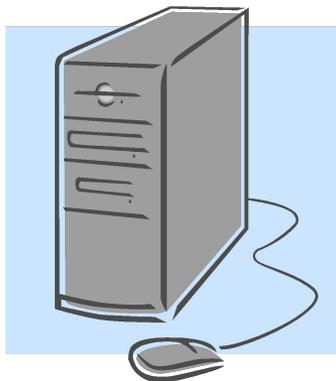
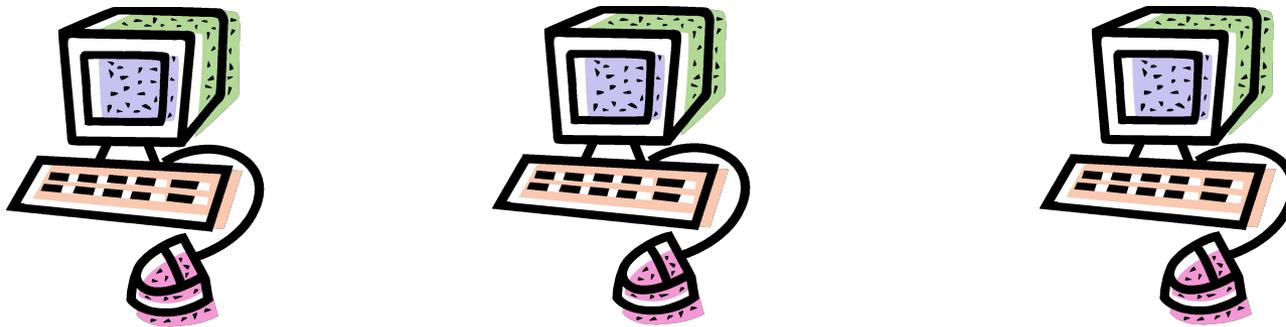


# Архитектура Oracle

Фёдоров Р.К.

- В Oracle термином база данных описывается физическое хранилище информации, а термином экземпляр - программное обеспечение, работающее на сервере и предоставляющее доступ к информации в базе данных. Экземпляр исполняется на конкретном компьютере или сервере; база данных хранится на дисках, подключенных к этому серверу.



Экземпляр  
ORACLE

Экземпляр состоит из  
процессов и оперативной  
памяти



База  
данных

- База данных- физическая сущность: она состоит из файлов, хранящихся на дисках.
- Экземпляр- сущность логическая: он состоит из структур в оперативной памяти и процессов, работающих на сервере. Например, Oracle использует область разделяемой памяти System Global Area (SGA, системная глобальная область) и области памяти в каждом процессе - Program Global Area (PGA, программная глобальная область). Экземпляр может быть частью одной и только одной базы данных. Напротив, с одной базой данных может быть ассоциировано несколько экземпляров. Время жизни экземпляров ограничено, тогда как база данных при должном обслуживании может существовать вечно.
- Пользователи не имеют прямого доступа к информации, хранящейся в базе данных Oracle; они должны запрашивать информацию у экземпляра Oracle.

# Структура базы данных Oracle

- База данных состоит из табличных пространств, управляющих файлов, журналов, архивных журналов, файлов трассировки изменения блоков, ретроспективных журналов и файлов резервных копий (RMAN).

# Табличные пространства

- Любые данные, хранящиеся в базе Oracle, должны находиться в каком то табличном пространстве. Табличное пространство (tablespace) – это логическая структура. Каждое табличное пространство состоит из физических структур, называемых файлами данных (datafiles). В одном табличном пространстве может быть один или несколько файлов данных, тогда как каждый файл данных принадлежит ровно одному табличному пространству. При создании таблицы можно указать, в какое табличное пространство ее поместить.



# Файлы базы данных

База данных Oracle состоит из физических файлов трех основных типов:

- управляющие файлы (control files);
- файлы данных (datafiles);
- журнальные файлы, или журналы (redo log files).

# Управляющие файлы (control files)

В управляющем файле хранится информация о местонахождении других физических файлов, составляющих базу данных, - файлов данных и журналов. Там же хранится важнейшая информация о содержимом и состоянии базы данных:

- имя базы данных;
- время создания базы данных;
- имена и местонахождение файлов данных и журнальных файлов;
- информация о табличных пространствах;
- информация о файлах данных в автономном режиме;
- история журналов и информация о порядковом номере текущего журнала;
- информация об архивных журналах;
- информация о наборах и фрагментах резервных копий, файлах данных и журналах;
- информация о копиях файлов данных;
- информация о контрольных точках.

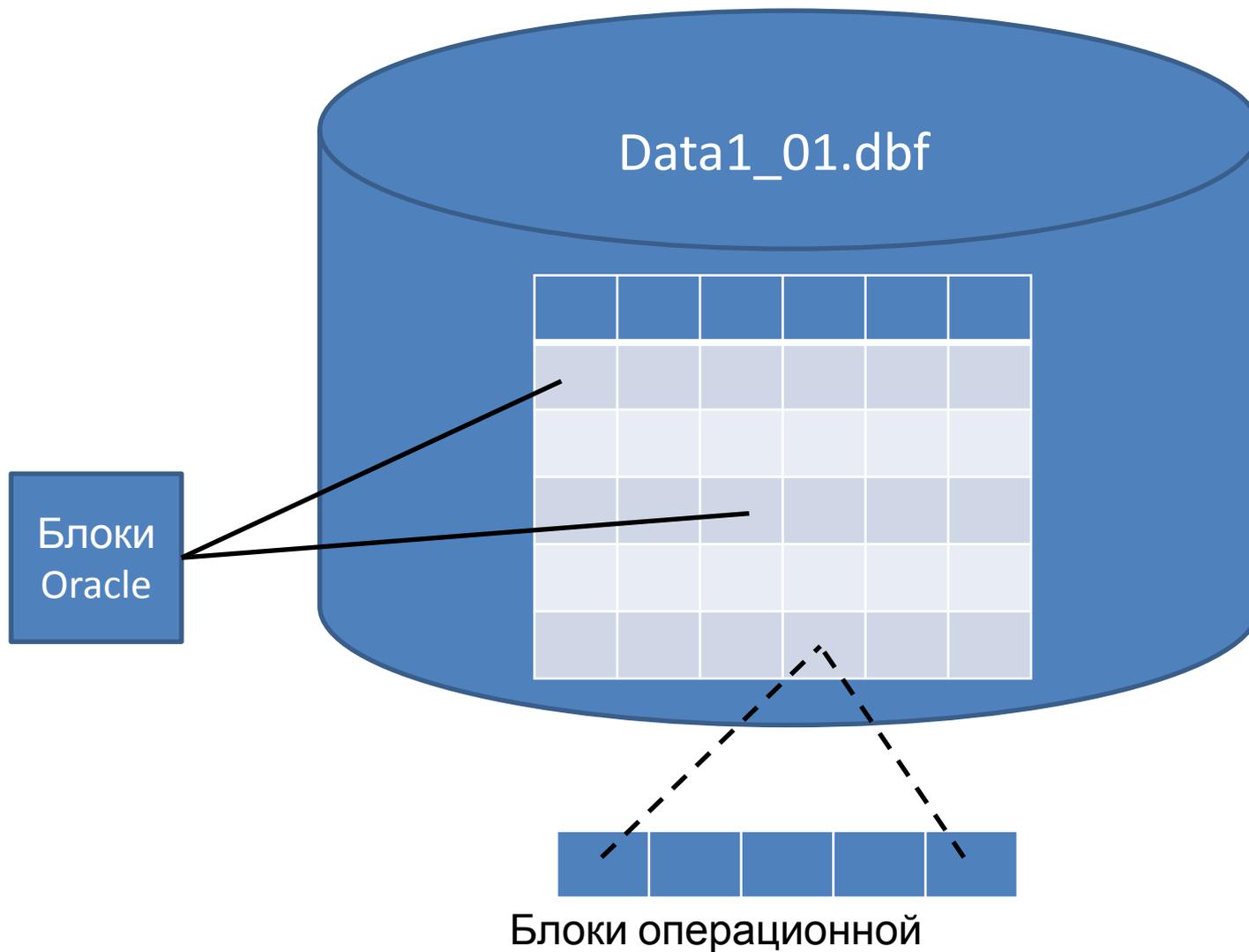
# Инициализация базы данных

- При запуске экземпляра Oracle считываются параметры
- инициализации. Они определяют, как база данных должна использовать физическую инфраструктуру и иную конфигурационную информацию об экземпляре. Параметры инициализации хранятся в файле параметров инициализации экземпляра, который обычно называют просто INIT.ORA, или, начиная с версии Oracle 9i, в репозитории, который называется файлом параметров сервера (или SPFILE). Количество обязательных параметров инициализации уменьшается с выходом каждой новой версии Oracle. В дистрибутиве Oracle есть пример файла инициализации, пригодный для запуска базы данных. Либо можно воспользоваться программой Database Configuration Assistant (DCA), которая подскажет обязательные значения (например, имя базы данных).

# Файлы данных

- В файлах данных находятся собственно данные, хранящиеся в базе: таблицы и индексы, словарь данных, в котором сохраняется информация об этих структурах, и сегменты отката, необходимые для реализации конкурентного доступа.
- Файл данных состоит из блоков базы данных, в свою очередь, состоящих из дисковых блоков операционной системы. Размер блока в Oracle варьируется от 2 до 32 Кбайт.
- Данные считываются с дисков в оперативную память блоками Oracle по мере необходимости, исходя из действий пользователей. Блоки данных переписываются из памяти в файлы данных на диске, когда это требуется для гарантии сохранности внесенных пользователями изменений.

# Файлы данных



# Журнальные файлы

- Журнальный файл содержит протокол всех изменений, произведенных в базе данных в результате выполнения транзакций и внутренних операций Oracle. Обычно измененные блоки кэшируются в памяти, поэтому в случае сбоя экземпляра может оказаться, что какие-то блоки не записались в файлы данных. Тогда можно воспользоваться протоколом, хранящимся в журнальных файлах, чтобы воспроизвести изменения, оставшиеся не записанными в момент сбоя, и тем самым обеспечить согласованность транзакции.
- Кроме того, журнальные файлы применяются для реализации операций отката (undo), инициируемых командой ROLLBACK. Незафиксированные изменения в базе данных откатываются, так что база остается в том состоянии, в котором находилась в момент последней фиксации.
- По умолчанию Oracle записывает в журнал все изменения, произведенные в базе данных. Но ведение журналов сопряжено с накладными расходами. Для ускорения некоторых операций можно подавить запись в журнал, однако это означает, что вы не сможете восстановить результат этой операции в случае сбоя.

# Журнальные файлы

Oracle поддерживает два типа журналов:

- Оперативные журналы, файлы операционной системы, в которые Oracle последовательно записывает изменения, произведенные в базе данных, циклически перебирая файлы.
- Архивные журналы, копии заполненных оперативных журналов, создаваемые во избежание потери данных при перезаписи оперативных журналов.

# Запуск СУБД

- В Windows достаточно запустить службы Oracle (или сконфигурировать их так, чтобы они запускались на этапе загрузки операционной системы), а в UNIX и Linux - выполнить команду STARTUP в SQL\*Plus или Enterprise Manager. При запуске СУБД автоматически выполняются следующие действия:

# Шаг 1. Запуск экземпляра

- Oracle считывает параметры инициализации экземпляра из файла SPFILE или INIT.ORA на сервере. Затем выделяется память для системной глобальной области и запускаются фоновые процессы экземпляра. В этот момент ни один из физических файлов базы данных еще не открыт, экземпляр находится в состоянии NOMOUNT.

# Шаг 2. Монтирование базы данных

- Экземпляр открывает управляющие файлы базы данных. Где их искать- сообщает параметр `CONTROL_FILES`. В этот момент открыты только управляющие файлы. Это состояние называется `MOUNT`, и в нем база данных доступна только администратору, который может выполнять с ней некоторые служебные операции. Например, администратор базы данных может переместить или переименовать один из файлов базы данных. Файлы данных, перечисленные в управляющем файле, в состоянии `MOUNT` еще не открыты. Для переименования файла данных администратор может выполнить команду `ALTER DATABASE`, которая запишет в управляющий файл новое имя файла данных.

# Шаг 3. Открытие базы данных

- Экземпляр открывает файлы журналов и файлы данных, пользуясь информацией, записанной в управляющем файле. В этот момент база данных наконец полностью открыта и доступна пользователям.

# Останов СУБД

1. **Закрытие базы данных.** Oracle сбрасывает на диск еще не записанные блоки данных из SGA, а также переписывает из журнального буфера в журналы информацию, необходимую для повторного выполнения. Затем Oracle записывает в файлы данных контрольную точку, отмечая, что их заголовки являются «текущими» на момент закрытия базы данных, после чего закрывает файлы данных и журналы. Начиная с этого момента, пользователи уже не могут обращаться к базе данных.
2. **Размонтирование базы данных.** Экземпляр Oracle размонтирует базу данных. В управляющих файлах помечается, что был выполнен корректный останов, после чего эти файлы закрываются. В этот момент сама база данных закрыта, остался лишь работающий экземпляр.
3. **Останов экземпляра.** Oracle останавливает фоновые процессы экземпляра и освобождает разделяемую память, занятую SGA.

# Аварийный останов

- В некоторых случаях (например, при сбое компьютера или после аварийного останова экземпляра администратором) корректного закрытия базы данных не происходит. Это означает, что у Oracle не было возможности сбросить модифицированные блоки из SGA в файлы данных.
- При следующем запуске экземпляр обнаружит, что имело место аварийное завершение и воспользуется хранящейся в журналах информацией для автоматического выполнения процедуры восстановления после сбоя. Гарантируется, что будут произведены все изменения, относящиеся к зафиксированным транзакциям, а изменения, относящиеся к незафиксированным или находившимся в процессе выполнения транзакциям, - отменены. Незафиксированные транзакции, выявленные после применения журналов, автоматически откатываются.

# Серверные процессы и клиенты

Для доступа к некоторой базе данных пользователь соединяется с экземпляром, предоставляющим доступ к ней. Программа, обращающаяся к базе данных, на самом деле состоит из двух частей - клиентской программы и серверного процесса, устанавливающего соединение с экземпляром Oracle. Например, при запуске текстовой утилиты SQL\*Plus создаются два процесса:

- сам процесс SQL\*Plus, выступающий в роли клиента;
- серверный процесс Oracle, иногда называемый теньвым процессом, который обеспечивает соединение с экземпляром Oracle.

# Серверный процесс

- Серверный процесс Oracle всегда работает на том же компьютере, что и экземпляр. Серверный процесс присоединяется к разделяемой памяти, в которой находится SGA, так что может читать и записывать в нее данные. Как и следует из его названия, серверный процесс обслуживает клиентский процесс - читает и возвращает запрашиваемые данные, выполняет изменения от имени клиента и так далее. Например, когда клиент хочет прочесть строку из конкретного блока базы данных, серверный процесс находит нужный блок и либо получает информацию из кэша буферов, либо считывает ее из соответствующего файла данных и помещает в кэш буферов. Далее, если пользователь требует внести изменения, серверный процесс модифицирует блок в кэше, а также порождает и сохраняет в журнальном буфере внутри SGA информацию, необходимую для повторного выполнения. Однако серверный процесс не записывает эту информацию в сами журнальные файлы и не сбрасывает модифицированные блоки из кэша буферов в файлы данных. Эти действия выполняют соответственно процессы Log Writer (LGWR) и Database Writer (DBWR).

# Клиентский процесс

- Клиентский процесс может работать на одной машине с экземпляром или на отдельном компьютере. Оба компьютера соединены сетью, которая обеспечивает обмен данными между двумя процессами. В любом случае суть не меняется - во взаимодействии клиента с базой данных участвуют два процесса. Если оба процесса работают на одной машине, то Oracle применяет локальные механизмы межпроцессной коммуникации (Inter Process Communication, IPC); если на разных, то для коммуникации по сети задействуется подсистема Oracle Net.

# Oracle Net Listener

Этот компонент «прослушивает» сеть в ожидании входящих запросов о соединении с экземплярами. Прослушиватель не является частью экземпляра, он лишь направляет экземпляру запросы о соединении. Прослушиватель запускается и останавливается независимо от экземпляра. Если прослушиватель остановлен, а экземпляр запущен, то клиенты не смогут найти экземпляр в сети, поскольку их некому направить в нужное место. Если, напротив, прослушиватель запущен, а экземпляр остановлен, то клиентов некуда направлять.

Принцип работы прослушивателя довольно прост:

1. Клиент обращается к прослушивателю по сети.
2. Прослушиватель обнаруживает входящий запрос и «знакомит» клиента с серверным процессом.
3. Процедура «знакомства» сводится к информированию клиента и сервера о сетевых адресах друг друга.
4. Прослушиватель самоустраняется, позволяя клиенту и серверу общаться между собой напрямую.

Найдя друг друга, клиент и сервер продолжают общаться напрямую. Прослушиватель больше не нужен.

# Разделяемый/многопоточный сервер

- серверные процессы могут быть выделенными, то есть каждый обслуживает только одного клиента. Следовательно, если с приложением одновременно работают 1000 клиентов, то с экземпляром Oracle будет связано 1000 серверных процессов. Каждый серверный процесс потребляет системные ресурсы, такие как память и время процессора. Для обслуживания множества пользователей может потребоваться очень много ресурсов.

# Разделяемый сервер (shared server)

- Разделяемый сервер позволяет экземпляру Oracle обслуживать многих пользователей с помощью небольшого набора серверных процессов. Теперь каждому клиенту не сопоставляется отдельный выделенный серверный процесс - вместо этого клиенты пользуются разделяемыми серверами. Это позволяет существенно уменьшить общее потребление ресурсов при обслуживании большого числа пользователей.

# Диспетчеры

- Прослушиватель организует соединение между клиентом и сервером, а потом самоустраняется. С этого момента клиент полагается на то, что серверный процесс всегда готов ответить ему. Но разделяемый серверный процесс может в это время обслуживать другого клиента, поэтому клиент соединяется с диспетчером, готовым принять запрос от клиента в любой момент. Для каждого поддерживаемого протокола есть свой диспетчер (например, диспетчер для TCP/IP и так далее). Диспетчеры играют роль выделенных серверов для клиентов. Напрямую клиент соединяется не с сервером, а с диспетчером. Диспетчер принимает запросы от клиента и помещает их в очередь запросов - структуру, находящуюся в SGA. Для каждого экземпляра имеется только одна очередь запросов.

# Разделяемые серверы

- Разделяемый серверный процесс читает запрос из очереди, обрабатывает его и помещает результат в очередь ответов соответствующего диспетчера. Для каждого диспетчера имеется ровно одна очередь ответов. Диспетчер читает результаты из очереди и посылает их клиентскому процессу.
- Есть пул диспетчеров и пул разделяемых серверов.

# Общий алгоритм работы

1. Клиент обращается по сети к прослушивателю.
2. Прослушиватель обнаруживает входящий запрос и, анализируя конфигурацию Oracle Net, понимает, что запрос адресован многопоточному серверу. Поэтому он передает клиента не выделенному серверу, а диспетчеру того протокола, которым воспользовался клиент.
3. Прослушиватель «знакомит» клиента с диспетчером, сообщая каждому сетевой адрес другой стороны.
4. Теперь клиент и диспетчер знают, как найти друг друга, и далее общаются напрямую. Прослушиватель больше не нужен. Клиент посылает запросы напрямую диспетчеру.
5. Диспетчер помещает запрос клиента в очередь запросов в SGA.
6. Оказавшийся свободным разделяемый серверный процесс извлекает запрос из очереди и обрабатывает его.
7. Разделяемый сервер помещает результат обработки клиентского запроса в очередь ответов того диспетчера, которому изначально поступил запрос.
8. Диспетчер извлекает результат из очереди.
9. Диспетчер посылает результат клиенту.

# Фрагментация таблиц в Oracle

Фрагментация может отрицательно сказаться на производительности, поэтому в прошлом многие администраторы всеми силами боролись с ней. Нежелательный эффект фрагментации - появление небольших кусочков несмежного «свободного пространства», которые невозможно использовать повторно. В Oracle набор смежных блоков называется экстентом, а набор экстентов - сегментом. В сегментах можно хранить все, что угодно: таблицу, индекс или сегмент отката. Как правило, сегмент состоит из нескольких экстентов. Когда один сегмент оказывается заполнен, система начинает использовать следующий экстент в сегменте. По мере того как в результате работы базы данных в непрерывной области, занятой экстентом, появляются «дырки» (это и называется фрагментацией), количество экстентов в сегменте растет. Чем сильнее фрагментирован сегмент, тем больше приходится выполнять операций ввода/вывода, что снижает производительность.

- В СУБД Oracle9i для устранения фрагментации обычно применялась оперативная реорганизация с помощью команды CREATE TABLE... AS SELECT. Иными словами, содержимое исходной таблицы копировалось в новую, без запрещения обновлений исходной таблицы. Все изменения, внесенные во время копирования, запоминались, а потом применялись к новой таблице. В ходе этой операции можно было изменять физические и логические атрибуты таблицы, что и позволяло выполнять оперативную реорганизацию.

- Ввод автоматизированной дефрагментации и уплотнения сегментов без прерывания доступа к данным в версии Oracle Database 10g и последующих существенно упростил решение этой проблемы. В результате постоянно поддерживаются условия для достижения оптимальной производительности.

# Резервное копирование и восстановление

- Даже если принять все меры предосторожности, критически важные записи иногда могут пропасть из базы данных в результате ошибки человека или программного либо аппаратного сбоя. Единственный способ подготовиться к такому развитию событий - регулярно выполнять резервное копирование.
- К потере данных в базе Oracle могут привести две причины: сбой экземпляра, когда экземпляр останавливается без выполнения должной процедуры, и отказ носителя, когда повреждаются диски, на которых хранится информация.
- В первом случае Oracle автоматически выполняет процедуру восстановления после сбоя. Например, Real Application Clusters самостоятельно выполнит восстановление после сбоя любого экземпляра.

# Администратор должен выполнять следующие действия:

- мультиплексировать оперативные журналы, располагая копии на разных дисках, подключенных к разным контроллерам;
- эксплуатировать базу данных в режиме ARCHIVELOG, чтобы журналы архивировались перед повторным использованием;
- хранить архивные журналы в разных местах;
- поддерживать несколько копий управляющих файлов;
- часто выполнять резервное копирование физических файлов данных и, в идеале, хранить несколько копий в разных местах.

- Эксплуатация базы в режиме ARCHIVELOG гарантирует возможность восстановления в состоянии, непосредственно предшествующем моменту отказа. При этом администратор может производить оперативное резервное копирование, не прекращая доступ к данным. Кроме того, архивные журналы можно применять к резервной базе данных.
- Компонент Recovery Manager (RMAN), впервые появившийся в версии Oracle8 и с тех пор значительно усовершенствованный, предоставляет удобный интерфейс для выполнения этой процедуры. Ныне к RMAN можно обращаться из Enterprise Manager.

# Типы резервного копирования и восстановления

Есть два основных типа резервного копирования.

- Полное резервное копирование. Могут копироваться отдельные файлы данных, табличные пространства, управляющие файлы (текущие или архивные) или вся база целиком (включая все файлы данных и текущий управляющий файл). В набор резервных копий включаются все блоки данных за исключением тех, что никогда не использовались (к управляющим файлам и журналам это не относится, для них никакие блоки не пропускаются).
- Инкрементное резервное копирование. Могут копироваться отдельные файлы данных, табличные пространства или вся база целиком. Включаются только блоки данных, изменившиеся с момента снятия предыдущей копии.

- Запустить процедуру резервного копирования можно с помощью Recovery Manager или из интерфейса Enterprise Manager к RMAN - в обоих случаях применяется механизм резервного копирования базы данных. А можно воспользоваться стандартными средствами резервного копирования из операционной системы.

- RMAN поддерживает большинство возможностей резервного копирования базы,
- включая копирование открытой базы (оперативное), закрытой базы,
- инкрементные копии на уровне блоков Oracle, обнаружение заперченных блоков,
- автоматическое резервное копирование,
- каталоги резервных копий и копирование на носители с последовательным доступом.
- В версии Oracle9i к RMAN добавлена возможность задать одноразовую конфигурацию резервного копирования, окна восстановления для задания и управления датой истечения срока хранения резервных копий и возможность перезапуска процессов резервного копирования и восстановления. Кроме того, включена поддержка восстановления в тестовом режиме.

- В версии Oracle Database 10g RMAN умеет копировать образы базы данных, табличных пространств или файлов данных и применять инкрементные резервные копии к образам файлов данных. Скорость инкрементного копирования повышена за счет механизма отслеживания изменений, когда считываются и помещаются в копию только изменившиеся блоки.

- Чтобы выработать адекватную стратегию резервного копирования и восстановления, нужно обязательно симитировать восстановление после сбоя на тестовой системе и только потом заниматься восстановлением живой промышленной базы данных. Часто выясняется, что носители, которые вы считали надежными, на деле такими не оказываются, или что частота копирования, казавшаяся вам достаточной, не обеспечивает восстановления в отведенные сроки. Будет куда лучше, если низкая скорость или невозможность восстановления проявится в условиях теста, чем подвергать бизнес риску из-за сбоя реальной системы.

# Безопасность

Управление безопасностью обычно осуществляется на трех уровнях:

- уровень базы данных;
- уровень операционной системы;
- сетевой уровень.

# Уровень операционной системы

На уровне операционной системы у администратора базы данных (АБД) должны быть права для создания и удаления относящихся к базе данных файлов. Напротив, у обычных пользователей таких прав быть не должно.

# Специальные роли - DBA, SYSDBA и SYSOPER

- В базе данных уже predeterminedены три специальные роли. Одна из самых важных - роль DBA. В нее включено большинство системных привилегий. По умолчанию она назначается пользователям SYS и SYSTEM, которые создаются на этапе инициализации базы данных.
- В схеме SYS хранятся таблицы и представления словаря данных.
- Таблицы из схемы SYSTEM используются для хранения административной информации, а также различными инструментами и опциями Oracle.

- Роль DBA не включает привилегий для выполнения основных административных задач, подразумеваемых системными привилегиями SYSDBA или SYSOPER.
- Поэтому привилегии SYSDBA или SYSOPER должны быть явно предоставлены администраторам. Подключаясь к базе данных, они выполняют команду CONNECT AS, указывая имя SYSDBA или SYSOPER, и могут таким образом получить доступ даже к закрытой базе данных.

# Привилегия SYSDBA

- Позволяет выполнять перечисленные ниже действия из командной строки в SQL\*Plus или с помощью графического интерфейса Oracle Enterprise Manager:
- **STARTUP** Запуск экземпляра базы данных.
- **SHUTDOWN** Останов экземпляра базы данных.
- **ALTER DATABASE OPEN** Открытие смонтированной, но еще не открытой базы данных.

- **ALTER DATABASE MOUNT** Монтирование базы данных для уже запущенного экземпляра.
- **ALTER DATABASE BACKUP CONTROLFILE** Запуск резервного копирования управляющего файла.
- **ALTER DATABASE ARCHIVELOG** Включение режима архивирования содержимого группы журналов перед повторным использованием этой группы.
- **ALTER DATABASE RECOVER** Применение журналов по отдельности или запуск автоматической процедуры применения журналов.
- **CREATE DATABASE** Создание базы данных с указанным именем, определение местоположения и размеров файлов данных и журналов, а также предельных значений параметров.

- **DROP DATABASE** Удаление базы данных и всех файлов, перечисленных в управляющем файле.
- **CREATE SPFILE** Создание файла параметров сервера из текстового описания параметров (INIT.ORA).
- **RESTRICTED SESSION** Разрешение соединения с базой данных, запущенной в ограниченном режиме. Ограниченный режим предназначен для поиска и устранения неполадок и некоторых видов обслуживания, аналогичных тем, что разрешены пользователю SYS.

- Администраторам, подключившимся от имени SYSOPER, доступно меньше команд: **STARTUP** и **SHUTDOWN**, **CREATE SPFILE**, **ALTER DATABASE OPEN**, или **MOUNT**, или **BACKUP**, **ALTER DATABASE ARCHIVELOG**, **ALTER DATABASE RECOVER**. Кроме того, им предоставлена привилегия **RESTRICTED SESSION**.

- Администраторы базы данных аутентифицируются с помощью механизмов операционной системы или файла паролей. Если применяется аутентификация с помощью операционной системы, то имена пользователей с правами администратора должны входить в группы OSDBA или OSOPER. Парольный файл для аутентификации создается утилитой ORAPWD. Новых пользователей может добавлять пользователь SYS или любой, у кого есть привилегия SYSDBA.

# Заключение