

## DBA 101 – Секреты упреждающего мониторинга баз данных

(DBA 101 – Proactive Database Monitoring Tricks, by Robert DuMoulin, Data General Company)

Источник: доклад на конференции Oracle Open World 2000, no.726

(<http://www.oracle.com/oow/upload/726.pdf>)

В течение 11 лет администрирования баз данных Oracle я видел почти все, что может случиться при работе сервера Oracle. Несмотря на то, что *большинство* аппаратных и *некоторые* программные сбои не поддаются предсказанию даже самым предусмотрительным DBA, DBA – единственный, кто оказывается на месте до появления эксперта в конкретной сфере деятельности, связанной со сбоем. Помимо того, что DBA становится экспертом де-факто в RAID-технологиях, методах резервирования и восстановления, настройке операционных систем, разработчики его больше знают как создателя “**тоталитарной политики**”, которая “**снижает их производительность и препятствует прогрессу**”. В действительности же, если опытного DBA назовут “берегущим задницу”, то он должен считать это наивысшим комплиментом. Работа DBA была бы намного более легкой, если бы можно было устранить один фактор их повседневной жизни: **пользователей**. В реальной жизни существуют (и вы также можете это обнаружить) два простых фактора, которые делают вашу жизнь гораздо более приятной:

обучение пользователей и разработчиков;

выполнение упреждающего мониторинга и администрирования баз данных.

Обучение пользователей, требующее терпения и планирования, в данной статье не рассматривается. Мы – администраторы базы данных, которые, вероятно, были достаточно удачливы, чтобы быстро изучить, как действительно управлять средой Oracle, – у нас нет язвы, хорошо спим ночью, имеем все свои собственные волосы и перенесли меньше двух сердечных приступов. В этой статье раскрыты некоторые из полезных для этого секретов.

Я обнаружил, что в среднем примерно 70% рабочего времени DBA тратит на сбор информации, решение проблем управления пространством и попытки повысить производительность системы. К сожалению, так много времени расходуется на оперативные вопросы, что остается слишком мало времени на долгосрочное планирование, встречи с конечными пользователями и разработчиками, ответственными за поддержку. Без знаний DBA в области архитектуры и настройки систем разработчики обычно создают плохие схемы и несовершенный SQL-код, которые в высшей степени негативно влияют на производительность системы. Плохая производительность приводит к непредвиденным ситуациям, требующим от DBA немедленного вмешательства. Такие непредвиденные ситуации оставляют все меньше времени, чтобы заниматься с разработчиками, которые начинают новый цикл разработки. Функциональные возможности Enterprise Manager были расширены для облегчения администрирования среды Oracle. К сожалению, обратная сторона этого – ухудшилась способность среднего администратора Oracle быстро войти в курс дела и устранить ошибки в неисправном экземпляре. Менее опытные DBA стали надеяться на возможности Enterprise Manager, а не на понимание того, что случилось внутри системы. Я обнаружил, что более опытные DBA ценят возможности Enterprise Manager, но по-прежнему полагаются на свой собственный набор утилит.

Как консультанта администраторов баз данных Oracle, меня регулярно вызывают, чтобы я взял на себя обязанности по администрированию базы данных, о которой я мало чего знаю и не подготовлен. Часто случается так, что отсутствует “разбор полетов” с присутствующим (или недавно ушедшим) DBA, и я должен быть готов к администрированию промышленной среды в ту же секунду, когда я вхожу в дверь. В других случаях меня приглашают для “санитарной проверки” текущего состояния баз данных Oracle и эксплуатируемых процедур. В любом случае, первым делом нужно быстро оценить текущее состояние и будущую стабильность каждого экземпляра Oracle. Поскольку среды отличаются друг от друга, для быстрого получения информации о конфигурации, использовании пространства, а также диагностической информации, я полагаюсь на несколько командных файлов SQL. Из двух основных командных файлов, на которые я чаще всего надеюсь, один из них, командный файл проверки использования пространства и выдачи диагностики, приведен в данной статье. Главное назначение этого командного файла – помочь определить проблемы, потенциально опасные для работы системы, сохранить информацию об использовании пространства и выдать рекомендации по настройке (создается отчет, в котором классифицируются различные уровни серьезности индикаторов ключевых проблем). Командный файл разработан так, чтобы архивировать статистики управления пространством и производительности первого и пятнадцатого числа каждого месяца. Эти статистики позднее могут запрошены, чтобы показать тренды использования пространства и тренды производительности. Я коротко опишу каждый тест, а завершу рекомендациями – какие дополнительные автоматизированные процедуры нужно создать.

## Подготовка

Командному файлу для мониторинга, приведенному в данной статье позже, для хранения статистик использования пространства и производительности требуется одна таблица. Перед выполнением командного файла для мониторинга необходимо создать таблицу `monitor_stats` и ряд представлений. Пользователь, выполняющий мониторинг, для доступа к представлениям DBA должен иметь соответствующие привилегии `SELECT`.

```
/* КОМАНДНЫЙ ФАЙЛ - monitor_setup.sql */
/* АВТОР - Rob DuMoulin (произносится Dew-Moo-Len) */
/* rdumouli@mindspring.com) */
/* НАЗНАЧЕНИЕ - Создает статистическую таблицу и представления для */
/* командного файла monitor.sql. */
/* ОГРАНИЧЕНИЕ - Свободное использование со следующими исключениями. */
/* Если вы найдете лучшие способы реализации каких-либо тестов */
/* или даже дополните командный файл новыми тестами, */
/* пожалуйста, вышлите мне по e-mail'у новые операторы SQL */
/* с блоком комментариев, описывающим высланное. */
/* Если вы продадите этот командный файл, вы */
/* должны выслать половину вырученной суммы ИЛИ 1963 */
/* Corvette (обязательно с разделенным окном, желательно красного */
/* цвета) - в зависимости от того, что стоит БОЛЬШЕ. */
/* ОГРАНИЧЕНИЕ - Ни я, ни мой наниматель не несут какой-либо */
/* ответственности или обязанностей в отношении */
/* использования, ошибочного использования, неиспользования или */
/* злоупотреблений во время использования этого командного файла. */
/* Удаление этого блока комментариев может привести к */
/* порче блоков базы данных. */
SPOOL monitor_setup.log /* Протоколировать текущее выполнение */
DROP TABLE monitor_stats; /* Очистить предыдущую попытку */
CREATE TABLE monitor_stats ( /* Создать таблицу для работы и архива */
SID_name CHAR(10) NOT NULL, /* SID */
```

```

rundate DATE NOT NULL, /* Время запуска */
tablespace_name VARCHAR2(30), /* Табличное пространство (ТП) */
free_count NUMBER, /* Счетчик фрагментов свободного пространства */
free_space_K NUMBER, /* Объем доступного свободного пространства */
free_percent NUMBER, /* Процентная доля свободного пространства в ТП */
largest_free_K NUMBER, /* Наибольший непрерывный свободный фрагмент ТП */
total_size NUMBER, /* Общий размер табличного пространства */
object_count NUMBER, /* Счетчик объектов в ТП */
largest_next_k NUMBER, /* Наибольший определенный параметр NEXT */
temp_flag CHAR DEFAULT 'T' /* Тип записи */
CHECK (temp_flag IN ('T','A'))
TABLESPACE USERS /* Табличное пространство, содержащее таблицу */
STORAGE (INITIAL 100K NEXT 100K PCTINCREASE 0); /* Параметры хранения */
CREATE OR REPLACE VIEW ts_free AS /* Представление, упрощающее запросы */
SELECT a.tablespace_name,
COUNT(a.tablespace_name) free_count,
SUM(a.bytes)/1024 free_space_K,
MAX(a.bytes)/1024 largest_free_K
FROM dba_free_space a
GROUP BY a.tablespace_name;
CREATE OR REPLACE VIEW ts_size AS /* Представление, упрощающее запросы */
SELECT a.tablespace_name,
SUM(a.bytes)/1024 total_size_k
FROM sys.dba_data_files a
GROUP BY a.tablespace_name;
CREATE OR REPLACE VIEW ts_report AS /* Представление, упрощающее запросы */
SELECT SID_NAME "SID", TO_CHAR(rundate,'MM-DD-YYYY') rundate,
tablespace_name, free_count "SEGS_FREE",
free_space_k "K FREE",
free_percent "PCT_FREE", largest_free_k "LGST_FREE_K",
total_size "TS_SIZE_K", object_count "OBJECTS",
largest_next_k "MAX_NEXT_K"
FROM monitor_stats
WHERE temp_flag = 'A' AND
Tablespace_name != 'Performance Stats ';
CREATE OR REPLACE VIEW stat_report AS /* Представление, упрощающее запросы */
SELECT SID_NAME "SID", TO_CHAR(rundate,'MM-DD-YYYY') rundate,
free_count "HIT_RATIO", free_space_k "DISK_SORT_RATIO",
free_percent "LIB_CACHE_HIT_RATIO",
largest_free_k "REDO_ALLOC_RETRY_COUNT"
FROM monitor_stats
WHERE temp_flag = 'A' AND
Tablespace_name = 'Performance Stats ';
CREATE OR REPLACE VIEW stat_hit_ratio AS
SELECT ROUND(1-(phy.value/(log.value+con.value)),3) BUFFER_HIT_RATIO
FROM v$sysstat phy, v$sysstat log, v$sysstat con
WHERE phy.name = 'physical reads' AND log.name='db block gets' AND
con.name = 'consistent gets';
CREATE OR REPLACE VIEW stat_redo_alloc_retries AS
SELECT value retries FROM v$sysstat
WHERE name = 'redo buffer allocation retries';
SPOOL OFF

```

## Сбор статистик

Командный файл для мониторинга, показанный ниже, должен запускаться ежедневно в одно и то же время, тем не менее, командный файл разработан так, чтобы его можно было приспособить для многократного запуска в течение дня. Если база данных является критически важной (mission-

critical), я предлагаю создать два варианта этого командного файла: один с проверкой производительности, а другой без ее проверки. Вариант с проверкой производительности должен запускаться ежедневно после пикового периода работы (перед остановкой экземпляра), а вариант без статистик производительности может запускаться так часто, как сочтет DBA.

Командный файл выводит три класса сообщений:

сообщения класса *'NOTICE'* (извещение) содержат общую информацию, которая может представлять интерес или не представлять (в зависимости от политики безопасности и типа приложения). Сообщения этого класса не являются критичными для работы системы;

сообщения класса *'WARNING'* (предупреждение) показывают потенциальные проблемы, которыми следует заняться, пока они не стали причиной более серьезных ситуаций. Сообщения этого класса не являются критичными, но вероятнее всего станут сообщениями класса *'ERROR'*, если ими быстро не заняться;

сообщения класса *'ERROR'* (ошибка) указывают на серьезную ситуацию, которая, если не будет устранена, наиболее вероятно приведет к сбою работающей системы или потребует дорогостоящих работ по обслуживанию базы данных. На эти сообщения нужно немедленно реагировать.

При первом запуске данного командного файла он может выдать много сообщений классов Warning и Notice. Если вы найдете время привести в порядок старые представления, исправить установки параметров пользователей, параметры хранения и другое “хозяйство”, эти сообщения перестанут появляться в отчете. Действующий командный файл для мониторинга показан ниже. Его первое действие – собрать системные статистики и определить форматы вывода.

```
/* КОМАНДНЫЙ ФАЙЛ – monitor.sql */
/* АВТОР – Rob DuMoulin (произносится Dew-Moo-Len) */
/* rdumouli@mindspring.com) */
/* НАЗНАЧЕНИЕ – Выдает диагностику и собирает статистики роста */
/* размеров баз данных Oracle. */
/* ОГРАНИЧЕНИЕ – Свободное использование со следующими исключениями. */
/* Если вы найдете лучшие способы реализации каких-либо тестов */
/* или даже дополните командный файл новыми тестами, */
/* пожалуйста, вышлите мне по e-mail'у новые операторы SQL */
/* с блоком комментариев, описывающим высланное. */
/* Если вы продадите этот командный файл, вы */
/* должны выслать половину вырученной суммы ИЛИ 1963 */
/* Corvette (обязательно с разделенным окном, желательно красного */
/* цвета) – в зависимости от того, что стоит БОЛЬШЕ. */
/* ОГРАНИЧЕНИЕ – Ни я, ни мой наниматель не несут какой-либо */
/* ответственности или обязанностей в отношении */
/* использования, ошибочного использования, неиспользования или */
/* злоупотреблений во время использования этого командного файла. */
/* Удаление этого блока комментариев может привести к */
/* порче блоков базы данных. */
SET ECHO OFF FEED OFF
/* Удалить все временные (Temp) или */
/* архивные (Archive) записи, добавленные сегодня */
DELETE FROM monitor_stats
WHERE temp_flag = 'T' OR
(rundate = TRUNC(SYSDATE) AND temp_flag = 'A');
```

```

/* Добавить записи Temp или Archive о сегодняшнем использовании пространства */
INSERT INTO monitor_stats
(SELECT d.name, TRUNC(SYSDATE) rundate,
b.tablespace_name,
a.free_count, a.free_space_K,
a.free_space_k / b.total_size_k,
a.largest_free_K,
b.total_size_K,
COUNT(c.segment_name) object_count,
MAX(c.next_extent)/1024 largest_next_K,
DECODE(TO_CHAR(SYSDATE, 'DD'), 1, 'A', 15, 'A', 'T')
FROM ts_free a, ts_size b, dba_segments c, v$database d
WHERE b.tablespace_name = a.tablespace_name (+) AND
b.tablespace_name = c.tablespace_name (+)
GROUP BY d.name, b.tablespace_name, a.free_count,
a.free_space_K, a.largest_free_K, b.total_size_K);
COMMIT;
/* Форматы для отчетов */
COLUMN tablespace_name FORMAT A15 HEADING TABLESPACE
COLUMN segment_type FORMAT A6 HEADING TYPE
COLUMN segment_name FORMAT A15 HEADING OBJECT
COLUMN object_name FORMAT A15 HEADING OBJECT
COLUMN owner FORMAT A15
COLUMN free_percent FORMAT .999
COLUMN block_gets NEW_VALUE b_gets NOPRINT
COLUMN gets NEW_VALUE c_gets NOPRINT
COLUMN reads NEW_VALUE p_reads NOPRINT
COLUMN ratio FORMAT 999.90
SET FEED OFF TERM OFF VERIFY OFF PAGESIZE 1000 HEAD OFF ECHO OFF LINESIZE 90
/* Собрать статистики */
SELECT value block_gets FROM v$sysstat WHERE name = ('db block gets');
SELECT value gets FROM v$sysstat WHERE name = ('consistent gets');
SELECT value reads FROM v$sysstat WHERE name = ('physical reads');

```

Определить предупреждения и записать их в spool-файл. Этот файл нужно отправить DBA по электронной почте.

```

/* Открыть файл предупреждений и записать в него заголовок */
SPOOL warning.out
SELECT 'Предупреждения и извещения на '||SYSDATE||
' для экземпляра '||name
FROM v$database;
/* Проверка коэффициентов попаданий в кеш буферов */
COLUMN block_gets NEW_VALUE b_gets NOPRINT
COLUMN gets NEW_VALUE c_gets NOPRINT
COLUMN reads NEW_VALUE p_reads NOPRINT

```

Buffer Cache Hit Ratio (коэффициент попаданий в кеш буферов) показывает, как часто требуемая строка или узел индекса находятся в буферах базы данных, а не читаются с диска. Мой критерий выдачи сообщений на уровне предупреждений – меньше 90 %. Постоянные сообщения о низком коэффициенте попаданий показывают, что параметр db\_block\_buffers следует увеличить (в зависимости от объема доступной памяти).

```

SELECT 1-(phy.value/(log.value+con.value)) "Buffer Cache Hit %"
FROM v$sysstat phy, v$sysstat log, v$sysstat con
WHERE phy.name = 'physical reads' AND log.name='db block gets' AND
con.name = 'consistent gets';

```

```

SELECT 'Warning: Buffer Cache Hit Ratio = '||buffer_hit_ratio*100||'%'
FROM stat_hit_ratio
WHERE buffer_hit_ratio <= .9;

```

Журнальный буфер кеширует информацию для оперативных журналов. Redo Latches (защелки журнального буфера) – механизм блокирования памяти, обеспечивающий целостность журнального буфера. Может возникнуть конкуренция, если для начала операций ввода-вывода буфера нужно какое-то время ждать. Конкуренция отслеживается в SGA и определяется следующим тестом: если значение столбца WAIT % (количество защелок в %, полученных в режиме ожидания) или столбца IMM. % (количество защелок в %, полученных в режиме без ожидания) превышает 1 процент (%), то может иметь место конкуренция за выделение пространства в журнальном буфере. Постоянное возникновение такой ситуации может быть уменьшено установкой в INIT.ORA меньшего значения параметра LOG\_SMALL\_ENTRY\_MAX\_SIZE, что приведет к повышению интенсивности работы с защелками копирования журнального буфера и, следовательно, уменьшению конкуренции.

```

/* Проверка конкуренции за защелки журнального буфера */
COL wpct HEADING 'Wait|%' FORMAT 990.99 JUS CEN
COL ipct HEADING 'Imm.|%' FORMAT 990.99 JUS CEN
SELECT 'Warning: '||name||' существует конкуренция. % неудач в режиме с ожиданием = '||
SUM(misses)/(SUM(gets)+1)*100||' и % неудач в режиме без ожидания = '||
SUM(immediate_misses)/(SUM(immediate_gets)+1)*100
FROM v$latch
WHERE name LIKE 'redo%'
HAVING SUM(misses)/(SUM(gets)+1) > .01 OR
SUM(immediate_misses)/(SUM(immediate_gets)+1) > .01
GROUP BY name;

```

Если размер области сортировки в SGA (параметр sort\_area\_size) недостаточен для выполнения операций сортировки, то для сортировок будут использоваться диски. Ввод-вывод на диски существенно медленнее, чем работа с оперативной памятью, поэтому сортировки будут растянуты во времени. Данный тест выдает предупреждение, если более 5 % сортировок выполняется на диске, – в зависимости от приложения и доступной памяти может потребоваться увеличение параметра sort\_area\_size.

```

/* Проверка дисковых сортировок */
COLUMN mem NEW_VALUE memory NOPRINT
COLUMN disk NEW_VALUE disks NOPRINT
SELECT value "mem" FROM v$sysstat WHERE name = 'sorts (memory)';
SELECT value "disk" FROM v$sysstat WHERE name = 'sorts (disk)';
SELECT 'Warning: '||&disks/(&disks+&memory)*100||
'% сортировок выполняется на диске. Увеличьте SORT_AREA_SIZE.'
FROM dual
WHERE &disks/(&disks+&memory) > .05;

```

Перезагрузки объектов библиотечного кеша (Library Cache) возникают тогда, когда требуется повторный разбор операторов SQL, так как их вход в библиотечный кеш был заменен другим оператором SQL. В идеале, перезагрузок библиотечного кеша в системе не должно быть. В реальности, все что ниже 5 % – приемлемо (вы можете снизить этот порог до 2 %, если критична производительность выполнения нерегламентированных запросов). Для увеличения размеров библиотечного кеша увеличивайте в INIT.ORA значение параметра SHARED\_POOL\_SIZE. Вы

можете также попробовать увеличить количество доступных открытых курсоров (параметр OPEN\_CURSORS), что также может сократить количество повторных разборов операторов SQL.

```
/* Проверка коэффициента попаданий в библиотечный кеш */
SELECT 'Warning: коэффициент попаданий в библиотечный кеш на уровне системы = '||
(SUM(b.pins - b.reloads))/SUM(b.pins)*100||
'%. Увеличьте SHARED_POOL_SIZE'
FROM v$librarycache
HAVING (SUM(b.pins - b.reloads))/SUM(b.pins)< .90;
/* Проверка чрезмерных перезагрузок в библиотечном кеше */
SELECT 'Warning: коэффициент перезагрузок библиотечного кеша на уровне системы = '||
SUM(reloads)/SUM(pins)||'%. Увеличьте SHARED_POOL_SIZE'
FROM v$librarycache
HAVING SUM(reloads)/SUM(pins) > .05;
```

Кеш словаря данных (Dictionary Cache) содержит информацию об объектах базы данных. Ввод-вывод для этого кеша может быть весьма значительным, поэтому желательно поддерживать размер кеша достаточно большим, чтобы в нем размещалось много данных словаря. Уменьшайте перезагрузки, увеличивая в INIT.ORA значение параметра SHARED\_POOL\_SIZE.

/

```
* Проверка перезагрузок кеша словаря данных (getmisses) */
SELECT 'Warning: перезагрузки библиотечного кеша на уровне системы = '||
(SUM(gets-getmisses-usage-fixed))/SUM(gets)*100||
'%. Increase SHARED_POOL_SIZE'
FROM v$rowcache
HAVING (SUM(gets-getmisses-usage-fixed))/SUM(gets) > .2;
/* Архивирование статистик производительности в табличном пространстве 'Performance
Stats.' */
INSERT INTO monitor_stats
SELECT d.name, TRUNC(SYSDATE) rundate,
'Статистики производительности',
a.buffer_hit_ratio*100, /* Коэффициент попаданий в буфера */
&disks/(&disks+&memory)*100, /* Дисковые сортировки */
(SUM(b.pins - b.reloads))/
SUM(b.pins)*100, /* Коэффициент попаданий в библиотечный кеш */
c.retries, /* Количество ожиданий пространства в журнальном буфере */
NULL, NULL, NULL,
DECODE(TO_CHAR(SYSDATE,'DD'),1,'A',15,'A','T')
FROM stat_hit_ratio a, v$librarycache b,
stat_redo_alloc_retries c, v$database d
GROUP BY d.name, sysdate, buffer_hit_ratio, retries;
COMMIT;
```

Фрагментация табличного пространства возникает при удалении объектов. После удаления объекта, размещенного в нескольких экстентах, остается столько свободных фрагментов, сколько во время удаления у объекта было экстенгов. Если даже два пустых сегмента расположены рядом, Oracle при выделении следующего экстенга для расширяющегося объекта не распознает два свободных фрагмента как одно непрерывное пространство (во время создания начальных экстенгов объектов Oracle находит время для определения смежных свободных фрагментов). Установка ненулевого значения параметра табличного пространства DEFAULT STORAGE PERCENT INCREASE указывает, что смежные свободные фрагменты должны объединяться автоматически. Для временного табличного пространства и табличного пространства для сегментов отката всегда выделяются экстенги одинакового размера, поэтому для них значение

DEFAULT STORAGE PERCENT INCREASE всегда должно быть равно 0. В Oracle для ручного объединения также предусмотрен оператор ALTER TABLESPACE ... COALESCE. Хотя объединение свободных фрагментов снижает объем фрагментации свободного пространства, в большинстве промышленных баз данных фрагментация табличных пространств неизбежна. Чрезмерная фрагментация может привести к неэффективному использованию пространства и невозможности расширения объектов. Данный тест определяет два уровня фрагментации и выдает сообщения класса 'Notice' или 'Warning.' Кроме того, определяется установка значения PCTINCREASE по умолчанию, что также способствует фрагментации.

```
/* Проверка фрагментации табличных пространств */
SELECT 'Warning: фрагментированное табличное пространство '
||tablespace_name||' имеет '||free_count||' свободных сегментов.'
FROM monitor_stats
WHERE free_count >= 90 AND
TRUNC(rundate) = TRUNC(SYSDATE) AND
tablespace_name not like 'TEMP%' AND
tablespace_name not like 'TMP%' AND
tablespace_name != 'Performance Stats ' AND
tablespace_name not in
(SELECT tablespace_name FROM sys.dba_rollback_segs)
ORDER BY free_count desc;
SELECT 'Notice: фрагментированное табличное пространство '
||tablespace_name||' имеет '||free_count||' свободных сегментов.'
FROM monitor_stats
WHERE free_count >= 40 AND free_count < 90 AND
TRUNC(rundate) = TRUNC(SYSDATE) AND
tablespace_name not like 'TEMP%' AND
tablespace_name != 'Performance Stats ' AND
tablespace_name not like 'TMP%' AND
tablespace_name not in
(SELECT tablespace_name FROM sys.dba_rollback_segs)
ORDER BY free_count desc;
/* Проверка причин фрагментации */
SELECT 'Warning: установлено значение PCT_INCREASE по умолчанию. Табличное
пространство '||
tablespace_name||'. PCT_INCREASE = '||pct_increase||'.'
FROM sys.dba_tablespaces
WHERE pct_increase != 1 AND tablespace_name NOT LIKE 'TEMP%' AND
tablespace_name not in
(SELECT tablespace_name from dba_rollback_segs)
ORDER BY tablespace_name;
```

В настоящее время объекты могут создаваться без ограничения максимального количества экстенгов, но я советую DBA ограничивать максимальное количество экстенгов и часто использовать тест, подобный приведенному ниже, который находит расширяющиеся объекты. Я также не поддерживаю установку для объектов значения PCTINCREASE, отличного от 0, исключая случаи, когда пользователи не имеют представления об ожидаемом расширении. Но этого никогда не случается – любой DBA знает, что нет невежественных пользователей!

```
/* Проверка максимального количества экстенгов */
SELECT 'Warning: '||segment_type||' '||owner||'. '||segment_name||
' имеет '||extents||' экстенгов из '||max_extents||'.'
FROM sys.dba_segments
WHERE (extents >= .75 * max_extents) AND
(extents <= .95 * max_extents) AND
segment_type in ('TABLE','INDEX') AND
```



```

tablespace_name not in
(SELECT tablespace_name FROM sys.dba_rollback_segs)
ORDER BY extents;
SELECT 'Notice: '||segment_type||' '||owner||'.'||segment_name||
' имеет '||extents||' экстенгов из '||max_extents||'.'
FROM sys.dba_segments
WHERE (extents >= .5 * max_extents) AND
(extents < .75 * max_extents) AND
segment_type in ('TABLE','INDEX') AND
tablespace_name not in
(SELECT tablespace_name FROM sys.dba_rollback_segs)
ORDER BY extents;
/* Проверка значения PCT_INCREASE */
SELECT 'Warning: '||segment_type||' '||owner||'.'||segment_name||
' имеет значение PCT_INCREASE равное '||pct_increase
FROM sys.dba_segments
WHERE pct_increase != (0) AND segment_type in ('TABLE', 'INDEX')
AND owner not in ('SYSTEM', 'SYS')
ORDER BY segment_type, pct_increase, owner, segment_name desc;

```

Следующий тест – проверка свободного пространства в табличных пространствах. Предупреждение выдается, если в табличном пространстве занято от 85 % до 95 % всего объема.

```

/* Проверка ограниченного свободного пространства в табличных пространствах */
SELECT 'Warning: в табличном пространстве '||tablespace_name||
' свободное пространство уменьшилось до '||free_percent*100||' %.'
FROM monitor_stats
WHERE free_percent <= .15 AND free_percent > .05 AND
tablespace_name != 'Performance Stats ' AND
TRUNC(rundate) = TRUNC(SYSDATE) AND tablespace_name not in
(SELECT tablespace_name FROM sys.dba_rollback_segs)
ORDER BY free_percent;

```

Объекты (таблицы, представления, хранимые процедуры и индексы) помечаются как ‘invalid’ (недействительный), если модифицируются определенные зависимости. При следующем доступе к таким объектам Oracle будет сначала повторно подтверждать их действительность. Если при этом возникает какая-либо проблема, процесс, начавший повторную компиляцию, выдает сообщение об ошибке возникновения исключительной ситуации. Обычно повторное подтверждение действительности объектов проходит без проблем и пользователи не замечают дополнительной нагрузки на систему. Данный тест часто идентифицирует недействительные объекты до их повторного использования, позволяя DBA решить проблемы до сбоя работающего процесса.

```

/* Проверка недействительных объектов */
SELECT 'Notice: '||object_type||' '||owner||'.'||object_name||
' - недействителен (INVALID).'
FROM sys.dba_objects
WHERE status = 'INVALID';
SPOOL OFF

```

В следующем разделе рассмотрены потенциально критические ситуации, которые легко могут быть обнаружены до возникновения проблем. Командный файл, приведенный ниже, выводит сообщения в файл “error.out.” Первый тест находит объекты (таблицы или индексы), которые имеют свыше 95 % экстенгов от максимального количества, определенного для них.

```

SPOOL error.out

```

```

SELECT 'Тестирование ошибок за '||SYSDATE||' для экземпляра '||name
FROM v$database;
/* Проверка максимального количества экстенгов */
SELECT segment_type||' '||owner||'.'||segment_name||' has reached '||
extents||' extents of '||max_extents||'.'
FROM sys.dba_segments
WHERE extents > .95 * max_extents AND
segment_type in ('TABLE','INDEX') AND
tablespace_name not in
(SELECT tablespace_name FROM sys.dba_rollback_segs)
ORDER BY extents;

```

Данный тест сравнивает объем доступного непрерывного свободного пространства каждого табличного пространства со значением параметра экстенга NEXT каждого резидентного объекта. Тест определяет объекты, для которых выделение дополнительных экстенгов завершится сбоем. Если возникает такая ошибка, попробуйте объединить смежные свободные фрагменты табличного пространства, а также проверьте, насколько реалистично значение параметра NEXT. Часто, если значение параметра PCTINCREASE не равно 0 или если объект не был создан с реалистичными параметрами хранения, DBA может изменить параметры хранения объекта и решить данную проблему.

```

/* Проверка объектов, которые не могут быть расширены */
SELECT 'Error: '||segment_type||' '||owner||'.'||segment_name||
' '||next_extent/1024||'K - выделение следующего экстенга невозможно. '||
max(a.bytes)/1024||'K free in '||a.tablespace_name||'.'
FROM sys.dba_free_space a, sys.dba_segments b
WHERE b.tablespace_name = a.tablespace_name AND
segment_type in ('TABLE','INDEX')
HAVING max(a.bytes) < next_extent
GROUP BY a.tablespace_name, owner, segment_name, segment_type,
next_extent;

```

Табличные пространства, у которых меньше 5 % свободного пространства, помечаются как проблемные, даже если в них отсутствуют объекты, выделение следующих экстенгов для которых может завершиться сбоем.

```

/* Проверка ограниченного свободного пространства в табличных пространствах */
SELECT ' Error: в табличном пространстве '||tablespace_name||
' свободное пространство уменьшилось до '||free_percent*100||' %.'
FROM monitor_stats
WHERE free_percent < .05 AND
tablespace_name != 'Performance Stats ' AND
TRUNC(rundate) = TRUNC(SYSDATE) AND
tablespace_name not in
(SELECT tablespace_name FROM sys.dba_rollback_segs)
ORDER BY free_percent;

```

Существует несколько установок параметров пользователей, которые могут стать проблемой для администратора базы данных или даже могут привести к серьезным ошибкам базы данных. Табличное пространство по умолчанию (default tablespace) и временное табличное пространство (temporary tablespace) никогда не следует устанавливать как табличное пространство SYSTEM. Как правило, все пользователи, исключая пользователей SYS и SYSTEM, должны иметь в табличном пространстве SYSTEM нулевую квоту. Если пользователи размещают в табличном пространстве SYSTEM свои объекты, эти объекты помечаются как ошибочные и должны быть удалены как

можно быстрее. Сохранение табличного пространство SYSTEM свободным от объектов пользователей и временных сегментов, сохранит системную область чистой, нефрагментированной и в безопасности.

```
/* Проверка недопустимых установок параметров пользователей */
SELECT 'Error: у пользователя '||username||
' табличное пространство по умолчанию: '||
default_tablespace||'.'
FROM sys.dba_users
WHERE (default_tablespace = 'SYSTEM' OR
default_tablespace like '%TEMP%' OR
default_tablespace like 'TMP%') AND
username not in ('SYS', 'SYSTEM');
SELECT 'Error: у пользователя '||username||
' временное табличное пространство - SYSTEM.'
FROM sys.dba_users
WHERE (temporary_tablespace = 'SYSTEM') AND
username not in ('SYS', 'SYSTEM');
/* Проверка наличия объектов пользователей в SYSTEM или TEMP */
SELECT 'Error: сегмент '||segment_type||' '||owner||'.'||segment_name||
' размещен в '||tablespace_name||'.'
FROM sys.dba_segments
WHERE owner not in ('SYS', 'SYSTEM') AND
(tablespace_name = 'SYSTEM' OR tablespace_name like '%TEMP%'
OR tablespace_name like 'TMP%')
AND segment_type in ('TABLE','INDEX')
ORDER BY owner, tablespace_name, segment_type, segment_name;
spool off
SET ECHO ON HEAD ON TERM ON FEED ON VERIFY ON
```

Рассмотренный командный файл идентифицирует большинство типовых проблем баз данных, с которыми ежедневно сталкивается DBA. Другие автоматизированные задачи, которые я рекомендую выполнять, включают сопровождение сигнального (alert) файла, резервирование управляющего файла и экспорт без данных (полный экспорт базы данных без строк). Сигнальные файлы следует проверять ежедневно, и о любых ошибках ORA 600 нужно автоматически сообщать DBA. Это легко сделать, создав автоматизированную задачу, которая просматривает сигнальный файл и по e-mail'у высылает все строки, содержащие конкретные текстовые сообщения (то есть 'ORA-600'). Каждый день сигнальный файл объединяется с архивным сигнальным файлом, а затем удаляется, чтобы освободить место для протоколирования системных событий следующего дня. Следует создать два задания, которые выдают файлы по дням недели (например, Mondayfull.exp), что позволит сохранять резервные копии управляющего файла за семь дней (оператор (ALTER DATABASE BACKUP CONTROLFILE), а также экспортные файлы (без данных). Экспорт без данных необходим для повторного создания структур индексов, таблиц и представлений в тех случаях, когда потребуется повторное создание одиночных объектов. Без экспорта строк, однако, данные восстановить нельзя.

Успешного вам администрирования.