

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

Дисциплина: Основы построения защищенных баз данных.

Лабораторная работа №8.

Администрирование Oracle. Мультиплексирование управляющих файлов, журнальных групп, обеспечение создания избыточных копий архивных журналов.

1. Учебные цели:

- Отработать вопросы восстановления экземпляра Oracle в зависимости от типа сбоя, конфигурирования режима ARCHIVELOG.
- Освоить приемы настройки мультиплексирования управляющих файлов, журнальных групп, обеспечения создания избыточных копий архивных журналов.

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 СУБД Oracle для восстановления экземпляра Oracle в зависимости от типа сбоя, методами настройки мультиплексирования управляющих файлов, журнальных групп, обеспечения создания избыточных копий архивных журналов.

3. Перечень материально-технического обеспечения

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

4. Краткие теоретические 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.

В обязанности администратора входят:

- Защита базы данных от всех типов сбоев
- Увеличение среднего времени безотказной работы (MTBF)
- Уменьшение среднего времени восстановления (MTTR)
- Минимизация потерь данных

Задачи резервирования и восстановления

Целью администратора базы данных (АБД) является поддержка базы данных в доступном для использования состоянии. Для этого АБД обычно вместе с системным администратором:

- Осуществляет профилактические меры по устранению общих причин сбоев.
- Добивается увеличения *среднего времени безотказной работы (mean-time-between-failure – MTBF)* за счет поддержания оборудования в надежном состоянии, используя избыточность важных компонентов и своевременно проводя в операционной системе работы

по сопровождению. Oracle предоставляет дополнительные опции, позволяющие увеличить MTBF, включающие:

- Real Application Clusters (рассматривается в курсе *Oracle Database 10g: Real Application Clusters*);

- Streams (рассматривается в курсе *Oracle Database 10g: Implement Streams*).

- Снижает *среднее время восстановления (mean time to recover – MTTR)*, применяя усовершенствованные процедуры восстановления и конфигурируя резервные объекты так, чтобы они были без задержки доступны, когда это необходимо.

- Минимизирует потери данных. АБД, применяющие наилучшие методы, могут сконфигурировать базу данных так, чтобы даже незафиксированные транзакции не терялись. В этом им помогают:

- Архивные журнальные файлы (рассматриваются далее)
- Резервные базы данных (*standby databases*) и Oracle Data.

Виды сбоев

Сбой можно разделить на несколько общих видов.

- **Сбой команды:** ошибочное завершение одной операции в БД (*select, insert, update, delete*).

- **Сбой пользовательского процесса:** аварийное завершение сеанса БД.

- **Сбой сети:** потеря соединения с базой данных.

- **Ошибка пользователя:** пользователь успешно завершил операцию, однако действие было неправильным (удаление таблицы, ввод некорректных данных).

- **Сбой экземпляра:** непредвиденная остановка экземпляра базы данных.

- **Сбой носителя:** один или несколько файлов базы данных потеряны (удалены или сломан диск).

Сбой команды

Типичные причины	Возможные решения
Попытка ввода неверных данных в таблицу	Совместная работа с пользователем по проверке и корректировке данных
Попытка выполнить операцию с недостаточными привилегиями	Предоставить необходимые объектные и системные привилегии
Неудачная попытка выделения пространства	Включить возможность возобновления операции после неудачного выделения пространства. Увеличить квоты пользователя. Увеличить табличное пространство.
Логическая ошибка в приложении	Взаимодействие с разработчиками по исправлению программных ошибок

Сбой пользовательского процесса

Типичные причины	Возможные решения
Пользователь неправильно отсоединился от базы данных	Администратору базы данных редко приходится исправлять ошибки после сбоя пользовательского процесса. Фоновые процессы откатывают незафиксированные изменения и освобождают блокировки. АБД должен выявлять тенденции.
Пользовательский сеанс завершился аварийно	
Пользовательская программа завершилась с ошибкой, что вызвало аварийное завершение сеанса.	

Сбой сети

Типичные причины	Возможные решения
Аварийное завершение процесса прослушивания	Сконфигурируйте резервный процесс прослушивания и перенаправление соединения при сбоях
Неисправность сетевой карты	Сконфигурируйте несколько сетевых карт
Аварийное завершение сетевого соединения	Сконфигурируйте резервное сетевое соединение

Ошибка пользователя

Типичные причины	Возможные решения
Пользователь случайно удалил или изменил данные в таблице	Откатите изменения или примените запрос прошлых данных для восстановления
Пользователь уничтожил таблицу	Восстановите таблицу, используя корзину удаленных объектов

Пользователь может непреднамеренно удалить или изменить данные в таблице. Когда такое случается, АБД должен помочь пользователю восстановить данные. Если пользователь еще не зафиксировал изменения и не вышел из своей программы, тогда просто откатите его операции. Если же пользователь уже зафиксировал изменения, можно выполнить *запрос прошлых данных (flashback queries)* для выяснения значений, которые были до изменения (и затем изменить данные, восстановив их старые значения).

```
SQL> SELECT salary FROM employees WHERE employee_id=100;
```

```
SALARY
```

```
-----
```

```
25
```

```
SQL> SELECT salary FROM employees  
AS OF TIMESTAMP(SYSTIMESTAMP-INTERVAL'10' minute)  
WHERE employee_id=100;
```

```
SALARY
```

```
-----
```

```
24000
```

В случае, когда запрос прошлых данных невозможен из-за того, что истек срок удержания информации отмены, АБД все же может восстановить первоначальные данные с помощью Oracle LogMiner.

Oracle LogMiner позволяет запросить данные из оперативных и архивных журнальных файлов с помощью интерфейса ввода SQL. Данные транзакций могут храниться в оперативных журналах дольше, чем в пространстве отмены, а также находиться в архивных журналах (если сконфигурировано их использование), пока те не будут удалены.

Oracle LogMiner рассматривается в документе *Oracle Database: Utilities*.

Используя *корзину удаленных объектов (recycle bin)*, пользователи, удалившие таблицу, могут ее восстановить такой, какой она было перед удалением. Дополнительные сведения см. в теме “Флэшбэк”.

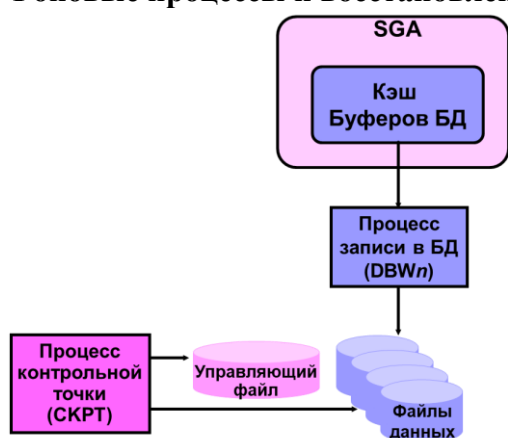
Если корзина удаленных объектов уже очищена или пользователь удалил таблицу с опцией PURGE, уничтоженная таблица *все еще* может быть восстановлена с помощью метода *восстановления на момент в прошлом (point-in-time recovery – PITR)*, если это соответствующим образом сконфигурировано в базе данных.

PITR рассматривается в документе *Oracle Database: Backup and Recovery Advanced User's Guide*.

Сбой экземпляра

Типичные причины	Возможные решения
Прекращение подачи электроэнергии	<p>Перезапустите экземпляр с помощью команды STARTUP. Восстановление после сбоя экземпляра происходит автоматически и включает шаги подката вперед изменений на основе оперативных журналов и последующего отката незафиксированных транзакций.</p> <p>Изучите причины сбоев, используя сигнальный файл, трассировочные файлы и Enterprise Manager.</p>
Аппаратный сбой	
Сбой одного из фоновых процессов	
Аварийная остановка экземпляра	

Фоновые процессы и восстановление: процесс контрольной точки (СКРТ)



СКРТ ответственен за:

- Передачу сигнала процессу DBWn о контрольной точке
- Обновление информации о контрольной точке в заголовках файлов данных
- Обновление информации о контрольной точке в управляющих файлах

Для понимания того, что происходит при восстановлении после сбоя экземпляра, необходимо разобраться, как работают некоторые фоновые процессы.

Каждые три секунды (или чаще), процесс СКРТ заносит данные в управляющий файл, чтобы документально отметить, какие блоки данных процесс DBWn записал из SGA на диск. Это называется “контрольной точкой” (“*checkpoint*”). Цель контрольной точки – обозначить место в оперативном журнальном файле, откуда начнется восстановление экземпляра после сбоя (это место называется “позицией контрольной точки”).

Когда происходит событие переключения журнала, процесс СКРТ также записывает информацию контрольной точки в заголовки файлов данных.

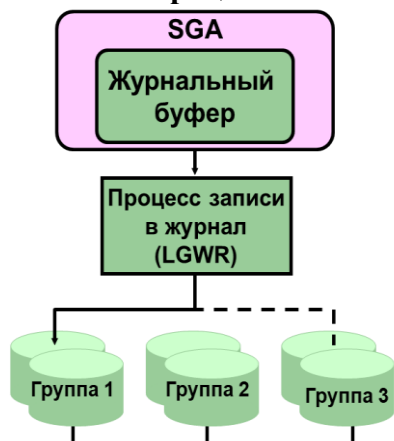
Контрольная точка производится по следующим причинам:

- Необходимо гарантировать регулярную запись на диск измененных в памяти блоков данных, так чтобы данные не потерялись в случае сбоя системы или базы данных.
- Необходимо уменьшить время, требуемое для восстановления экземпляра. Только записи оперативного журнала, расположенные за последней контрольной точкой, обрабатываются при восстановлении.
- Необходимо гарантировать, что все зафиксированные данные были записаны в файлы данных, в ходе остановки экземпляра БД.

Информация контрольной точки, записываемая процессом СКРТ, включает позицию (номер) контрольной точки, *системный номер изменений* (*system change number*), место в оперативном журнальном файле для начала восстановления, сведения о журналах и т.д.

Примечание: процесс СКРТ не пишет блоки данных на диск, а также журнальные блоки в оперативные журнальные файлы.

Фоновые процессы и восстановление: оперативные журналы и процесс записи в журнал



Оперативные журналы:

- В них записываются изменения, сделанные в БД
- Журнальные файлы необходимо мультимплексировать, чтобы защититься от их потери

LGWR пишет в журнал:

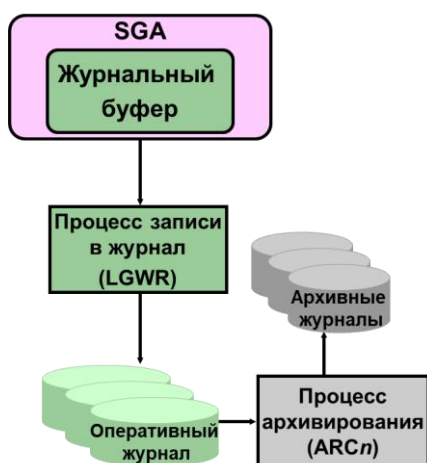
- При фиксации транзакции
- Когда журнальный буфер заполнен на одну треть
- Когда более 1 MB журнальных записей
- Каждые три секунды
- Перед записью процессом DBWn модифицированных блоков в файлы данных

В оперативные журнальные файлы повторного выполнения (*redo log files*) пишутся записи об изменениях, выполняемых в базе данных транзакциями и внутренними операциями сервера Oracle. (Транзакция – логическая единица работы, состоящая из одной или нескольких команд SQL, выполняемых пользователем). Оперативные журналы позволяют восстановить целостность базы данных после системных сбоев, вызванных прекращением подачи электроэнергии, дисковых сбоев и т.п. Оперативные журнальные файлы необходимо мультимплексировать, чтобы гарантировать сохранность располагаемой в них информации в случае дисковых сбоев.

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

Подсказка по наилучшему практическому методу работы: если это возможно, располагайте мультимплексируемые оперативные журнальные файлы на различных дисках.

Фоновые процессы и восстановление: процесс архивирования (ARCn)



- Необязательный фоновый процесс
- Автоматически архивирует оперативные журнальные файлы, если в базе данных установлен режим ARCHIVELOG
- Сохраняет записи обо всех изменениях, сделанных в базе данных

ARCn – необязательный фоновый процесс. Однако он необходим для восстановления потерянных данных из-за сбоев дисков. Как только текущие журнальные файлы заполнятся, Oracle начинает писать в следующие доступные журнальные файлы. Этот процесс называется переключением (log switch). Процесс ARCn запускает резервирование (или архивирование) группы заполненных журнальных файлов после переключения журнальных файлов. Он

автоматически архивирует журналы до их повторного использования, сохраняя все изменения, произведенные в базе данных. Это предоставляет возможность администратору восстановить базу данных на определенный момент времени, даже в случае выхода из строя дисков.

Одно из важных решений, которое должен принять АД, состоит в выборе между двумя режимами работы базы данных: ARCHIVELOG или NOARCHIVELOG.

- В режиме NOARCHIVELOG информация в журнальных файлах перезаписывается после переключения.

- Для базы данных, сконфигурированной в режиме ARCHIVELOG, неактивная группа заполненных журнальных файлов должна быть архивирована перед повторным использованием.

Примечание: режим ARCHIVELOG требуется для реализации большинства стратегий резервирования (и его очень просто сконфигурировать).

Переведите БД в режим ARCHIVELOG используя SQL*Plus:

Остановите экземпляр.

```
SQL> shutdown immediate
```

Database closed.

Database dismounted.

ORACLE instance shut down.

Запустите экземпляр в режиме MOUNT.

```
SQL> STARTUP MOUNT
```

ORACLE instance started.

Total System Global Area 663908352 bytes

Fixed Size 2291904 bytes

Variable Size 285214528 bytes Database

Buffers 369098752 bytes

Redo Buffers 7303168 bytes

Database mounted.

Установите режим ARCHIVELOG.

```
SQL> ALTER DATABASE ARCHIVELOG;
```

Database altered.

Откройте БД.

```
SQL> ALTER DATABASE OPEN;
```

Database altered.

Используйте команду ARCHIVE LOG LIST чтобы убедиться, что БД работает в режиме ARCHIVELOG.

```
CA. Command Prompt - sqlplus "/as sysdba"
SQL> archive log list
Database log mode                Archive Mode
Automatic archival               Enabled
Archive destination              USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence       79
Next log sequence to archive     81
Current log sequence             81
SQL>
```

В режиме ARCHIVELOG БД будет постоянно создавать архивные копии оперативных журналов транзакций. Учитывайте дополнительные требования к дисковому пространству.

Создайте дополнительное место хранения архивных журналов: «C:\app\IEUser\archive_dir2»

Установите значения параметра «LOG_ARCHIVE_DEST_1 на путь к FRA» и параметра «LOG_ARCHIVE_DEST_2» на новую директорию.

```
SQL> ALTER SYSTEM SET
      log_archive_dest_1='LOCATION=C:\app\IEUser\fast_recovery_area';
SQL> ALTER SYSTEM SET
      log_archive_dest_2= 'LOCATION= C:\app\IEUser\archive_dir2';
```

Проверьте правильность внесенных изменений запросом к V\$ARCHIVED_LOG.

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
      System altered.
SQL> ALTER SYSTEM SWITCH LOGFILE;
      System altered.
SQL> ALTER SYSTEM SWITCH LOGFILE;
      System altered.
SQL> SELECT name from V$ARCHIVED_LOG ORDER BY STAMP;
```

```
CA. Command Prompt - sqlplus

SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.
SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.
SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.
SQL> SELECT name from U$ARCHIVED_LOG ORDER BY STAMP;
NAME
-----
C:\APP\IEUSER\FAST_RECOVERY_AREA\ARC0000000081_0980752503.0001
C:\APP\IEUSER\FAST_RECOVERY_AREA\ARC0000000082_0980752503.0001
C:\APP\IEUSER\ARCHIVE_DIR2\ARC0000000082_0980752503.0001
C:\APP\IEUSER\FAST_RECOVERY_AREA\ARC0000000083_0980752503.0001
C:\APP\IEUSER\ARCHIVE_DIR2\ARC0000000083_0980752503.0001
C:\APP\IEUSER\FAST_RECOVERY_AREA\ARC0000000084_0980752503.0001
C:\APP\IEUSER\ARCHIVE_DIR2\ARC0000000084_0980752503.0001

7 rows selected.

SQL>
```

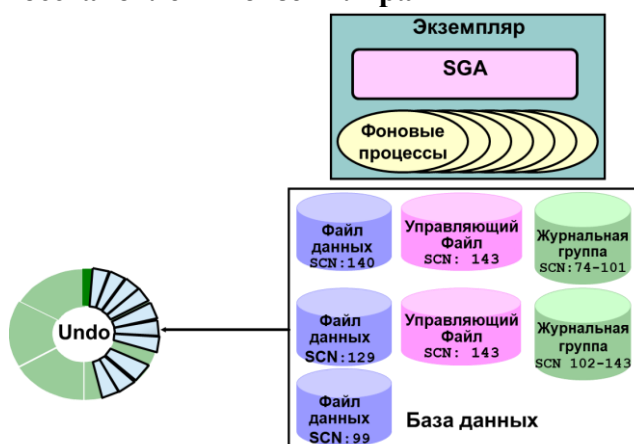
Восстановление после сбоя экземпляра

Восстановление после сбоя или отказа экземпляра:

- вызывается необходимостью открыть базу данных, файлы которой не были синхронизированы при остановке экземпляра
- выполняется автоматически
- для синхронизации файлов используется информация, хранимая в журнальных группах
- включает в себя две отдельные операции:
 - подкат вперед (восстановление состояния файлов данных на момент перед сбоем экземпляра);
 - откат назад (данные незафиксированных изменений возвращаются в первоначальное состояние).

При этом используется информация в группах оперативных журналов, которая позволяет также подкатить вперед и файлы табличного пространства отмены, чтобы затем откатить незафиксированные транзакции.

Фазы восстановления экземпляра



Чтобы экземпляр смог открыть файл данных, *системный номер изменений* (*system change number – SCN*) в заголовке файла данных должен соответствовать текущему SCN в управляющих файлах базы данных.

Если это не так, экземпляр применяет *оперативные журнальные файлы* (*online redo logs*) последовательно “повторно выполняя” транзакции, пока файлы данных не будут синхронизированы с управляющими файлами. Сразу после этого открывается база данных, и пользователи могут к ней подсоединяться.

Когда применяются записи повторного выполнения (*redo*) оперативного журнала, *все* транзакции накатываются до момента сбоя. В их число входят и те, которые не были зафиксированы. После открытия базы данных для таких незафиксированных транзакций выполняется откат. В конце фазы отката все файлы содержат только зафиксированные данные.

Настройка процесса восстановления после сбоя экземпляра

Для восстановления транзакций информация между позицией контрольной точки и концом журнала применяется к файлам данных.

Настраивайте процесс восстановления, контролируя расстояние между позицией контрольной точки и концом журнала



До того, как экземпляр вернет сообщение о завершении транзакции (commit complete), информация о ней всегда записывается в журнальные группы. Эта информация гарантирует, что транзакции могут быть восстановлены в случае сбоя. Информация той же самой транзакции также должна быть записана в файлы данных. Такая запись данных завершается обычно через какое-то время после того, как данные были помещены в журнальную группу, так как процесс записи в файл данных обычно более медленный, чем запись в оперативный журнал (запись в файлы данных, производимая в случайном порядке, медленнее, чем последовательная запись в журнальные файлы).

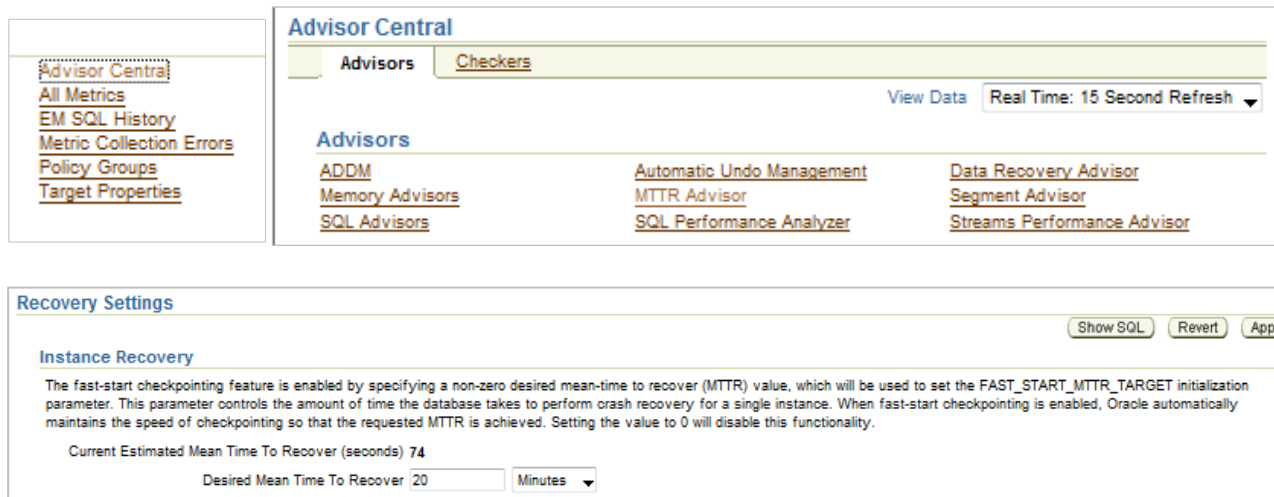
Каждые три секунды процесс контрольной точки записывает информацию в управляющий файл о позиции контрольной точки в журнале. Поэтому база данных Oracle знает, что все журнальные записи (входы), записанные до этой позиции, не потребуются для восстановления базы данных. На рисунке розовые блоки содержат такие не записанные еще данные.

Время, необходимое для восстановления экземпляра, – это время для перевода файла из состояния его последней контрольной точки к состоянию, соответствующему самому последнему SCN, хранимому в управляющем файле. Администратор контролирует это время, задавая целевое значение для MTTR (в секундах) и создавая журнальные файлы определенного размера.

Расстояние между позицией контрольной точки и концом журнала никогда не должно превышать 90% размера наименьшего оперативного журнального файла.

Использование консультанта MTTR

- Задайте требуемое время (desired time) в секундах и минутах.
- Значение по умолчанию – 0 (отключение использования MTTR).
- Максимальное значение – 3600 секунд (1 час).



Консультант позволяет задать требуемое *среднее время восстановления (mean-time-to-recover)*. Чтобы получить подсказку при установлении целевого значения MTTR, перейдите следующим образом с домашней страницы в Enterprise Manager: Administration > Advisor Central > MTTR Advisor. Консультант преобразует величину параметра FAST_START_MTTR_TARGET в значения нескольких параметров, позволяющие выполнить восстановление экземпляра за требуемое или меньшее время, если это возможно.

Явное задание значения 0 для параметра FAST_START_MTTR_TARGET отключает автоматическую настройку контрольной точки.

Явное задание для параметра FAST_START_MTTR_TARGET значения, отличного от нуля, включает *консультант оперативного журнала (Redo Log Advisor)*.

Устанавливаемое значение параметра FAST_START_MTTR_TARGET должно удовлетворять соглашению об уровне обслуживания для вашей системы. Установка слишком

малой величины для целевого значения MTTR приводит к росту нагрузки на систему со стороны операций ввода-вывода из-за выполнения дополнительных операций записи в файлы данных (что влияет на производительность). Установка же слишком большого значения MTTR означает, что экземпляру понадобится больше времени, чтобы восстановиться после аварийного завершения.

Сбой носителя

Типичные причины	Возможные решения
Поломка дисководов	1. Восстановление соответствующих файлов из резерва. 2. Изменение в базе данных указателя расположения файла, когда это необходимо. 3. При необходимости восстановление файла путем применения журнальных данных.
Поломка дискового контроллера	
Удален или поврежден файл базы данных	

Oracle подразумевает под сбоем носителя любой сбой, приведший в результате к потере или повреждению одного или нескольких файлов базы данных (файла данных, управляющего файла или оперативного журнального файла).

При восстановлении необходимо скопировать из резерва и восстановить потерянные файлы. Используйте рекомендуемые приемы, кратко изложенные далее. Они обеспечивают возможность восстановления базы данных после сбоя носителя.

Конфигурирование, способствующее восстанавливаемости

Для обеспечения наилучшей защиты данных:

- **Планируйте проведение регулярного резервирования.** В большинстве случаев после сбоя носителя необходимо скопировать из резерва потерянный или поврежденный файл.
- **Мультиплексируйте управляющие файлы.** Все управляющие файлы одной и той же БД одинаковые. Не очень сложно восстановиться после потери одного управляющего файла. Более сложно – после потери *всех* управляющих файлов. Защититься от потери всех управляющих файлов можно, если иметь много копий (по крайней мере, три).
- **Мультиплексируйте группы оперативных журналов.** Информация повторного выполнения используется при восстановлении после сбоя экземпляра или носителя для подката вперед файлов данных до момента завершения последней фиксируемой транзакции. Когда группа журналов содержит один файл, тогда при его потере нельзя восстановиться до конца и данные теряются. Обеспечьте наличие хотя бы трех файлов в каждой журнальной группе.
- **Хранение архивных копий оперативных журналов.** При потере файла его копию восстанавливают из резерва и применяют к ней информацию повторного выполнения, подкатывая копию вперед до самого последнего SCN, содержащегося в управляющем файле. В конфигурации по умолчанию журнальные данные могут быть переписаны сразу после того, как они попали в файлы данных. БД можно сконфигурировать так, чтобы журнальная информация копировалась и сохранялась в архивных копиях. Для этого базу данных надо перевести в режим ARCHIVELOG .

С помощью консультанта MTTR в Enterprise Manager задайте время среднее время восстановления (mean-time-to-recover) равным 5 минут.

Управляющие файлы

Управляющий файл – это небольшой двоичный файл, который содержит описание структуры базы данных. Когда база данных смонтирована или открыта, он должен быть

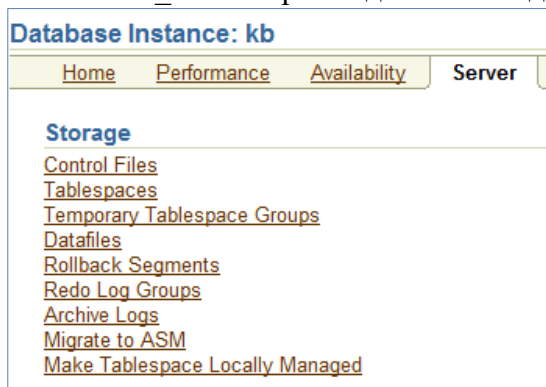
доступен серверу Oracle для записи. Без него база данных не может быть смонтирована, и поэтому может потребоваться восстановление или пересоздание управляющего файла. В базе данных следует иметь два управляющих файла (предпочтительнее три) на разных дисках, чтобы минимизировать последствия потери одного управляющего файла.

База данных, созданная с помощью утилиты DBCA, содержит три управляющих файла.

Потеря одного управляющего файла вызовет отказ экземпляра, поскольку все управляющие файлы должны быть все время доступны. Однако восстановиться можно простым копированием одного из неповрежденных управляющих файлов. При потере всех управляющих файлов восстановиться немного сложнее, но обычно это не катастрофическая ситуация.

Дублирование управляющего файла

Oracle рекомендует использовать по крайней мере два идентичных файла, хранящихся на разных дисках (при установке Oracle они по умолчанию создаются на одном диске, см. рис. ниже). Если управляющий файл утерян, использование его копии позволяет перезапустить экземпляр без необходимости восстановления базы данных. База данных может иметь до восьми полноценных управляющих файлов, что задается параметром инициализации CONTROL_FILES при создании базы данных.



Control File Mirror Images		
Oracle strongly recommends that your database has a minimum of two control files and that they are located on separate disks. If a control file is damaged due to a disk failure, it could be restored using the intact copy of the control file from the other disk. You can specify their location in the database's initialization parameter file.		
Valid	File Name	File Directory
VALID	CONTROL01.CTL	C:\APP\IEUSER\ORADATA\KB\
VALID	CONTROL02.CTL	C:\APP\IEUSER\FAST_RECOVERY_AREA\KB\

Администратор может дублировать управляющий файл следующим образом:

- создать нескольких копий управляющих файлов – включить имена управляющих файлов в параметр инициализации CONTROL_FILES во время создания базы данных:
CONTROL_FILES=\$HOME/ORADATA/u01/ctrl01.ctl,
\$HOME/ORADATA/u02/ctrl02.ctl

Добавить управляющий файл после создания базы данных. Далее приведены шаги по добавлению управляющего файла для случая, когда используется файл серверных параметров (SPFILE – файл в двоичном виде, поэтому править можно только командами SQL).

1. Измените SPFILE

пример Linux:

```
SQL> ALTER SYSTEM SET control_files =  
'$HOME/ORADATA/u01/ctrl01.ctl',  
'$HOME/ORADATA/u02/ctrl02.ctl' SCOPE=SPFILE;
```

Пример средствами EM (Windows):

Select	Name	Help	Revisions	Value	Comments
<input checked="" type="radio"/>	control_file_record_keep_time	?			
<input type="radio"/>	control_files	?		'C:\app\IEuser\fast_recovery_area\kb\control02.ctl', 'C:\app\IEuser\oradata\kb\control01.ctl'	
<input type="radio"/>	control_management_pack_access				

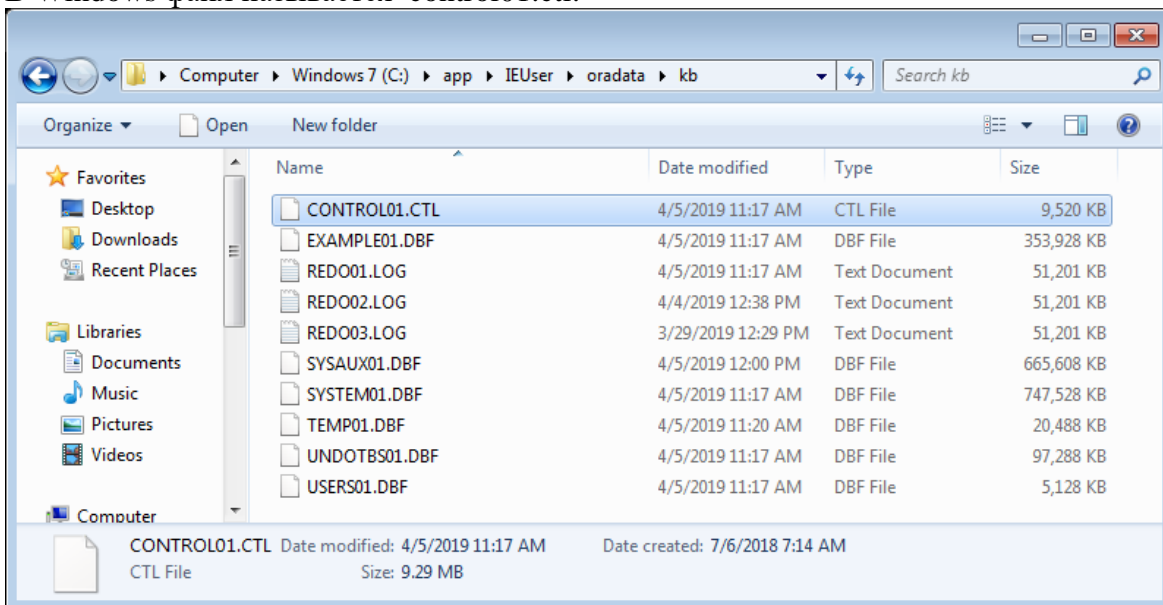
2. **Остановите** БД в нормальном режиме:

```
SQL> shutdown
```

3. Создайте копии управляющего файла (пример Linux) в каталогах, указанных в SPFILE.

```
$ cp $HOME/ORADATA/u01/ctrl01.ctl $HOME/ORADATA/u02/ctrl02.ctl
```

В Windows файл называется control01.ctl:



4. Запустите базу данных:

```
SQL> startup
```

В настройках виртуальной машины добавьте дополнительный диск (авторасширяемый), размером 1 GB.

Измените SPFILE – добавьте описание еще одного контрольного файла CONTROL03.CTL, размещенного в каталоге CONTROL на созданном диске виртуальной машины.

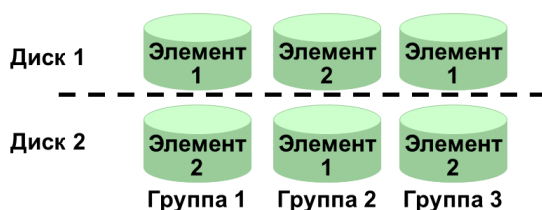
Остановите базу данных.

Скопируйте контрольный файл CONTROL01.CTL в каталоге CONTROL. Переименуйте его в CONTROL03.CTL.

Запустите базу данных.

Проверьте состояние контрольных файлов (Server-Storage-Control Files), должно быть VALID

Оперативные журнальные файлы



Журнальные группы (*redo log groups*) состоят из одного или нескольких журнальных файлов (*журнальных файлов повторного выполнения, redo log files*). Каждый файл внутри группы – точная копия других элементов группы. Oracle рекомендует иметь, по крайней мере, два файла на группу, а также распределять их по разным дискам/контроллерам так, чтобы отказ какого-то одного устройства не повредил всю журнальную группу.

Потеря всей журнальной группы – одна из наиболее серьезных среди возможных потерь, вызываемых сбоем носителя, поскольку в результате могут быть утрачены данные. Потеря одного элемента группы не оказывает никакого влияния на работоспособность базы данных, только соответствующее сообщение помещается в сигнальный файл. Восстановление после потери одного журнального файла будет рассмотрено далее. При восстановлении после потери всей журнальной группы применяются специальные способы, рассматриваемые в курсе *Oracle Database 10g: Администрирование II*.

Журналы обычно сильно влияют на производительность, так как фиксация не завершается до тех пор, пока информация транзакции не запишется в журнальные файлы. Такие файлы следует размещать на самых быстрых дисках, подсоединенных к наиболее скоростным контроллерам. Если это возможно, не располагайте другие файлы базы данных и журналы на одних и тех же дисках. Размещение элементов разных групп на одном диске не влияет на производительность, поскольку в каждый момент времени запись идет только в одну группу.

Мультиплексирование журнала

Redo Log Groups

Object Type: Redo Log Group

Search

Enter an object name to filter the data that is displayed in your results set.

Object Name

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode:

Actions:

Select	Group	Status	# of Members	Archived	Size (KB)	Sequence	First Change#
<input checked="" type="radio"/>	1	Inactive	1	No	51200	10	1204553
<input type="radio"/>	2	Inactive	1	No	51200	11	1228453
<input type="radio"/>	3	Current	1	No	51200	12	1233138

Edit Redo Log Group: 1

Group # 1

File size 51200 KB

Status INACTIVE

Redo Log Members

Select	File Name	File Directory
<input checked="" type="radio"/>	REDO01.LOG	C:\APP\IEUSER\ORADATA\KB\

Actions:

Чтобы мультиплексировать журнал, добавьте файл в существующую журнальную группу. Выполните следующие шаги по добавлению члена журнальной группы. При этом база данных может быть открыта и никакого влияния на производительность работы пользователя данная операция не окажет.

1. Перейдите на страницу Redo Log Groups (Server – Storage).
2. Выберите группу и щелкните на кнопке Edit, или же щелкните на ссылке, представляющей номер группы.
3. В разделе Redo Log Members щелкните на кнопке Add. Появится страница Add Redo Log Member.
4. Введите имя файла и имя каталога. Щелкните на кнопке Continue.

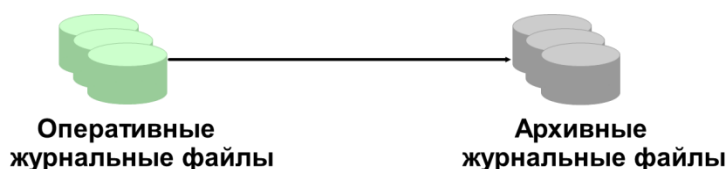
Примечание: рекомендуется хранить члены группы на разных дисках, чтобы защититься от потери всей группы в случае отказа одного диска.

5. Повторите шаги для каждой существующей группы.

Сразу после добавления члена группы его статус INVALID. Это ожидаемое состояние, поскольку в новый файл группы еще не было записи. После переключения журнала, когда группа станет текущей, статус нового файла изменится на CURRENT.

Настройте мультиплексирование оперативных журнальных файлов в каталог RedoLog второго диска виртуальной машины (созданного в предыдущем упражнении)

Архивные журнальные файлы



Экземпляр рассматривает группы оперативных журналов как циклический буфер, предназначенный для хранения транзакционной информации, и поэтому после использования одной группы переходит к следующей. После заполнения всех групп экземпляр начинает переписывать информацию в первой журнальной группе.

Для обеспечения максимально возможной восстанавливаемости необходимо сконфигурировать базу данных так, чтобы перед переписыванием оперативной журнальной группы обязательно делалась ее копия. Такие копии называются *архивными журналами* (*archived logs*). Для их создания следует:

1. Задать формат имени архивных журнальных файлов.
2. Указать одно или несколько мест хранения архивных журналов.
3. Перевести базу данных в режим ARCHIVELOG.

Примечание. Место хранения архивных журналов должно существовать до перевода базы данных в режим ARCHIVELOG. При задании каталога в качестве места хранения следует указывать конечную косую черту в имени каталога.

Имена и места расположения архивных журнальных файлов

Media Recovery

The database is currently in NOARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time are possible, but you must provide space for archived redo log files. If you change the database to ARCHIVELOG mode, you should perform a backup immediately. In NOARCHIVELOG mode, only cold backups are possible and data may be lost in the event of database corruption.

☐ ARCHIVELOG Mode*

Log Archive Filename Format*

Number	Archived Redo Log Destination	Status	Type
1	<input type="text" value="USE_DB_RECOVERY_FILE_DEST"/>	VALID	Local

[Add Another Row](#)

☒ TIP It is recommended that archived redo log files be written to multiple locations spread across the different disks.

☒ TIP You can specify up to 10 archived redo log destinations.

☐ Enable Minimal Supplemental Logging

Minimal supplemental logging logs the minimal amount of information needed for LogMiner (and any product building on LogMiner technology) to identify, group, and merge the redo operations associated with DML changes.

Щелкните на ссылке Recovery Settings на странице Availability.

Чтобы не переписывался старый архивный журнал, каждому архивному файлу должно присваиваться уникальное имя. Выше показано, как задается формат имени. База данных Oracle предоставляет шаблоны, помогающие создавать архивные файлы с уникальными именами.

- %s: включает в имя файла *номер журнала (log sequence number)*;
- %t: включает в имя файла *номер потока (thread number)*;
- %r: *идентификатор сброса журналов (Resetlogs ID)*; гарантирует уникальность имени архивных журнальных файлов после выполнения специального восстановления, вызывающего сброс номера журнала;
- %d: включает в имя файла *идентификатор БД (database ID)*.

Формат *должен* содержать шаблоны %s, %t и %r. Если одно место хранения используется для размещения архивных журналов нескольких баз данных, следует использовать шаблон %d.

Максимально архивные журналы могут записываться в десять различных мест расположения. Эти места расположения могут быть локальными (каталог) или удаленными (псевдоним Oracle Net для резервной базы данных). Имя локального местоположения должно заканчиваться косой чертой (/) или (в Windows) обратной косой чертой (\).

По умолчанию архивный журнал создается в месторасположение под номером 10, определяемом параметром инициализации DB_RECOVERY_FILE_DEST. Этот параметр задает *флэш-область восстановления (flash recovery area)*. Месторасположение *флэш-области* выводится в нижней части страницы Configure Recovery Settings в поле Flash Recovery Area Location. Чтобы это месторасположение не использовалось для хранения архивных журналов, просто удалите USE_DB_RECOVERY_FILE_DEST.

Операции по изменению параметров восстановления необходимо выполнять, подсоединившись как SYSDBA или SYSOPER.

Режим ARCHIVELOG

Чтобы перевести базу данных в режим ARCHIVELOG:

1. Отметьте поле ARCHIVELOG Mode .
2. Щелкните на кнопке Apply. База данных может быть переведена в режим ARCHIVELOG только, когда она находится в состоянии MOUNT.
3. Щелкните на кнопке Yes, когда будет выдан запрос на перезапуск базы данных.
4. Выполните резервирование базы данных.

*Базы данных в режиме ARCHIVELOG могут использовать все опции резервирования и восстановления.

Перевод базы данных в режим ARCHIVELOG запрещает перезапись оперативных журналов до их архивирования.

Следующая команда SQL переводит базу данных в режим ARCHIVELOG:
SQL> ALTER DATABASE ARCHIVELOG;

Эта команда может быть выполнена, когда база данных находится в состоянии MOUNT. Поэтому до нее надо перезапустить экземпляр, чтобы завершить последний шаг перевода базы данных в архивный режим. При перезапуске запрашиваются *данные для аутентификации (credentials)* в операционной системе и базе данных. Для базы данных *должен быть* указан пользователь с привилегиями SYSDBA.

После перезапуска экземпляра начнут действовать сделанные вами изменения, касающиеся архивных процессов, формата имен журналов и мест расположения журнальных файлов.

В режиме NOARCHIVELOG (по умолчанию) восстановление возможно только на момент получения последнего резерва. Все транзакции после этого момента будут потеряны.

В режиме ARCHIVELOG восстановление возможно до момента времени последней фиксации. Большинство промышленных баз данных работает в режиме ARCHIVELOG.

Примечание: выполните резервирование базы данных после перевода в режим ARCHIVELOG, поскольку в этом режиме база данных может быть восстановлена с использованием последних резервных объектов, полученных только в этом режиме.

В Enterprise Manager проверьте, что включен режим ARCHIVELOG (Recovery Settings на странице Availability), настройте формат имени архивных файлов: Arhiv%S_%R.%T%D.

Время на выполнение лабораторной работы – 2 часа.