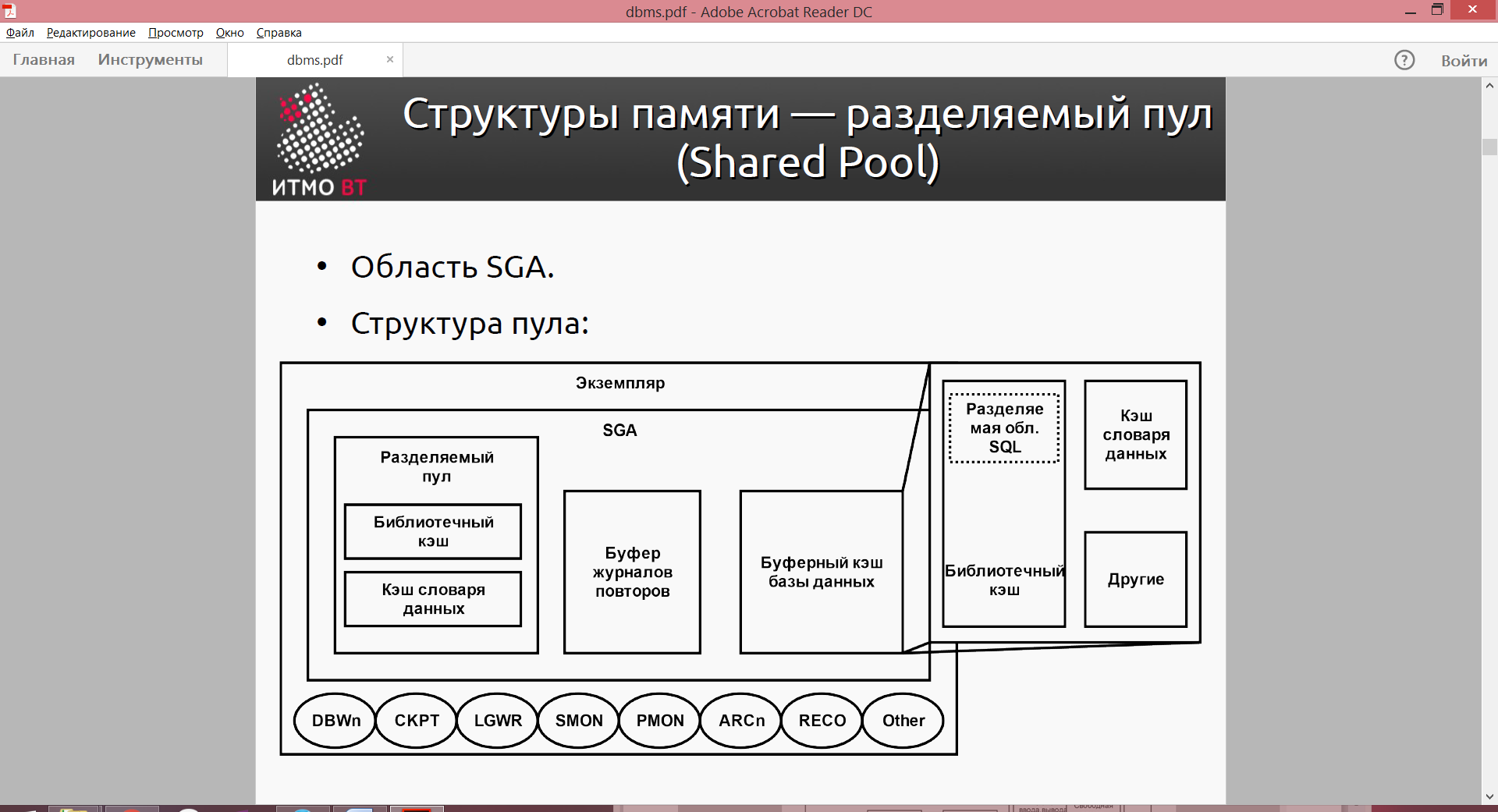
● Данные вытесняются из пула по алгоритму LRU.

● 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 (Large Pool)**

● Необязательная область SGA.

● Выделяется вручную администратором БД.

● В отличие от разделяемого пула, нет автоматического освобождения памяти по LRU.

● Может быть использован:

● Для операций передачи большого объёма данных между разными БД.

● Для операций резервного копирования /восстановления.

● Размер задаётся параметром инициализации LARGE\_POOL\_SIZE.



**4. Архитектура процессов.**

**2 вида процессов:**

● Пользовательские процессы. Запускаются в момент подключения пользователя к БД.

● Процессы базы данных:

● Серверный процесс: подключается к экземпляру Oracle и запускается при установлении сеанса пользователем.

● Фоновые процессы: запускаются при запуске экземпляра Oracle.

Когда пользователь запускает приложение или инструмент Oracle, например SQL\*Plus, база данных Oracle создает пользовательский процесс для выполнения приложения пользователя. База данных Oracle также создает серверный процесс для выполнения команд пользовательского процесса. Кроме того, сервер Oracle так же порождает набор фоновых процессов для экземпляра, которые взаимодействуют друг с другом и с операционной системой, управляя структурами памяти, выполняя асинхронные операции ввода/вывода для записи данных на диск и другие задачи.

Структура процессов различна для каждой конфигурации базы данных Oracle в зависимости от используемой операционной системы и настроек базы данных Oracle. Код для подключенных пользователей можно настроить как выделенный или разделяемый сервер.

**Выделенный сервер (Dedicated Server)**

В режиме выделенного сервера каждому пользовательскому процессу создаётся свой «персональный» серверный.

**Разделяемый сервер (Shared Server)**

В режиме разделяемого сервера каждому серверные процессы выделяются диспетчером из специального пула.



**Процесс системного монитора**

**SMON – System monitor**

● Выполняет восстановление при запуске экземпляра (если в этом есть необходимость).

● Выполняет очистку временных сегментов, которые больше не используются.

● Если во время восстановления экземпляра какая- либо из завершенных транзакций была пропущена (из-за ошибки чтения файла или ошибки автономного режима), SMON восстановит их при переводе табличного пространства или файла обратно в оперативный режим.

**Процесс монитора процессов**

**PMON – Process monitor**

● Выполняет восстановление пользовательского процесса при сбое:

● Выполняет очистку буферного кэша базы данных.

● Освобождает ресурсы, использовавшиеся пользовательским процессом.

● Выполняет мониторинг времени ожидания при отсутствии действий в сеансах.

● Динамически регистрирует службы базы данных в процессах-слушателях (Network Listeners).

● Периодически проверяет состояние процессов диспетчера и сервера и перезапускает любой из этих процессов в случае его остановки.

**Процесс восстановления**

**RECO – Recovery**

● Используется при распределенной конфигурации БД.

● Автоматически подключается к другим БД, задействованным в сомнительных распределенных транзакциях.

● Автоматически разрешает все сомнительные транзакции.

● Удаляет все строки, соответствующие сомнительным транзакциям.

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

**Процесс записи в БД**

**DBWn** (Database writer)

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.

● Изменения записываются в файлы в том порядке, в котором они были сделаны (согласно **SCN — System Change Number**).

● Для этого используется LRUW (LRU-Write) — список заполненных буферов в кэше, отсортированный по SCN.

● При записи данных в файл DBW одновременно «перемещает» указатель на контрольную точку, с которой будет начато восстановление в случае сбоя (инкрементальная установка контрольных точек).

Немного другими словами:

Процесс записи в базу данных (DBWn) записывает содержимое буферов в файлы данных. Процессы DBWn осуществляют запись измененных (заполненных) буферов из буферного кэша базы данных на диск.

Если буфер в буферном кэше базы данных изменяется, он помечается как заполненный и добавляется в список LRUW (LRU-запись) заполненных буферов, отсортированный по номеру SCN. Когда количество доступных буферов в буферном кэше становится меньше внутреннего порогового значения (при котором серверным процессам становится тяжело получать доступные буферы), DBWn записывает заполненные буферы в файлы данных в порядке их изменения, следуя порядку в списке LRUW.

SGA содержит структуру памяти, в которой находится журнальный байтовый адрес (RBA) позиции в потоке повторов, с которой следует начать восстановление в случае отказа экземпляра.

**6. Запись журнала повторов, логика работы процесса записи.**

**Процесс записи в журнал LGWR – Log Writer**

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(заполненные) 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.

● Это называется механизмом быстрой фиксации транзакции.

**7. Создание контрольной точки, процесс архивирования журнала повторов.**

**Процесс создания контрольной точки**

**Checkpoint – CKPT**

● Контрольная точка – это структура данных, определяющая SCN в потоке повторов базы данных.

● Контрольные точки записываются в управляющий файл и в заголовок каждого из файлов данных. Эту операцию выполняет процесс CKPT.

● Процесс CKPT 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.

**Процесс архиваторов**

**ARCn – Archiver**

● Копируют файлы журнала повторов на указанное устройство хранения после заполнения журнала.

● Могут собирать данные для восстановления транзакций.

● Функционируют, только если БД работает в режиме ARCHIVELOG.

● Можно изменить максимальное количество процессов архиваторов при помощи параметра инициализации LOG\_ARCHIVE\_MAX\_PROCESSES.

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

Обычно DBA занимается установкой программного обеспечения Oracle и созданием базы данных. Также DBA может отвечать за создание структур хранения базы данных, например, табличных пространств. Кроме того, он может создавать схемы или наборы объектов для хранения данных приложений.

Переменные окружения:

• **ORACLE\_BASE.** Устанавливает путь к «корню» иерархии каталогов Oracle. Например: export ORACLE\_BASE=/u01/app/oracle

• **ORACLE\_HOME.** Устанавливает путь к «корневому» каталогу БД. Свой для каждого экземпляра БД. Пример: export ORACLE\_HOME=$ORACLE\_BASE/product/11.1.0/db\_1

• **ORACLE\_SID.** Задаёт имя экземпляра Oracle. Значение по умолчанию - ORCL. Формат — строка, состоящая из цифр и букв и начинающаяся с буквы.

Способы установки:

1) Oracle Universal Installer — GUI на Java

./runInstaller

2) «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:

$ su

# password:

# cd /u01/app/oracle/oraInventory

# ./orainstRoot.sh

# cd /u01/app/oracle/product/11.1.0/db\_1

# ./root.sh

**9. Архитектура хранения базы данных. Подход OFA.**

Файлы, из которых состоит БД, делятся на следующие категории:

● **Управляющие файлы.** Содержат данные о самой базе данных (то есть информацию о физической структуре базы данных).

● **Файлы данных.** Содержат данные пользователя или приложения БД, а также метаданные и словарь данных.

● **Оперативные файлы журнала повторов**. Если происходит сбой сервера БД, при котором не теряются файлы данных, экземпляр позволит восстановить базу данных с помощью информации, содержащейся в этих файлах.

Помимо перечисленных, экземпляр БД использует следующие файлы:

● **Файл параметров.** Используется для определения конфигурации экземпляра для запуска.

● **Файл паролей.** Позволяет удаленно подключаться к БД пользователям sysdba, sysoper и sysasm.

● **Резервные копии файлов.** Используются для восстановления БД.

● **Архивные файлы журнала повторов**. Содержат непрерывную историю изменений данных (повторных операций), которую создает экземпляр. С помощью этих файлов и резервной копии БД можно восстановить утраченные файлы данных.

● **Файлы трассировки.** Любой серверный или фоновый процесс может выполнять запись в определенный файл трассировки. При обнаружении процессом внутренней ошибки он записывает дамп информации об ошибке в свой файл трассировки.

● **Файлы журнала предупреждений.** Журнал предупреждений БД – это хронологический журнал сообщений и ошибок. Каждый экземпляр использует один файл журнала предупреждений.

**Optimal Flexible Architecture (OFA) - метод конфигурирования БД Oracle или других БД.**

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

Суть OFA заключается в схеме, определяющей стандартные имена точек монтирования, каталогов и подкаталогов на этих точках, а так же самих файлов.

\* *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, 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.

● Универсальный подход (шаблон) к конфигурированию СУБД (не только Oracle).

● Описывает структуру каталогов БД и других ресурсов в ФС.

● Как и остальные шаблоны, предназначен для построения максимально гибкой структуры экземпляра БД и избежания возможных «типовых» проблем.

● Не является обязательным.

Имена каталогов в сответствии с OFA:

● Точки монтирования — /pm (например, /u01 или /u02).

● Домашние каталоги — /pm/h/u (например, /u01/app/oracle или /u02/home/oracle).

● Каталоги бинарных файлов СУБД — /pm/h/u/product/v (например, /u01/app/oracle/product/11.1.0/db\_1).

● Каталоги с конфигурационными и другими административными файлами — /pm/h/u/admin/d/a (например, /u01/app/oracle/admin/db01/arch).

● Файлы БД:

● Управляющие файлы: /pm/q/d/controln.ctl.

● Файлы журнала повторов: /pm/q/d/redon.log.

● Файлы данных: /pm/q/d/tn.dbf.

**10. Параметры инициализации экземпляра БД. Файлы параметров.**

● Параметры инициализации экземпляра БД 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<SID>.ora.

● **Текстовый файл параметров инициализации (PFILE)** — может быть только считан сервером, но не записан. Настройки параметров инициализации необходимо задавать и изменять вручную (с помощью текстового редактора). Имя по умолчанию — init<SID>.ora. При наличии SPFILE этот файл игнорируется.

**Текстовый файл с параметрами инициализации БД (PFILE)**

Он должен содержать как минимум 3 параметра:

● DB\_NAME — имя БД (максимум 8 символов).

● CONTROL\_FILES — список управляющих файлов БД.

● MEMORY\_TARGET — общее количество памяти, которое будет выделено экземпляру БД.

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.х.

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.**

С помощью Oracle Enterprise Manager или SQL \*Plus.

Режимы запуска[STARTUP] - NOMOUNT, MOUNT, OPEN

• **NOMOUNT** - запуск без монтирования. Читается файл параметров, инициализируются фоновые процессы и структуры памяти, но не связываются с дисковыми структурами БД. Позволяет выполнять разные задачи, например создание БД. В NOMOUNT можно попасть при ошибке запуска в режиме MOUNT.

• **MOUNT** - то что в NOMOUNT + связывает структуры БД, получает инфу из управляющих файлов.

• **OPEN (по ум.)** - NOMOUNT + MOUNT + открывает файлы, как описано в упр файле данного экземпляра (делает БД доступной для пользователей)

*При монтировании (MOUNT) БД выполняются следующие операции:*

● БД 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.

*При открытии (OPEN) БД выполняются следующие операции:*

● Открытие файлов данных.

● Открытие оперативных файлов журнала повторов.

*Режимы остановки[SHUTDOWN]*

• **ABORT -** выполняется минимальное количество операций до остановки.

• **IMMEDIATE -** наиболее часто используемый режим. Для всех неподтвержденных транзакций выполняется откат.

• **TRANSACTIONAL -** позволяет завершить транзакции.

• **NORMAL -** ожидает завершения сеансов.

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

**Словарь данных (Data Dictionary, DD)** — это набор доступных только для чтения таблиц и представлений, которые содержат различную информацию о БД:

● Информацию обо всех объектах схемы БД (таблицах, представлениях, индексах, кластерах, синонимах, последовательнастях, процедурах, функциях, пакетах, триггерах и т. д.).

● Информацию о том, сколько дискового пространства выделено объектам схемы, и какой процент этого пространства уже использован.

● Информацию о значениях полей таблиц по умолчанию.

● Информацию о существующих в БД ограничениях целостности.

● Сведения обо всех пользователях БД, их ролях и выданных привилегиях.

● Результаты текущего аудита — кто в данный момент имеет обращается и/или модифицирует объекты схемы.

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.

Сведения словаря данных:

- используется сервером баз данных Oracle для поиска информации о клиентах, объектах, ограничениях и хранилищах

- обновляются сервером базы данных Oracle 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

Словарь данных состоит из двух «уровней»:

● Системные таблицы — содержат информацию в логичном с точки зрения архитектуры системы виде.

● Представления — содержат ту же самую информацию в более удобном для чтения и обработки формате.

**Dynamic Performance Views (DPV) – динамические представления производительности**

Полезные представления:

● 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\_WAIT — детальная посеансовая информация о ресурсах, которые сеанс ожидает в данный момент или ждал в последний раз.

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

Популярные представления:

● CAT, USER\_CATALOG, ALL\_CATALOG, DBA\_CATALOG.

● TAB, TABS, USER\_TABLES, ALL\_TABLES, DBA\_TABLES.

● DICT, DICTIONARY — описание таблиц и представлений словаря.

● IND, USER\_INDEXES, ALL\_INDEXES, DBA\_INDEXES.

● XXX\_SNAPSHOTS, XXX\_DB\_LINKS, XXX\_CLUSTERS, XXX\_ERRORS,

XXX\_SOURCES, XXX\_DEPENDENCIES…

Динамические представления производительности содержат сведения о:

- сеансах

- состоянии файлов

- ходе выполнения заданий и задач

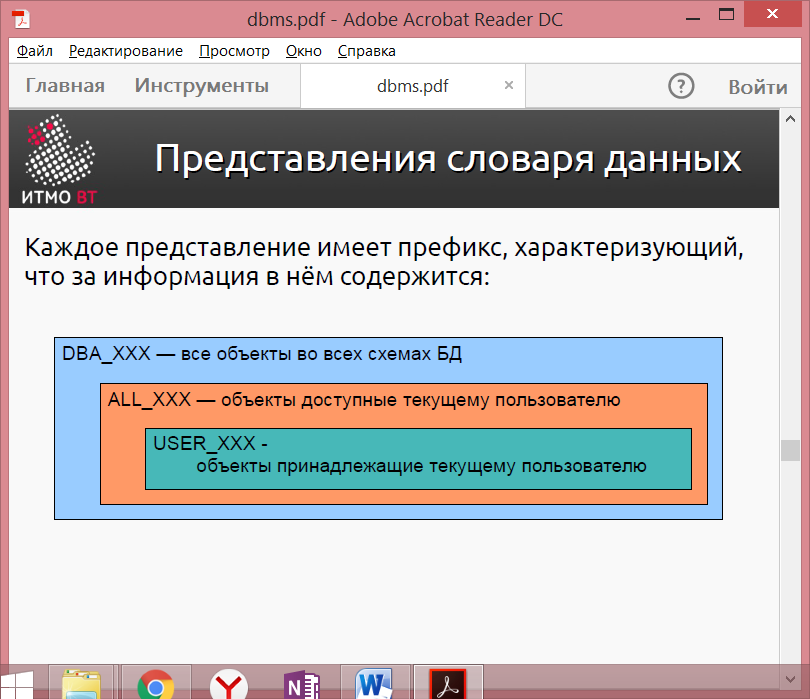
- блокировках

- состоянии резервного копирования

- использовании и выделении памяти

- параметрах системы и сеансов

- выполнении SQL



For almost every V$ view, a corresponding GV$ view exists. In Oracle Real Application Clusters (Oracle RAC), querying a GV$ view retrieves the V$ view information from all qualified database instances

**13. Методы разрешения имен, настройка псевдонимов.**

Разрешение имен: определение простых имен, идентификаторов соединения, а также их отображения в дескрипторы соединения, что позволяет определять сетевое расположение и идентифицировать службы.

\* 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.

- Локальное разрешение имен :: при данном методе дескрипторы соединения 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.

Чтобы создать локальный псевдоним службы Oracle Net, в раскрывающемся списке «Administer» (Администрирование) выберите пункт «Local Naming»(Локальное разрешение имен) и нажмите кнопку «Go» (Перейти). Затем нажмите кнопку «Create»(Создать).

Чтобы настроить псевдонимы служб для разрешения имен в сервере каталогов, выберите пункт «Directory Naming» (Разрешение имен в сервере каталогов), а не«Local Naming»(Локальное разрешение имен).

**tnsping** — утилита, предназначенная для проверки псевдонимов служб Oracle Net (аналог ping):

● Проверяет соединение между клиентом и Oracle Net Listener.

● Не проверяет доступность самой запрошенной службы — только слушателя.

● Поддерживает разрешение имен Easy Connect: tnsping db.us.oracle.com:1521/dba11g

● Поддерживает локальное разрешение имен и разрешение имен на сервере каталогов:

tnsping orcl

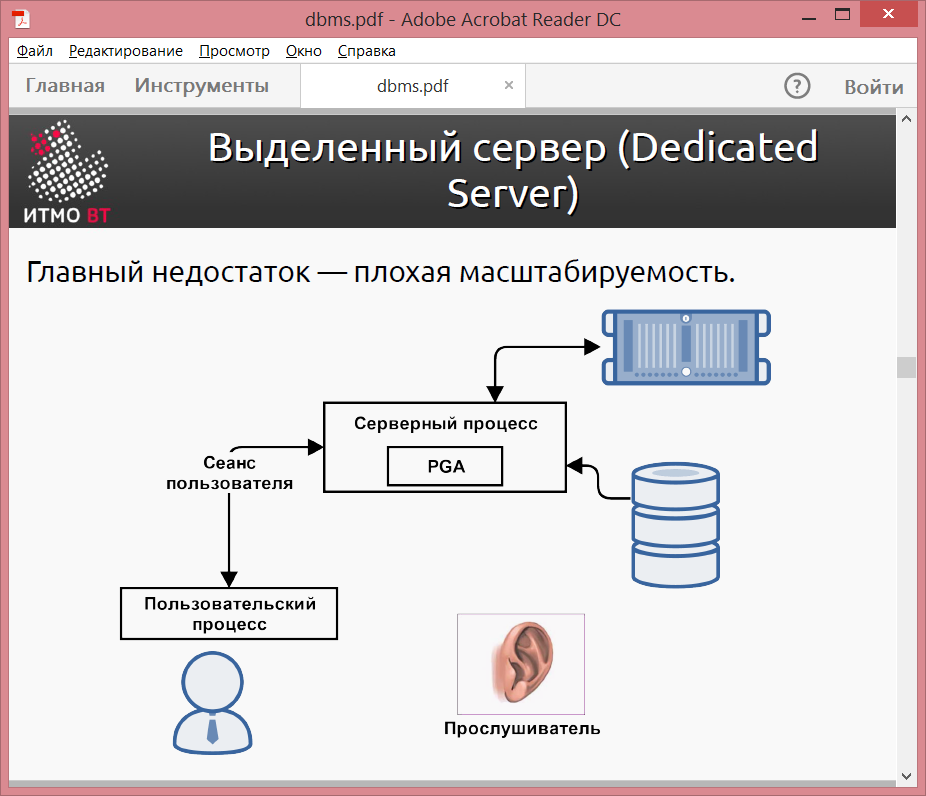
**14. Выделенный и разделяемый режим работы сервера. Преимущества и недостатки.**

**Выделенный сервер (Dedicated Server)**

В режиме выделенного сервера каждому пользовательскому процессу создаётся свой «персональный» серверный.

**Главный недостаток** – плохая масштабируемость

Выделенный сервер: для каждого пользователя приложение базы данных запускается пользовательским процессом, который обслуживается выделенным серверным процессом, исполняющим код сервера баз данных Oracle. В системах с интенсивной нагрузкой выделение ресурсов памяти и процессора выделенным серверным процессам может оказаться недопустимым и негативно отразиться на масштабируемости таких систем.



**Разделяемый сервер (Shared Server)**

В режиме разделяемого сервера каждому серверные процессы выделяются диспетчером из специального пула.

● Для каждой службы в архитектуре разделяемого сервера используется как минимум один процесс диспетчера (обычно больше).

● Слушатель хранит список доступных диспетчеров для каждого имени службы, а также информацию о загрузке соединения (количество одновременных соединений) для каждого диспетчера.

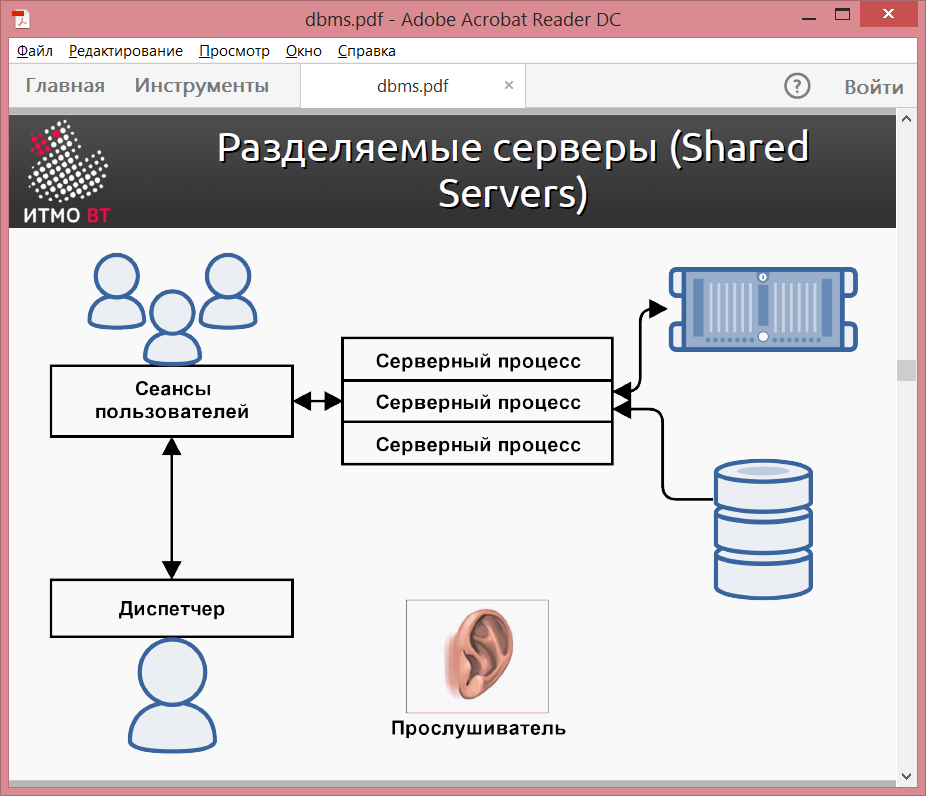
● Запрос перенаправляется к наименее загруженному диспетчеру, обслуживающему службу с запрошенным именем.

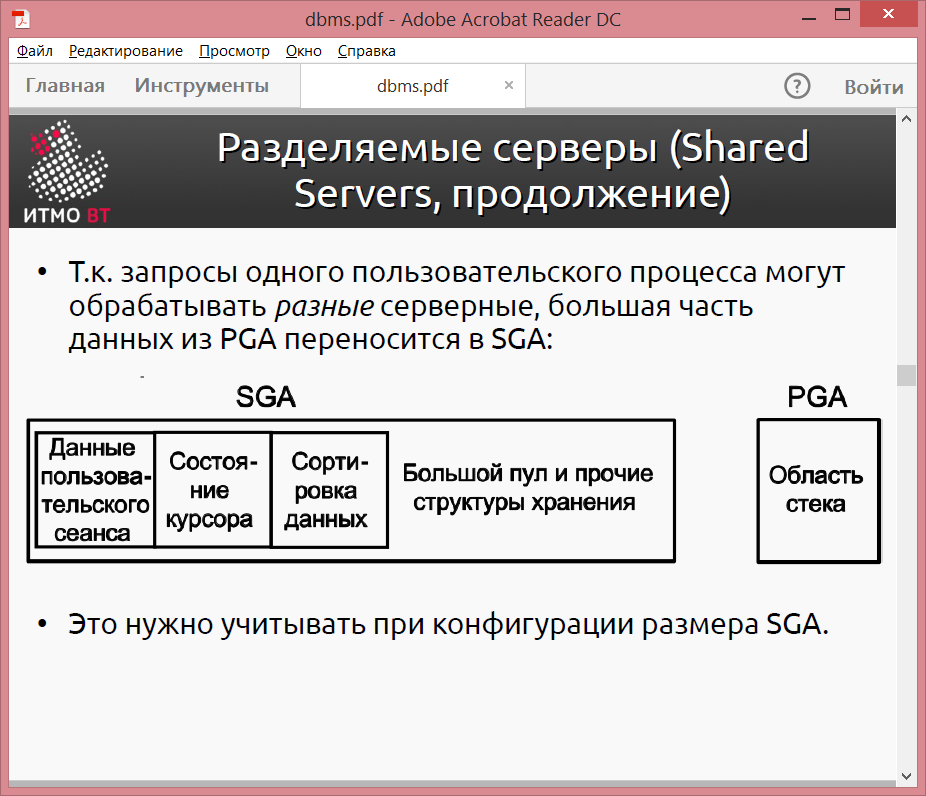
● Во время сеанса пользователь поддерживает соединение с одним и тем же диспетчером (но запросы могут обрабатывать разные серверные процессы).

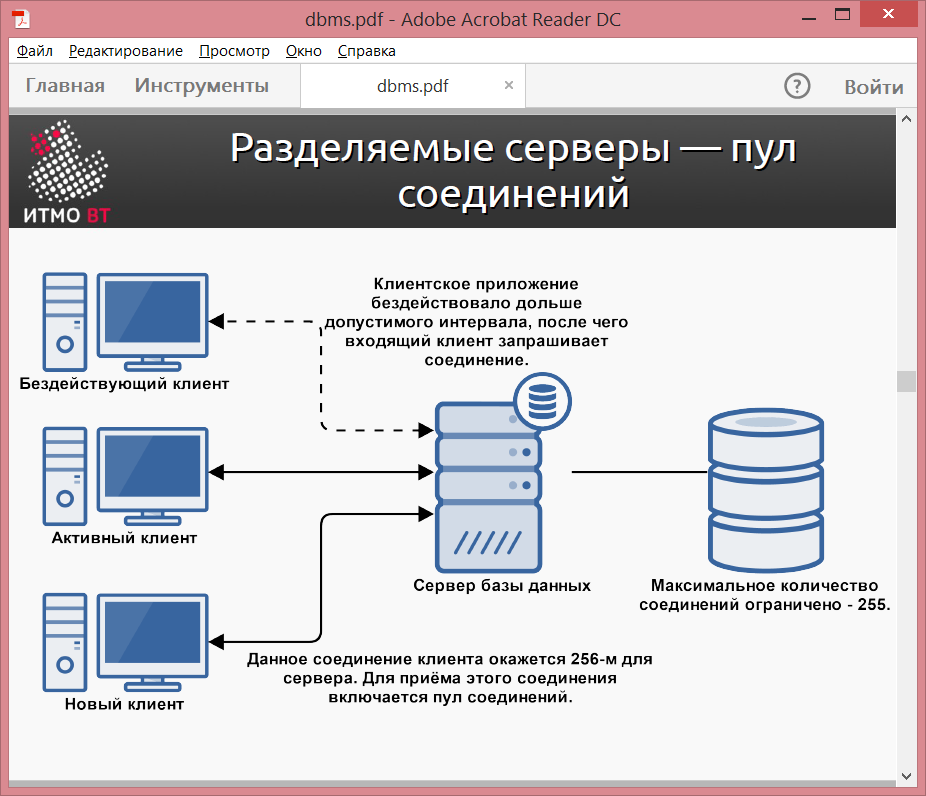
● Один диспетчер может обслуживать сотни сеансов пользователей.

● Диспетчеры направляют запросы пользователей в общую очередь, которая размещена в области SGA, выделенной для разделяемого пула.

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. На самом деле диспетчеры не занимаются обработкой запросов пользователей. Они 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 обрабатываютих.







**Когда не стоит использовать разделяемые серверы**

Определённые операции с БД не стоит выполнять с помощью разделяемых серверов:

● Администрирование БД.

● Операции резервного копирования и восстановления.

● Пакетную обработку и операции с массовой загрузкой.

● Операции с хранилищами данных.

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

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 по некоторым причинам фрагменты строки могут храниться в разных местах.

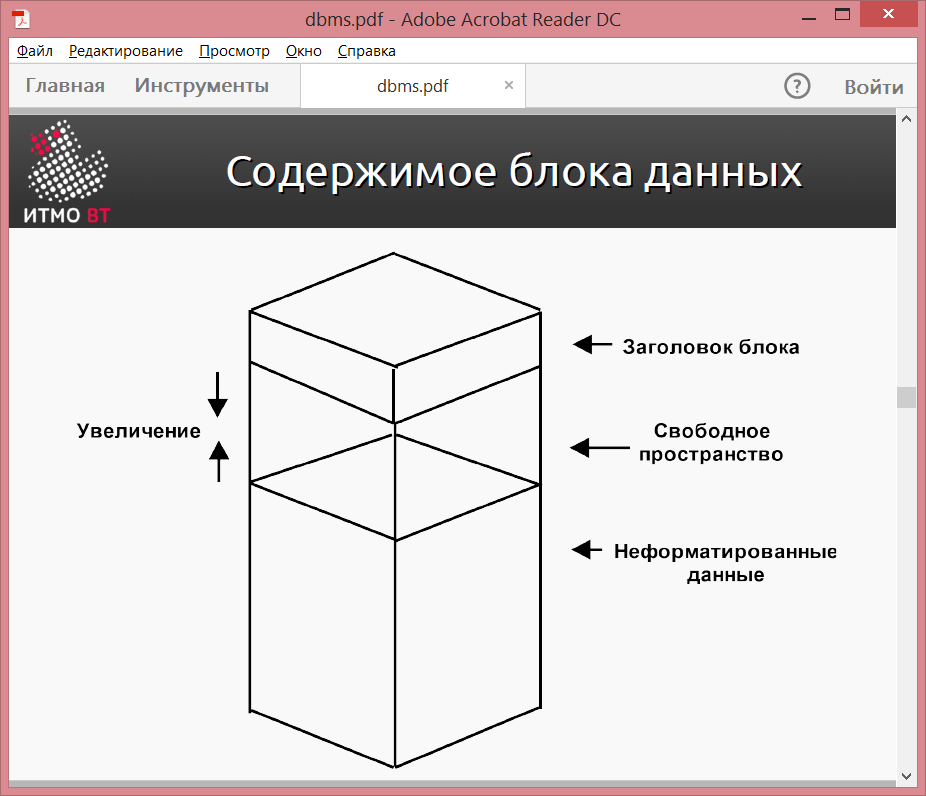
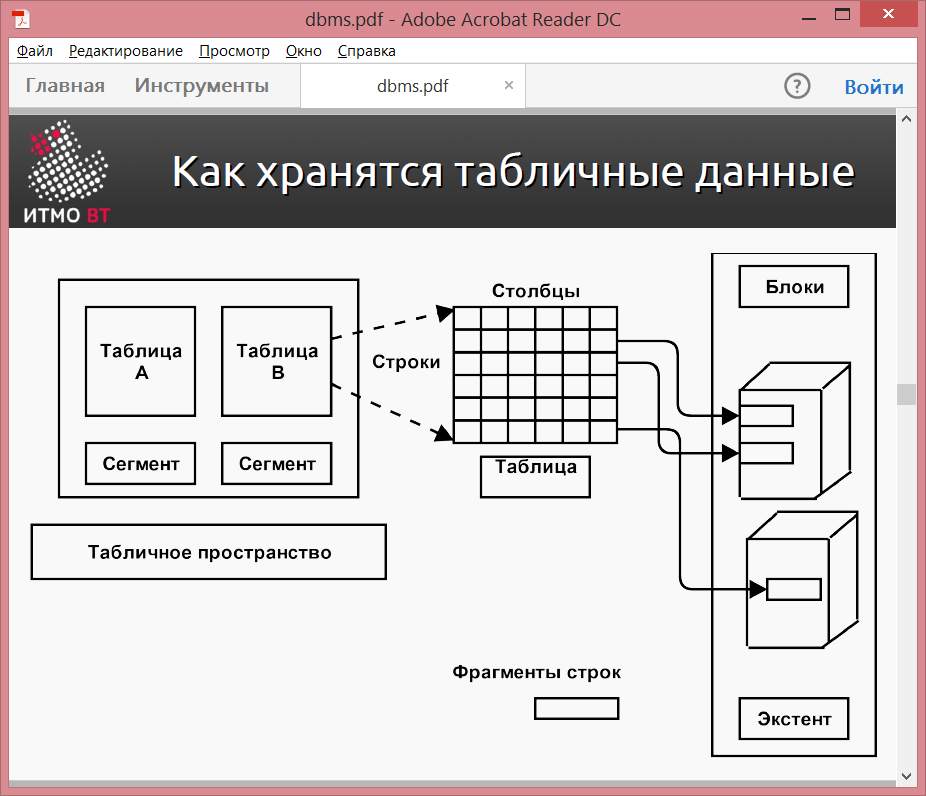
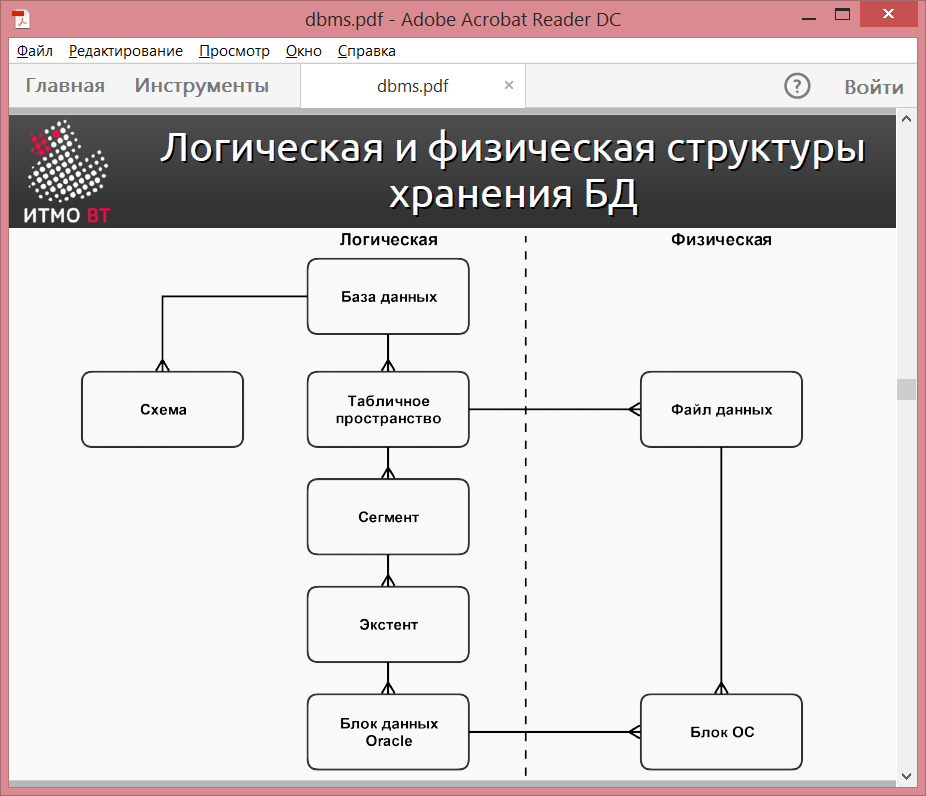
**Размер блока данных:**

● Определяется параметром инициализации DB\_BLOCK\_SIZE.

● Размер — от 2 до 32 КБ.

● По умолчанию — 8 КБ.

● Изменить размер блока данных можно только путём повторного создания БД.



● 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. Заполняется сверху вниз.

● Неформатированные данные — данные для строк блока. Заполняются снизу вверх.

● Свободное пространство внутри блока, за счет которого могут увеличиваться заголовок и пространство неформатированных данных.

● События, которые могут увеличить размер заголовка:

● Требуется больше неформатированных записей.

● Требуется большее количестве транзакционных слотов.

● Изначально свободное пространство является непрерывным, однако в ходе выполнения операций удаления и обновления может стать фрагментированным.

● Свободное пространство может быть дефрагментировано сервером Oracle.

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

● БД состоит из одного или нескольких логических табличных пространств.

● Каждое табличное пространство базы данных Oracle состоит из одного или нескольких файлов данных.

● БД должна содержать минимум два табличных пространства — SYSTEM и SYSAUX, каждое из которых представлено минимум одним файлом данных.

● Одна БД может иметь до 65534 файлов данных.

● Если табличное пространство в течение всего жизненного цикла представлено одним (и только одним) файлом данных, оно называется табличным пространством типа BIGFILE.

● Временный файл – это файл, который принадлежит временному табличному пространству:

● Создается с параметром TEMPFILE:

alter tablespace temp

add tempfile 'c:\oracle\oradata\temp3\temp02.dbf'

size 50m

reuse

autoextend on

next 1m

maxsize 500m;

● Временные табличные пространства используются для операций сортировки и не могут содержать постоянных объектов базы данных, таких как таблицы.

**Управление табличными пространствами**

● Управляемое локально:

● управление свободными экстентами осуществляется в табличном пространстве;

● для записи свободных экстентов используется битовый образ;

● каждый бит соответствует блоку или группе блоков;

● значение бита описывает экстент: свободен или занят.

● Управляемое словарём:

● свободными экстентами управляет Oracle через словарь данных;

● при выделении и освобождении экстентов обновляются соответствующие представления

**Табличные пространства, управляемые локально**

● Экстенты могут выделяться одним из двух способов:

● Автоматически — размерами экстентов управляет система (он всегда кратен 64 КБ). Cпособ неприменим ко временным табличным пространствам.

● Унифицированно — табличные пространства используют унифицированный размер экстентов указываемый администратором (по умолчанию — 1 МБ). Режим нельзя использовать для табличных пространств отмены операций.

● Управление пространством сегментов можно осуществлять:

● Автоматически — используются битовые образы. Битовый образ описывает состояние каждого блока данных в сегменте в соответствии с объемом пространства блока, которое доступно для вставки строк.

● Вручную — используются списки свободных сегментов (списки блоков данных, в которых имеется свободное пространство). Требуется вручную задавать и настраивать значения параметров хранения PCTUSED, FREELISTS и FREELIST GROUPS для объектов схемы.

**Состояния табличных пространств**

● **(Оперативное) Чтение-запись:** доступно как для чтения, так и для записи.

● **(Оперативное) Только чтение:** доступно только для чтения. После перевода в это состояние текущие транзакции будут завершены (с фиксацией или откатом), новые операции DML будут запрещены. Табличные пространства SYSTEM и SYSAUX нельзя перевести в этот режим.

● **Автономное:** эта часть БД становится временно недоступной для использования (остальная БД будет оставаться открытой и доступной). При переводе в автономный режим можно использовать следующие параметры:

● Normal: режим можно использовать, если во всех файлах данных табличного пространства отсутствуют ошибки. СУБД установит контрольную точку для всех файлов данных перед переводом их в автономный режим.

● Temporary: табличное пространство временно переводится в автономный режим даже при наличии ошибок в файлах данных. При «обратном» переводе в оперативный режим может потребоваться восстановление данных из журнала повторов.

● Immediate: табличное пространство можно перевести в автономный режим немедленно, без установки контрольной точки для файлов данных БД Oracle. При «обратном» переводе в оперативный режим обязательно потребуется восстановление данных из журнала повторов.

Для удаления табличных пространств используется команда **DROP TABLESPACE**

● Нельзя удалить табличное пространство SYSTEM.

● Табличное пространство SYSAUX может удалить только DBA и только если БД запущена в режиме MIGRATE.

● Нельзя удалить табличное пространство отката если оно используется и в нём есть незавершённые транзакции.

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

**Формуляр (account**) пользователя БД — это способ организации принадлежности и доступа к объектам БД.

**Пароль** необходим для аутентификации в БД Oracle.

**Полномочие (privilege)** — это право на выполнение определенного типа SQL-оператора или на доступ к объекту пользователя.

**Роль (role)** — это именованная группа связанных полномочий, которая предоставляется пользователям или другим ролям.

**Профили (profiles)** представляют собой именованные наборы ограничений на использование ресурсов БД и экземпляра.

**Квота (quote)** — это допустимый объем пространства в заданном табличном пространстве.

У каждого пользователя БД есть свой уникальный **формуляр БД.**

Для доступа к БД пользователю необходимо указать соответствующий формуляр пользователя БД и успешно пройти аутентификацию для данного формуляра.

**У каждого формуляра пользователя есть:**

● *Уникальное имя пользователя.* Не может превышать 30 байт, не может содержать специальные символы, должно начинаться с буквы.

● *Метод аутентификации.* По умолчанию — пароль.

● *Табличное пространство* по умолчанию.

● *Временное табличное пространство.*

● *Профиль пользователя*. Набор ресурсов и ограничений с помощью паролей, присвоенных пользователю.

● *Состояние формуляра*. Пользователям доступны только «открытые» формуляры.

**Формуляр SYS:**

● получает роль администратора БД;

● обладает всеми полномочиями с параметром ADMIN OPTION;

● необходим для запуска, остановки и выполнения некоторых служебных команд;

● является владельцем словаря данных;

● является владельцем репозитория автоматической рабочей нагрузки (AWR).

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.

**Формуляру SYSTEM** предоставляется роль администратора БД.

Оба этих формуляра не должны использоваться для стандартных операций.

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

**Полномочие (privilege)** — это право на выполнение определенного типа SQL-оператора или на доступ к объекту пользователя.

Существует два типа полномочий пользователя:

**● Системные:** позволяют пользователям выполнять определённые действия с базой данных.

**● Объектные:** позволяют пользователям получать доступ к определенным объектам и манипулировать ими.

**Предоставление и аннулирование полномочий:**

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;

Если системные полномочия были предоставлены с помощью команды GRANT, их можно аннулировать с помощью SQL-оператора REVOKE. Пользователи, которым системные полномочия были предоставлены с параметром ADMIN OPTION, могут аннулировать полномочия других пользователей базы данных. Аннулировать полномочия может не только пользователь, предоставивший их.

**Системные полномочия**

● Предоставление полномочий, которые содержат фразу **ANY**, означает выход за пределы схемы. Например, обладая полномочиями CREATE TABLE, можно создать таблицу, однако только в собственной схеме. А полномочия SELECT ANY TABLE позволяют выбирать среди таблиц других пользователей.

● Пользователь SYS и пользователи, владеющие ролью DBA, обладают всеми полномочиями ANY.

● **SYSDBA** и **SYSOPER**. Эти полномочия позволяют выполнять административные задачи в БД. SYSOPER позволяет пользователю выполнять оперативные задачи, но без возможности просмотра пользовательских данных. В него входят следующие системные полномочия:

● STARTUP и SHUTDOWN

● CREATE SPFILE

● ALTER DATABASE OPEN/MOUNT/BACKUP

● ALTER DATABASE ARCHIVELOG

● ALTER DATABASE RECOVER (только полное восстановление).

● RESTRICTED SESSION

Системное полномочие SYSDBA дополнительно дает разрешение на неполное восстановление и удаление базы данных.

● **SYSASM**. Позволяет запустить, остановить экземпляр ASM и управлять им.

● DROP ANY объект. Позволяет удалять объекты других пользователей схемы.

● CREATE, MANAGE, DROP и ALTER TABLESPACE. Позволяют управлять табличными пространствами.

● CREATE LIBRARY. В БД Oracle имеется возможность создавать и вызывать внешний код (например библиотеку C) с помощью PL/SQL. Такой библиотеке должно быть присвоено имя объектом LIBRARY в базе данных.

● CREATE ANY DIRECTORY. В качестве меры безопасности каталог ОС, в котором хранится код, необходимо связать с виртуальным объектом каталога Oracle. Владелец полномочия CREATE ANY DIRECTORY потенциально может вызвать небезопасные объекты кода.

● GRANT ANY OBJECT PRIVILEGE. Позволяет предоставлять разрешения для объектов, которыми его обладатель не владеет.

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

**Объектные полномочия**

GRANT privileges ON object TO user;

REVOKE privileges ON object FROM user;

Виды 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.

**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.

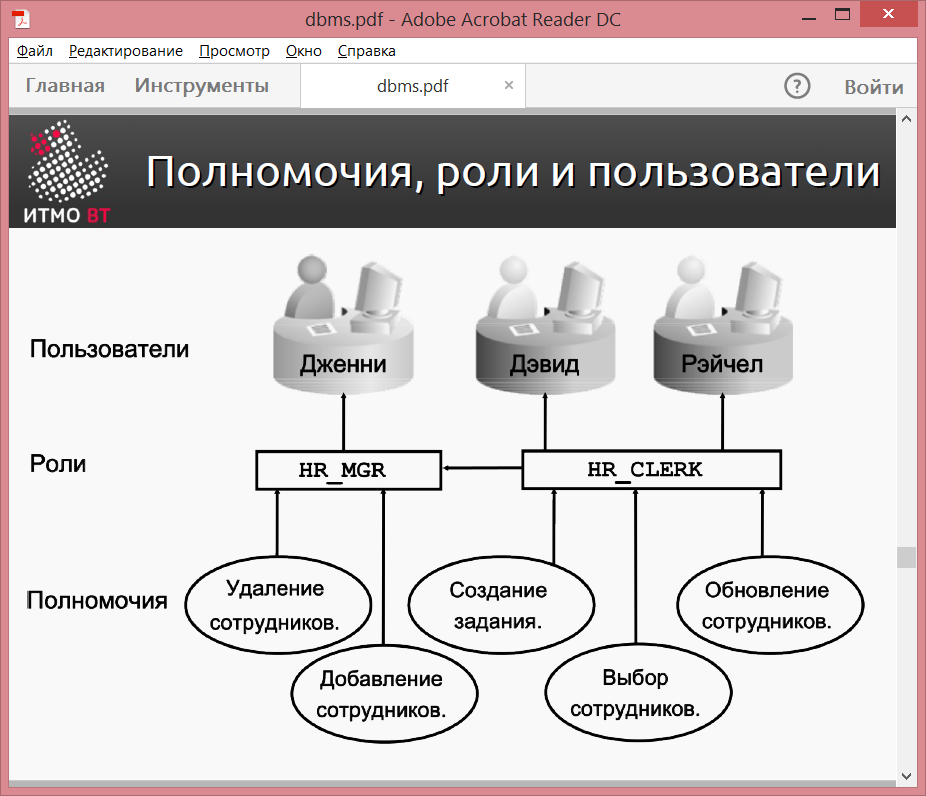
**19. Роли БД. Предопределенные роли. Профили пользователей.**

**Преимущества использования ролей**

● **Упрощенное управление полномочиями.** Роли используются для упрощенного управления 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.



Существует несколько ролей, которые автоматически определяются в базах данных Oracle при запуске сценариев создания БД.

**● Профили (profiles)** — это именованные наборы ограничений на использование ресурсов БД и экземпляра.

● Каждому пользователю назначается профиль, причем одновременно только один.

● Если при изменении профиля пользователь уже находился в системе, изменения вступят в силу со следующего сеанса работы пользователя.

● Профиль DEFAULT является базовым для всех остальных профилей.

● Профили не накладывают ограничения на использование ресурсов пользователями, если параметру инициализации RESOURCE\_LIMIT не присвоено значение TRUE.

● Если параметру RESOURCE\_LIMIT присвоено стандартное значение FALSE, ограничения на использование ресурсов, заданные профилем, игнорируются. Настройки пароля для профиля всегда применяются принудительно.

С помощью профилей администратор может контролировать следующие системные ресурсы:

**● Процессор**. Ресурсы процессора можно ограничивать на основе сеансов или вызовов:

● Ограничение процессор/сеанс, равное 1000, означает, что если сеанс потребляет более 10 секунд процессорного времени, то будет вызывана ошибку и выполнен выход:

ORA-02392: exceeded session limit on CPU usage, you are being logged off

● При ограничении вызовов место всего сеанса пользователя контролируется процессорное время, потребляемое отдельными командами. Если команда пользователя превышает его, её выполнение прерывается и пользователь получает сообщение об ошибке: ORA-02393: exceeded call limit on CPU usage

● **Сеть/память.** Можно задать следующие параметры:

● Время соединения: указывает 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.

● Параллельные сеансы: определяет, сколько параллельных сеансов может быть создано с помощью формуляра пользователя БД.

**● Личная область SGA.** Ограничивает размер области, выделяемой в глобальной системной области (SGA) для сортировки, слияния растровых изображений и т.д.

**● Дисковый ввод/вывод**. Ограничивает объем данных, которые пользователь может считать за сеанс или за вызов.

● Кроме того, для профиля можно задать **смешанные ограничения.**

**Квоты**

Если пользователю не предоставлено системное полномочие UNLIMITED TABLESPACE, для создания объектов в табличном пространстве такому пользователю следует задать квоту.

Квота может иметь:

● значение в мегабайтах или килобайтах;

● неограниченное значение.

**20. Целостность, ограничения, состояние ограничений.**

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

- NOT NULL

- UNIQUE

- PRIMARY KEY

Если при выполнении DML-оператора проверки условия ограничения получен результат FALSE, выполняется откат оператора, если наложено ограничение IMMEDIATE. Если ограничение является отложенным (DEFERRED), откат производится при фиксации, а не выполнении DML.

Чтобы проще разрешать ситуации, когда данные временно могут нарушать ограничение, можно использовать различные состояния ограничений. Ограничение целостности можно включить(ENABLE) или выключить(DISABLE).

**21. Понятие блокировки, уровни блокировки и операторы их использующие.**

Перед тем как база данных позволит сеансу изменить данные, он должен заблокировать изменяемые данные. Блокировка дает сеансу монопольные права на работу с данными, благодаря чему другая транзакция не сможет изменить заблокированные данные до тех пор, пока блокировка не будет снята.

\* Уровни блокировок

- строка :: для транзакций, которые изменяют данные

- объект :: изменение самого объекта целиком

LOCK, UPDATE(?)

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

Конфликты блокировок обычно устраняются временем и механизмом обслуживания очереди. В некоторых редких случаях для устранения конфликта блокировок может потребоваться вмешательство администратора.

Чтобы устранить конфликт блокировок, сеанс должен освободить созданную блокировку. Наилучший способ достичь этого – связаться с пользователем и попросить его завершить транзакцию.

Взаимная блокировка – это особый случай конфликта блокировок. Взаимные блокировки возникают при участии двух (или больше) сеансов, когда сеансы ожидают данные, взаимно заблокированные друг другом.

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

База данных Oracle сохраняет исходные значения данных, когда процесс изменяет данные в базе данных. Данные сохраняются до внесения в них изменений. Сохранение данных отмены операций позволяет выполнить откат неподтвержденных данных.

\* Данные отмены операций сохраняются для всех транзакций как минимум до завершения транзакции по одной из причин:

- пользователь отменил транзакцию

- пользователь завершил транзакцию

- пользователь выполнил DDL-оператор, такой как CREATE, DROP, RENAME или ALTER

- неплановое завершение сеанса пользователя

- запланированное завершение сеанса пользователя

Данные отмены операций и данные повторов на первый взгляд похожи, однако они используются для разных целей. **Данные отмены** операций нужны для того, чтобы иметь возможность отменить внесенные изменения. **Данные повторов** нужны для того, чтобы иметь возможность повторно применить изменения, если они будут утеряны по каким-либо причинам. Изменения в блоках данных отмены операций так же сохраняются в журнале повторов.

Процесс фиксации сохраняет внесенные транзакцией изменения в файл журнала повторов, который является постоянным файлом на диске, в отличие от данных, хранимых в памяти.

**24. Автоматическое и ручное управление основной и разделяемой памятью.**

**???**

**25. Ошибки пользователя БД. Область мгновенного восстановления.**

**Область мгновенного восстановления (Flash Recovery Area):**

● Упрощает управление хранением резервных копий.

● Использует отдельное пространство на диске (отдельно от рабочих файлов БД); рекомендуется использовать отдельный накопитель.

● Расположение задается параметром USE\_DB\_RECOVERY\_FILE\_DEST.

● Должна быть достаточного размера.

● Управляется автоматически.

Настройка области мгновенного восстановления предполагает определение ее расположения, размера и методики сохранения.

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

**Автоматическое восстановление** после сбоя **экземпляра**:

● Производится вследствие попыток открыть базу данных, файлы которой не были синхронизированы при завершении работы.

● Не требует от администратора никаких действий, за исключением запуска экземпляра БД.

● Использует информацию, хранящуюся в группах журналов повторов, для синхронизации файлов.

● Состоит из двух отдельных операций:

● накат: восстанавливается состояние файлов данных до момента сбоя экземпляра;

● откат: внесенные, но не зафиксированные изменения возвращаются к исходному состоянию.

**Этапы восстановления экземпляра**

1)Файлы данных не синхронизированы.

2) Накат (повтор).

3) Зафиксированные и незафиксированные данные в файлах.

4) Открытие БД.

5) Откат (отмена).

6) Зафиксированные данные в файлах.

**27. Резервное копирование. Полная и инкрементальная резервные копии.**

28. Менеджер резервного копирования RMAN

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

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

31. Экземпляр ASM. Конфигурация, взаимодействие с экземпляром Oracle.

Дополнительные процессы в составе экземпляра ASM и экземпляра Oracle.

32. Дисковые группы. Назначение, особенности конфигурации. Добавление и удаление

дисков. Allocation Units. Coarse- & Fine-Grained Striping.

33. Избыточность в Oracle ASM. Виды избыточности, конфигурация зеркалирования.

Failure-группы.

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

35. Архитектура быстрого восстановления с использованием сервера горячего резерва и

кластерная архитектура RAC.

36. Понятие кластера. Аппаратная и программная реализация кластера.

Масштабируемость и отказоустойчивость кластерных решений.

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

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

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

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

недостатки.