

Oracle 9i Features*

(* The ones they DIDN'T tell you about)

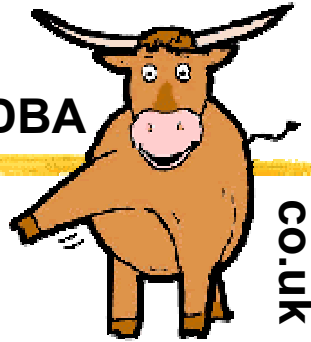
www.dmr.com.au / www.oracledba.co.uk

Presented by
CONNOR MCDONALD
Senior Consultant
DMR Consulting, Asia Pacific

Sep 2003



OracleDBA



Advertisements

- Performance / Configuration
- ~~Trouble making shooting~~
- <http://www.oracledba.co.uk>
- <http://www.dmr.com.au>
- <http://www.oaktable.net>



Agenda

- Strangely enough...the same as the abstract
- Questions...see the OakTable

A show of hands



Why New Features?

- **Usefulness**
 - Don't reinvent a slower and dumber wheel
- **Half the battle**
 - knowing that features are there
- **The other half**
 - test, test, test !!!!
 - examples are not proof
- **Why “under-hyped” and “under valued”?**
 - Stuff you can use today on *existing* databases

■ **A** **sumable space alloc.**

Test! Test! Test!

```
SQL> create table T ( x, y, z)
      2  as select 1,2,3 from dual;
```

```
SQL> create index T_IX on T (x,y);
```

```
SQL> alter table T add constraint C primary key (x,y);
```

```
SQL> alter index T_IX rebuild compress 1;
```

```
SQL> insert into t values (1,2,3);
```

ERROR at line 1:

ORA-08102: index key not found, obj# 69517, dba 29368266

- “Fixed” 9.2.0.3

Tools / Utilities

www.dmr.com.au / www.oracledba.co.uk



Lets start with SQLPlus

- **SQLPLUS -L**

- one attempt at user/pass only

- **Connection details available**

```
SQL> SET SQLPROMPT ' &_CONNECT_IDENTIFIER > '
mysid >
```

- **URL's for scripts**

```
SQL> @http://script.repository/scripts/sessions.sql
```

Export

- **'exp' can unload a tablespace**
 - TABLESPACES=(MY_DATA1, MY_DATA2)
 - *based on the TABLE*

SQL Loader

- **BINDSIZE default increased to 256k**
- **Expression enhancements**
 - EXPRESSION clause for column population
 - FILLER fields available (via BOUNDFILLER)
- **External table script generation**
 - EXTERNAL_TABLE=GENERATE_ONLY
- **Direct Load improvements**
 - Date cache
 - Default 1000
 - Expressions allowed
 - Kernel based
 - User PL/SQL



Explain Plan – the old

```
SELECT ID, PARENT_ID PID, LPAD(' ',2*LEVEL) ||
      OPERATION || ' ' ||
      DECODE(ID,0,' (OPTIMIZER='||OPTIMIZER||') ',NULL) ||
      OPTIONS || ' ' || OBJECT_NAME ||
      NVL2(COST,' (COST='||COST||') ',NULL) ||
      NVL2(CARDINALITY,' (ROWS='||CARDINALITY||') ',NULL) ||
      NVL2(PARTITION_START,' (PART:'||PARTITION_START||
          '-'||PARTITION_STOP||') ',NULL) ||
      NVL2(OTHER_TAG,' ('||OTHER_TAG||') ',NULL) PLAN
FROM PLAN_TABLE
CONNECT BY PRIOR ID = PARENT_ID AND
STATEMENT_ID = '&id' START WITH ID = 0
AND STATEMENT_ID = '&id'
ORDER BY ID;
```

Explain Plan – the new

■ Use DBMS_XPLAN

```
SQL> explain plan for
      2  select ename, sal
      3  from emp e , dept d
      4  where e.deptno=d.deptno
      5  and e.sal > 500;
```

```
SQL> select * from
      2  table(dbms_xplan.display) ;
```

Explain Plan – the new

Id	Operation	Name	Rows	Bytes	Cost
0	SELECT STATEMENT				
1	NESTED LOOPS				
* 2	TABLE ACCESS FULL	EMP			
* 3	INDEX UNIQUE SCAN	PK_DEPT			

Predicate Information (identified by operation id):

- 2 - filter("E"."SAL">500)
- 3 - access("E"."DEPTNO"="D"."DEPTNO")

Note: rule based optimization

Explain Plan – the new

- Detect the index columns used

```
SQL> explain plan for
  2  select * from T1
  3  where I_COL=10 and I_CHAR_COL=20
```

Index (I_COL,I_CHAR)
but datatype mismatch

Id	Operation	Name	Rows	Bytes
0	SELECT STATEMENT		1	12
1	TABLE ACCESS BY INDEX ROWID	T1	1	12
* 2	INDEX RANGE SCAN	T1X_1	1	

```
2 - access("T1"."I_COL"=10)
    filter(TO_NUMBER("T1"."I_CHAR_COL")=20)
```

Explain Plan – the new

- Automatically adjusts its output (format=TYPICAL)

```
SQL> explain plan for
      2  select *
      3  from PAR_TABLE
      4  where PCOL < 750
```

Id	Operation	Name	...	Pstart	Pstop
0	SELECT STATEMENT		...		
1	PARTITION RANGE ITERATOR		...	1	2
2	TABLE ACCESS FULL	PAR_TABLE	...	1	2

Explain Plan – the new

```
SQL> explain plan for
  2  select /*+ PARALLEL(p,4) */
  3  from PAR_TABLE p
  4  where PCOL < 750
```

```
----- ... -----
| Id | ... Pstart| Pstop | TQ  | IN-OUT| PQ Distrib |
----- ... -----
|  0 | ...      |      |      |      |      |
|  1 | ...      1 |      2 | 90,00 | PCWP |      |
|*  2 | ...      1 |      2 | 90,00 | P->S | QC (RANDOM) |
----- ... -----
```

Analytics

- **So many hidden opportunities...**
 - FIRST / LAST
 - PERCENTILE_CONT / PERCENTILE_DISC

Analytics – the past

- “Show me lowest salary for each department...”

```
SQL> select deptno, min(sal)
2   from emp
3   group by deptno;
```

- “...and I need to see employee number as well”

```
SQL> select deptno, empno, min(sal)
2   from emp
3   group by deptno;
```

ORA-00979: not a GROUP BY expression

Analytics – the new

- FIRST / LAST ≡ “include non-grouped columns”

```
SQL> select deptno, min(sal), min(empno)
  2      KEEP ( dense_rank FIRST order by sal) empno
  3 from emp
  4 group by deptno
  5 /
```

DEPTNO	MIN (SAL)	EMPNO
10	1300	7934
20	800	7369
30	950	7900

Emp 7934 is the person with the lowest salary in dept 10

Analytics – the past

■ “What is the median salary?”

```
select avg(sal) from
( select      x.sal
  from EMP x, EMP y
  group by x.sal
  having sum(decode(x.sal,y.sal,1,0)) >=
            abs(sum(sign(x.sal - y.sal))))
```

Cartesian product

AVG (SAL)

1550

“What the ?”

Analytics – the new

- “What is the median salary?”

```
SQL> select PERCENTILE_CONT(0.5)
2      within group (ORDER BY sal DESC) sal
3  from emp;
```

```
      SAL
-----
1550
```

Groupings

- More control over groupings

```
SQL> select deptno, job, mgr, sum(sal) from emp
      2  group by grouping sets ((deptno), (job,mgr), ()) ;
```

DEPTNO	JOB	MGR	SUM(SAL)	
	CLERK	7698	950	
	CLERK	7902	800	
	MANAGER	7839	8275	
	SALESMAN	7698	5600	(job,mgr)
	PRESIDENT		5000	
10			8750	
30			9400	(deptno)
			29025	()

WITH clause

- WITH clause for complex subqueries

```
SQL>  WITH dept_salaries AS (  
2      SELECT dname, SUM(sal) dept_sal  
3          FROM emp e, dept d  
4          WHERE e.deptno = d.deptno  
5          GROUP BY dname) ,  
6      avg_sal AS ( SELECT AVG(dept_sal) asal  
7                  FROM dept_salaries)  
8  
9      SELECT * FROM dept_salaries d, avg_sal a  
10     WHERE d.dept_sal > a.asal  
11     ORDER BY d.dname;
```

Its not always a subquery...

```

-----
|  0 | SELECT STATEMENT          |          |
|  2 |   TEMP TABLE TRANSFORMATION |          |
|  1 |     RECURSIVE EXECUTION    | SYS_LE_2_0 |
|  0 |       INSERT STATEMENT     |          |
|  1 |         LOAD AS SELECT     |          |
|  2 |           SORT GROUP BY    |          |
|*  3 |             HASH JOIN      |          |
|  4 |               TABLE ACCESS FULL | DEPT      |
|  5 |               TABLE ACCESS FULL | EMP      |
|  3 |         SORT ORDER BY     |          |
|*  4 |           VIEW            |          |
|  5 |             TABLE ACCESS FULL | SYS_TEMP_0FD9D6602_B7911 |
|  6 |             VIEW          |          |
|  7 |             SORT AGGREGATE |          |
|  8 |             VIEW          |          |
|  9 |             TABLE ACCESS FULL | SYS_TEMP_0FD9D6602_B7911 |
-----

```

Hierarchy

- Lots of improvements

```
SQL> select rpad(' ',level)||
2      sys_connect_by_path(ename,'-') full_tree,
3      empno, dname
4  from emp e, dept d
5  where d.deptno = e.deptno
6  start with mgr is null
7  connect by prior empno = mgr
8  order siblings by ename
```

Hierarchy

FULL_TREE	EMPNO	DNAME
-KING	7839	ACCOUNTING
-KING-BLAKE	7698	SALES
-KING-BLAKE-ALLEN	7499	SALES
-KING-BLAKE-JAMES	7900	SALES
-KING-BLAKE-MARTIN	7654	SALES
-KING-BLAKE-TURNER	7844	SALES
-KING-BLAKE-WARD	7521	SALES
-KING-CLARK	7782	ACCOUNTING
-KING-CLARK-MILLER	7934	ACCOUNTING

- Improved optimizer plans as well
 - BUFFER SORT
 - CONNECT PUMP

Locks – the past

```
SQL> select SAL from EMP
      2  where EMPNO = 7369
      3  for update of SAL;
(waiting... waiting... waiting...)
```

```
SQL> select SAL from EMP
      2  where EMPNO = 7369
      3  for update of SAL NOWAIT;
```

ERROR at line 1:

ORA-00054: resource busy and acquire with NOWAIT specified

Locks – the new

```
SQL> select SAL from EMP
      2  where EMPNO = 7369
      3  for update of SAL WAIT 3;
```

(1 ... 2 ... 3 ...)

ERROR at line 1:

ORA-30006: resource busy; acquire with WAIT timeout expired

■ Note

- It's a different error code
- Careful with “LOCK TABLE ... IN EXCLUSIVE MODE”

DDL

www.dmr.com.au / www.oracledba.co.uk



Grants – the past

```
SQL> conn / as sysdba
```

```
Connected.
```

```
SQL> grant select on SCOTT.EMP to PUBLIC;
```

```
ERROR at line 1:
```

```
ORA-01031: insufficient privileges
```

■ Ugly Workarounds

- create procedure scott.do_grant ...
- password hacking

SYS is SYS

```
SQL> conn / as sysdba
```

Connected.

```
SQL> grant select on SCOTT.EMP to PUBLIC;
```

Grant succeeded.

- “GRANT ANY OBJECT PRIVILEGE” privilege

Constraints – the past

```
SQL> create table MY_TAB ( UNIQ_COL number
2      constraint MY_TAB_UQ unique );
```

```
SQL> select index_name from user_indexes
3      where table_name = 'MY_TAB';
```

INDEX_NAME

MY_TAB_UQ

```
SQL> alter table MY_TAB disable constraint MY_TAB_UQ;
```

```
SQL> select index_name from user_indexes
2      where table_name = 'MY_TAB';
```

no rows selected

Constraints – the new

```
SQL> alter table MY_TAB disable constraint MY_TAB_UQ
      2  KEEP INDEX;
```

```
SQL> select index_name from user_indexes
      3  where table_name = 'MY_TAB';
```

INDEX_NAME

MY_TAB_UQ

- ‘drop’ or ‘disable’, unique or primary keys
- Hmmmmmmmmmm....

Constraints – the past

- (7.3+) index columns $\neq$ constraint columns

```
SQL> create index EMP_MGR_HIRE_SAL_IX
      2  on EMP (MGR, HIREDATE, SAL) ;
```

```
SQL> alter table EMP
      2  add constraint EMP_MGR_HIRE_UQ
      3  UNIQUE (MGR, HIREDATE) ;
```

- But how do I know ?

Constraints – the past

```
SQL> SELECT c.constraint_name,o.object_name
2      FROM sys.con$ s, user_constraints c,
3           sys.cdef$ s2, user_objects o
4  WHERE c.table_name = 'EMP'
5        AND c.constraint_name = 'EMP_MGR_HIRE_UQ'
6        AND c.constraint_name = s.name
7        AND s.con# = s2.con#
8        AND s2.enabled = o.object_id;
```

CONSTRAINT_NAME	OBJECT_NAME
EMP_MGR_HIRE_UQ	EMP_MGR_HIRE_SAL_IX

Constraints – the new

```
SQL> alter table EMP
2  add constraint EMP_MGR_HIRE_UQ
3  unique (MGR,HIREDATE)
4  using index
5      ( create index EMP_MGR_HIRE_SAL_IX
6        on EMP (MGR,HIREDATE,SAL) ) novalidate;
```

```
SQL> select CONSTRAINT_NAME, INDEX_NAME
2  from user_constraints
3  where TABLE_NAME = 'EMP';
```

CONSTRAINT_NAME	INDEX_NAME
-----	-----
EMP_MGR_HIRE_UQ	EMP_MGR_HIRE_SAL_IX

Constraints – the new

■ Note

- No relationship stored for foreign keys
- Use NOVALIDATE to ensure index creation succeeds

■ Other goodies

- alter table DEPT **rename constraint** SYS_C01234
to DEPT_CK1;

DDL Extraction – the past

- **exp rows=n file=ddl.dmp**
- **strings ddl.dmp > my_ddl.sql**
 - 1024 character limit
 - Additional editing required
- **Custom home-grown dictionary scripts**
- **imp indexfile=my_ddl.sql**
 - Hack out all of the REM's
- **imp show=y**
 - Line breaks wherever, whenever

DDL Extraction – the new

```
SQL> select dbms_metadata.get_ddl('TABLE','EMP','SCOTT')
2 from dual;
```

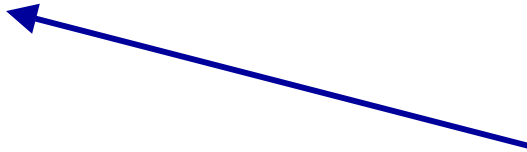
```
CREATE TABLE "SCOTT"."EMP"
(
  "EMPNO" NUMBER(4,0),
  "ENAME" VARCHAR2(10),
  ...
  CONSTRAINT "PK_EMP" PRIMARY KEY ("EMPNO")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
STORAGE(...) TABLESPACE "USERS" ENABLE,
  CONSTRAINT "FK_DEPTNO" FOREIGN KEY ("DEPTNO")
  REFERENCES "SCOTT"."DEPT" ("DEPTNO") ENABLE
) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING STORAGE(...) TABLESPACE "USERS"
```

DDL Extraction – the new

```
begin
```

```
  DBMS_METADATA.SET_TRANSFORM_PARAM(  
    DBMS_METADATA.SESSION_TRANSFORM,  
    'STORAGE',  
    false);
```

```
end;
```



```
SQLTERMINATOR  
SEGMENT_ATTRIBUTES  
TABLESPACE  
CONSTRAINTS  
REF_CONSTRAINTS  
CONSTRAINTS_AS_ALTER
```

```
SQL> select dbms_metadata.get_ddl('TABLE','EMP','SCOTT')  
2  from dual;
```

DDL Extraction – the new

```
SQL> select dbms_metadata.get_ddl('USER', 'SCOTT')
       2  from dual;
```

```
CREATE USER "SCOTT"
IDENTIFIED BY VALUES 'F894844C34402B67'
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP"
```

DDL Extraction – the new

ASSOCIATION

AUDIT

AUDIT_OBJ

CLUSTER

COMMENT

CONSTRAINT

CONTEXT

DB_LINK

DEFAULT_ROLE

DIMENSION

DIRECTORY

FUNCTION

INDEX

INDEXTYPE

JAVA_SOURCE

LIBRARY

MATERIALIZED_VIEW

MATERIALIZED_VIEW_LOG

OBJECT_GRANT

OPERATOR

OUTLINE

PACKAGE

PACKAGE_SPEC

PACKAGE_BODY

PROCEDURE

PROFILE

PROXY

REF_CONSTRAINT

ROLE

ROLE_GRANT

ROLLBACK_SEGMENT

SEQUENCE

SYNONYM

SYSTEM_GRANT

TABLE

TABLESPACE

TABLESPACE_QUOTA

TRIGGER

TRUSTED_DB_LINK

TYPE

TYPE_SPEC

TYPE_BODY

USER

VIEW

XMLSCHEMA



Online Operations

www.dmr.com.au / www.oracledba.co.uk



Index Online – the past

- **alter index MY_IX rebuild online;**
 - **No** secondary index on IOT
 - **No** IOT with overflow segment
 - **No** bitmap
 - **No** reverse
 - **No** function-based
 - **No** descending
 - **No** compress
 - **No** parallel
 - **No** compute statistics

+ **BUG:1475310**
(corruption/missing data)

Index Online – the past

```
SQL> analyze index MY_IX validate structure;
```

ERROR at line 1:

ORA-00054: resource busy and acquire with NOWAIT specified

(or exclusive lock)

```
SQL> alter table MY_IOT coalesce;
```

ERROR at line 1:

ORA-14004: missing PARTITION keyword

Index Online – the new

- **alter index MY_IX rebuild online;**
 - secondary index on IOT
 - overflow segment
 - bitmap
 - reverse
 - function-based
 - descending
 - compress
 - parallel
 - compute statistics
 - function-based / compressed secondary



Index Online – the new

- **Validate structure online*** ✓
- **Coalesce IOT** ✓
- **Secondary index**
 - UPDATE BLOCK REFERENCES
 - online operation
- **Bug 2307224**
 - online index rebuilds much slower than normal (fixed 9.2.0.2)
- *** - cannot be used for INDEX_STATS**

Table Online – the past

- *(IOT's...but that's cheating)*

Table Online – the new

- alter (heap) table EMP move online; ... *not yet*

```
SQL> create table EMP_NEW_DEFN(
  2     larger_empno      number(8) ,
  3     salary_plus_10pct number(7,2)) ;
```

```
SQL> exec  DBMS_REDEFINITION.START_REDEF_TABLE( -
          'SCOTT','EMP','EMP_NEW_DEFN', -
          'empno larger_empno, sal*1.1 salary_plus_10pct');
```

```
SQL> exec  DBMS_REDEFINITION.FINISH_REDEF_TABLE( -
          'SCOTT','EMP','EMP_NEW_DEFN');
```

```
SQL> desc EMP
```

Name	Null?	Type

LARGER_EMPNO		NUMBER(8)
SALARY_PLUS_10PCT		NUMBER(7,2)

Table Online – the new

- **dbms_redefinition**
 - primary key required* on source (*M_ROW\$\$, I_SNAP\$_tft)
 - interim table placed in same schema
 - manually copy constraints (NB: novalidate if pre-emptive)
 - no current materialised views
 - failures may require cleanup (“abort_redef_table”)
 - no transformations on primary key columns
 - no row filtering (NB: faking it with partitions)
 - no “special” columns (user-defined, long, etc)
 - no sorting of rows / build by index
 - no SYS, SYSTEM, GTT
 - parallel allowed
 - LOGGING always

Other “Online” Operations

- alter table EMP **rename column** JOB to JOBNO;

```
SQL> select count(*) from V$RESERVED_WORDS;
```

COUNT (*)

809

- create or replace [public] synonym
- alter trigger MY_TRIG rename to YOUR_TRIG

I/O

www.dmr.com.au / www.oracledba.co.uk



File Mapping

```
SQL> create tablespace MY_DATA
      2 datafile '/ora01/my_data_01.dbf' size ...
```

/ora01 = /dev/dsk/v01 =

■ File Mapping

- file_mapping = true
- EMC only?
- V\$MAP...
- *contributions welcome*

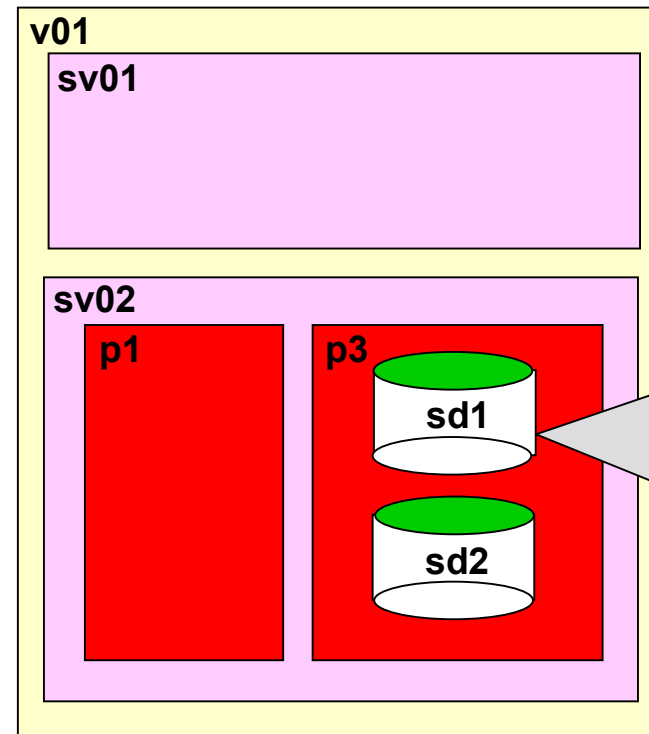


Table Compression

```
SQL> create table UC as
2   select * from dba_objects union all ...;    (x 9)
```

```
SQL> create table C1 COMPRESS as
2>  select * from UC;
```

```
SQL> create table C2 COMPRESS as
2>  select * from UC order by owner, object_type;
```

TABLE_NAME	PCT_FREE	BLOCKS
-----	-----	-----
UC	10	3567
C1	0	1742
C2	0	1682

Table Compression

- **Table compression**
 - eliminate common column values per block
 - Direct mode operations (don't touch that data!)
 - Info missing from DBA_TABLES
 - 11th bit on seg\$.spare1

Parallel DML

■ Parallel DML

- (8.0) partitioned tables only
- (9i) **non-partitioned** tables as well

■ The normal I/O rules apply!

1,000,000 heap update		200,000 IOT update	
▪ Serial	05:07	Serial	00:34
▪ Parallel 2	05:17	Parallel 2	00:27
▪ Parallel 4	05:48		

Partitioning – the past

```
create table EMP_PAR
(empno number(4),
ename varchar(10),
sal number(7,2),
deptno number(4) )
partition by range ( deptno )
subpartition by hash ( ename )
subpartitions 8
( partition p1 values less than (50)
  tablespace t1
    (subpartition p1_s1 tablespace ts1,
      subpartition p1_s2 tablespace ts2,
      subpartition p1_s3 tablespace ts3,
      subpartition p1_s4 tablespace ts4,
      subpartition p1_s5 tablespace ts5,
      subpartition p1_s6 tablespace ts6,
      subpartition p1_s7 tablespace ts7,
      subpartition p1_s8 tablespace ts8),
  partition p2 values less than (100)
    tablespace t2
```

```
(subpartition p2_s1 tablespace ts1,
  subpartition p2_s2 tablespace ts2,
  subpartition p2_s3 tablespace ts3,
  subpartition p2_s4 tablespace ts4,
  subpartition p2_s5 tablespace ts5,
  subpartition p2_s6 tablespace ts6,
  subpartition p2_s7 tablespace ts7,
  subpartition p2_s8 tablespace ts8),
partition p3 values less than (150)
  tablespace t3
    (subpartition p3_s1 tablespace ts1,
      subpartition p3_s2 tablespace ts2,
      subpartition p3_s3 tablespace ts3,
      subpartition p3_s4 tablespace ts4,
      subpartition p3_s5 tablespace ts5,
      subpartition p3_s6 tablespace ts6,
      subpartition p3_s7 tablespace ts7,
      subpartition p3_s8 tablespace ts8), ...
```

Partitioning – the past

partition p4 values less than (200)

tablespace t4

```
(subpartition p4_s1 tablespace ts1,
 subpartition p4_s2 tablespace ts2,
 subpartition p4_s3 tablespace ts3,
 subpartition p4_s4 tablespace ts4,
 subpartition p4_s5 tablespace ts5,
 subpartition p4_s6 tablespace ts6,
 subpartition p4_s7 tablespace ts7,
 subpartition p4_s8 tablespace ts8)
```

partition p5 values less than (250)

tablespace t5

```
(subpartition p5_s1 tablespace ts1,
 subpartition p5_s2 tablespace ts2,
 subpartition p5_s3 tablespace ts3,
 subpartition p5_s4 tablespace ts4,
 subpartition p5_s5 tablespace ts5,
 subpartition p5_s6 tablespace ts6,
 subpartition p5_s7 tablespace ts7,
 subpartition p5_s8 tablespace ts8)
```

partition p6 values less than (300)

tablespace t6

```
(subpartition p6_s1 tablespace ts1,
 subpartition p6_s2 tablespace ts2,
 subpartition p6_s3 tablespace ts3,
 subpartition p6_s4 tablespace ts4,
 subpartition p6_s5 tablespace ts5,
 subpartition p6_s6 tablespace ts6,
 subpartition p6_s7 tablespace ts7,
 subpartition p6_s8 tablespace ts8)
```

partition p7 values less than (350)

tablespace t7

```
(subpartition p7_s1 tablespace ts1,
 subpartition p7_s2 tablespace ts2,
 subpartition p7_s3 tablespace ts3,
 subpartition p7_s4 tablespace ts4,
 subpartition p7_s5 tablespace ts5,
 subpartition p7_s6 tablespace ts6,
 subpartition p7_s7 tablespace ts7,
 subpartition p7_s8 tablespace ts8));
```

Partitioning – the new

```
create table EMP_PAR
(empno number(4),
ename varchar(10),
sal number(7,2),
deptno number(4) )
partition by range ( deptno )
subpartition by hash ( ename )
```

SUBPARTITION TEMPLATE

```
(SUBPARTITION s1 tablespace ts1,
SUBPARTITION s2 tablespace ts2,
SUBPARTITION s3 tablespace ts3,
SUBPARTITION s4 tablespace ts4,
SUBPARTITION s5 tablespace ts5,
SUBPARTITION s6 tablespace ts6,
SUBPARTITION s7 tablespace ts7,
SUBPARTITION s8 tablespace ts8)
```

```
( partition p1 values less than (50),
partition p2 values less than (100),
partition p3 values less than (150),
partition p4 values less than (200),
partition p5 values less than (250),
partition p6 values less than (300),
partition p7 values less than (350));
```

- **DBA_SUBPARTITION_TEMPLATES**
- **DBA_LOB_TEMPLATES**

Global Indexes – the past

```
create table fin_results (
  dte date,
  dept varchar2(20),
  tot number )
partition by range (dte)
( partition p1 values
  less than (to_date('02012002','ddmmyyyy')),
partition p2 values
less than (to_date('03012002','ddmmyyyy')),
  ...
  partition p365 values
  less than (to_date('01012003','ddmmyyyy')));
```

global index fin_dept_ix
on fin_results (dept);

Unusable

Rebuild

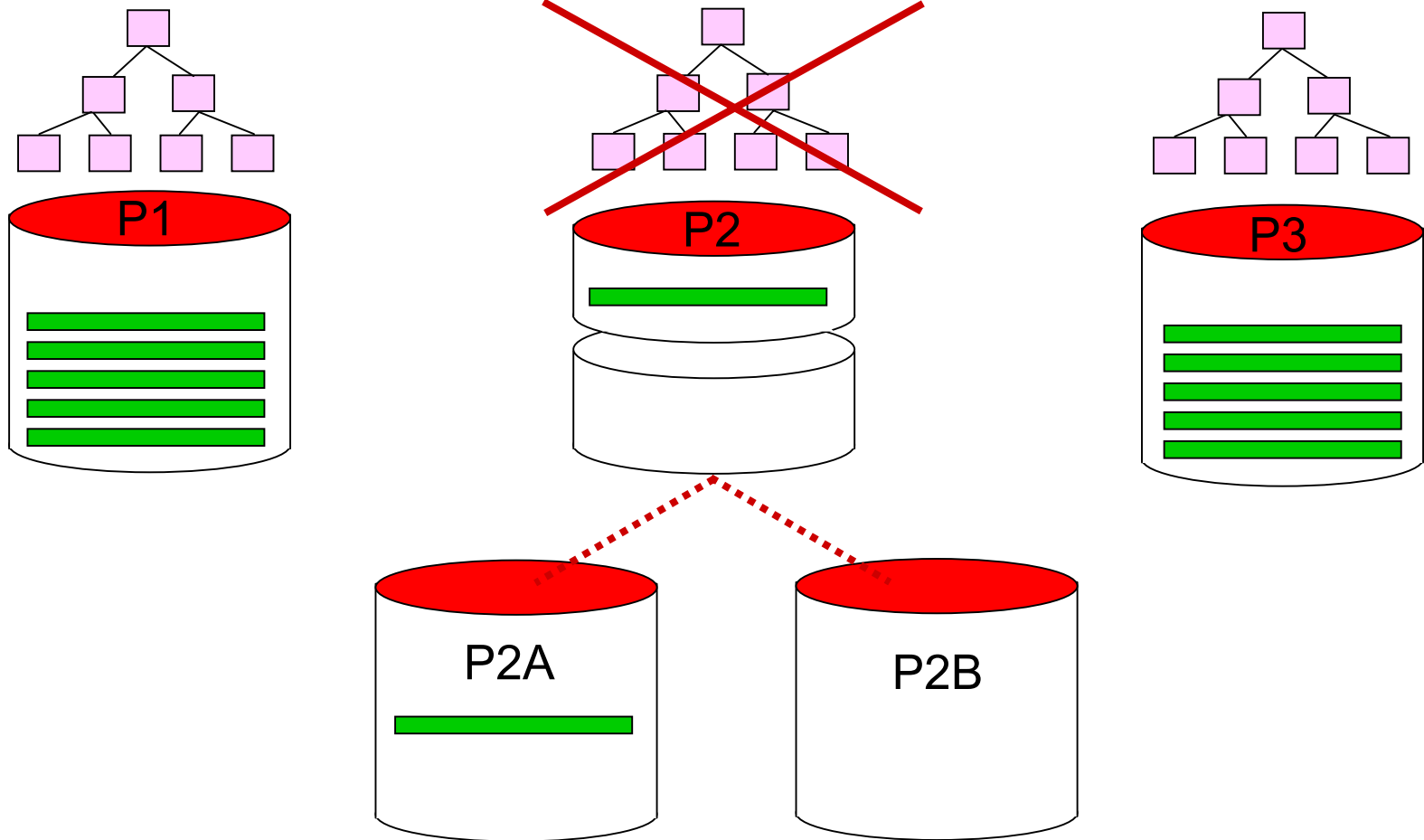
Global Indexes – the new

```
SQL> alter table fin_results
      2 drop partition p2
      3 update global indexes;
```

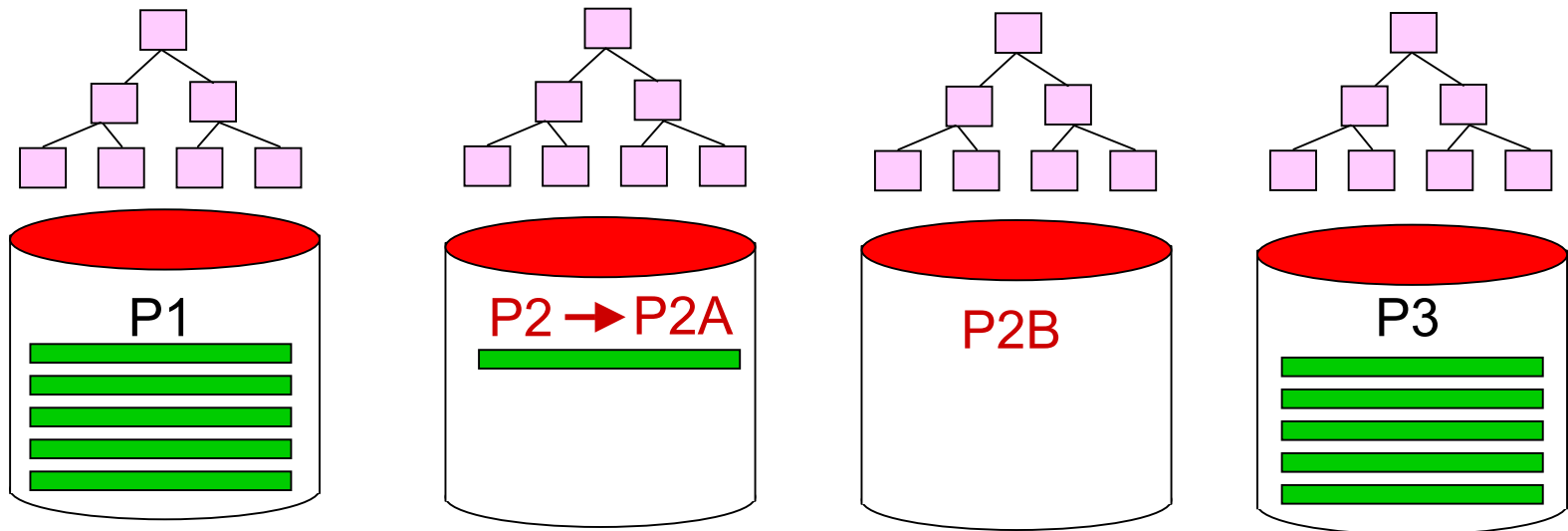
■ UPDATE GLOBAL INDEXES

- Online (≡ mass deletion, logging)
- not on IOT
- Take care with skewed partitions
- reverse / function-based problems (?)
- parallel allowed as well

Split Partitioning – the old



Split Partitioning – the new



■ Note

- existence check from P2 where $PAR_COL < split_val$
- existence check from P2 where $PAR_COL > split_val$

PL/SQL

www.dmr.com.au / www.oracledba.co.uk



Record based DML

```
declare
    D_REC dept%rowtype;
begin
    select *
    into    d_rec
    from    dept
    where   rownum = 1;
    insert into dept2
    values D_REC;
end;
```

```
declare
    D_REC dept%rowtype;
begin
    select *
    into    d_rec
    from    dept
    where   rownum = 1;
    update dept2
    set ROW = D_REC
    where deptno = d_rec.deptno;
end;
```

Arrays of arrays of ...

- Can be indexed tables or collections

```
declare
  type t_dim1 is table of number index by binary_integer;
  type t_dim2 is table of t_dim1 index by binary_integer;
  type t_dim3 is table of t_dim2 index by binary_integer;
  the_array t_dim3;
begin
  the_array(2)(3)(7) := 13;
  the_array(2)(4)(8) := 14;
  the_array(2)(4)(9) := 15;
  dbms_output.put_line(the_array.count);           -- =1
  dbms_output.put_line(the_array(2).count);        -- =2
  dbms_output.put_line(the_array(2)(4).count);     -- =2
end;
```

Associative Arrays

```
SQL> declare
  2   type elist is table of number index by varchar2(80);
  3   e emp_list;
  4   begin
  5     for i in ( select ename, empno from emp ) loop
  6       e(i.ename) := i.empno;
  7     end loop;
  8     dbms_output.put_line( e('KING') );
  9   end;
```

7839

- No bulk collect support yet
- Faster than numeric arrays for sparsely indexed data!

Dynamic + Bulk

```
SQL> declare
    ...
11  begin
12      execute immediate 'select '||colname||' from emp'
13      bulk collect into v_col_vals;
14
15      open c for 'select * from emp';
16      fetch c
17      bulk collect into v_emp_recs;
    ...
27  end;
```

- PL/SQL code should no longer be per-row processing

Pipelined PL/SQL

```
SQL> create type strlist as table of varchar2(80);
SQL> create or replace
  2  function MY_FUNC return strlist pipelined is
  3  begin
  4      pipe row('Some output');
  5      pipe row('Some more');
  6      return;
  7  end;
  8  /
```

```
SQL> select * from table(my_func);
```

COLUMN_VALUE

Some output

Some more

Pipelined PL/SQL

```
SQL> function WORK_WRAPPER return strlist pipelined
  2  begin
  3      pipe row('Start');
      ... Some work ...
  37  pipe row('Up to here');
      ... Some more work ...
  112  pipe row('Done');
  113  end;

SQL> set arraysize 1
SQL> select sysdate, * from table(WORK_WRAPPER);
```

PL/SQL Packages

- **UTL_FILE** no longer just basic “read/write”
 - no more “utl_file_dir = ...”
 - use directory object (including privileges)
 - **put_raw/get_raw** - support binary data
 - **fseek/fgetpos** - random positioning within files
 - **frename** - rename files
 - **fremove** - delete files
 - **fgetattr** - existence/length check
 - **fcopy** - copy files

PL/SQL Packages

- **utl_smtp / utl_encode**

- send email (kernelised)
- create uuencoded data

- **dbms_stats**

- convert_raw_value
- get_system_stats
- flush_..._monitoring_info
- statistics on external tables and GTT

← important !

PL/SQL Jobs

- **CJQ0**
 - “job queue coordinator”
- **“instant” execution**
- ***Jnnn* processes now transient**
- **up to 1000 concurrent jobs**
- **`dbms_job.user_export(jobno, :vc2);`**

Other PL/SQL

SQL> desc DBA_PROCEEDURES

Name	Null?	Type
OWNER	NOT NULL	VARCHAR2 (30)
OBJECT_NAME	NOT NULL	VARCHAR2 (30)
PROCEDURE_NAME		VARCHAR2 (30)
AGGREGATE		VARCHAR2 (3)
PIPELINED		VARCHAR2 (3)
IMPLTYPEOWNER		VARCHAR2 (30)
IMPLTYPENAME		VARCHAR2 (30)
PARALLEL		VARCHAR2 (3)
INTERFACE		VARCHAR2 (3)
DETERMINISTIC		VARCHAR2 (3)
AUTHID		VARCHAR2 (12)

LOBs

www.dmr.com.au / www.oracledba.co.uk



LOB's really are LOB's

- (In most cases) normal operators now apply

```
declare
  x clob; y clob;

  if x > y then ...
  if substr(x,25,36) like '%oracle%' then ...
  select lobcol||'some text', rpad(lobcol,20) from ...
  x := '<html>'||x||'</html>';
```

- Common sense still applies
 - Beware the hidden temporary lob

LOB's really are LOB's

- **Trigger manipulation**

```
create or replace
trigger LOB_TO_HTML
before insert on LOB_TABLE
for each row
begin
    :new.lobcol := '<b>' || :new.lobcol || '</b>';
end;
```

- **Function based indexes on LOB's**

What about old apps?

```
SQL> create table OLD_STYLE ( longcol long);
```

Table created.

```
SQL> insert into OLD_STYLE  
      2 values ('long stuff');
```

1 row created.

```
SQL> alter table old_style modify longcol clob;
```

Optimizer

www.dmr.com.au / www.oracledba.co.uk



System Stats

```
SQL> exec dbms_stats.gather_system_stats('INTERVAL',nnn);
```

```
SQL> exec dbms_stats.gather_system_stats('START|STOP');
```

- via DBMS_JOB, stored in AUX_STATS\$

SNAME	PNAME	PVAL1
-----	-----	-----
SYSSTATS_MAIN	SREADTIM	1.754
SYSSTATS_MAIN	MREADTIM	5.295
SYSSTATS_MAIN	CPUSPEED	89
SYSSTATS_MAIN	MBRC	16
SYSSTATS_MAIN	MAXTHR	6957056
SYSSTATS_MAIN	SLAVETHR	-1

“Analyze” is dead

■ “Smart” histograms

```
SQL> begin
  2   dbms_stats.gather_table_stats(
  3     ownname=>'SCOTT', tabname=>'EMP'
  4     METHOD_OPT=>'FOR COLUMNS SIZE AUTO');
  5   end;
```

■ “Smart” sampling

```
SQL> begin
  2   dbms_stats.gather_table_stats(
  3     ownname=>'SCOTT', tabname=>'EMP'
  4     ESTIMATE_PERCENT=>DBMS_STATS.AUTO_SAMPLE_SIZE);
  5   end;
```

Optimizer

■ Missing statistics

- optimizer_dynamic_sampling = 0 .. 10
- /*+ DYNAMIC_SAMPLING (table_alias nn)*/

■ Additional modes

- first_rows_nn (Note: first_rows $\neq$ first_rows_1)

■ Cursor sharing

- “similar” setting
- /*+ cursor_sharing_exact */
- bind variable peeking
- see Bjørn Engsig’s paper on OTN

Sampling / CPU costing

```
SQL> select *
      2   from   t1
      3   where  ln(object_id) > 0 and
      4   status = 'INVALID'
      5   /
```

33 rows selected.

Elapsed: 00:00:06.05

```
SQL> select /*+ cpu_costing */ *
      2   from   t1
      3   where  ln(object_id) > 0 and
      4   status = 'INVALID'
      5   /
```

Elapsed: 00:00:00.07

Monitoring

www.dmr.com.au / www.oracledba.co.uk



Locks – the past

- **V\$LOCK**
 - Real time information
- **V\$SYSTEM_EVENT / V\$SESSION_EVENT**
 - Total time spent waiting on enqueue
- **X\$KSQST**
 - Count (but not duration) by lock type

SQL> desc **X\$KSQST**

Name	Null?	Type
-----	-----	-----
KSQSTTYP		VARCHAR2 (2)
KSQSTGET		NUMBER
KSQSTWAT	← gets	NUMBER
	← waits	

Locks – the new

- **V\$ENQUEUE_STAT**
 - Failures as well as successful enqueue requests
 - Cumulative wait time

SQL> desc V\$ENQUEUE_STAT

Name	Null?	Type

EQ_TYPE		VARCHAR2 (2)
TOTAL_REQ#		NUMBER
TOTAL_WAIT#		NUMBER
SUCC_REQ#		NUMBER
FAILED_REQ#		NUMBER
CUM_WAIT_TIME		NUMBER

New columns

SQL> desc V\$PROCESS

...

LATCHWAIT

LATCHSPIN

PGA_USED_MEM

PGA_ALLOC_MEM

PGA_FREEABLE_MEM

PGA_MAX_MEM

} *v\$latch_children*

} *auto or manual pga*

SQL> desc V\$SQL_PLAN

...

DEPTH

CPU_COST

IO_COST

TEMP_SPACE

ACCESS_PREDICATES

FILTER_PREDICATES

New columns

```
SQL> desc V$SQL
```

```
...
FETCHES
CPU_TIME
ELAPSED_TIME
```

```
SQL> desc V$DB_OBJECT_CACHE (and V$SQL, V$SQLAREA)
```

```
...
CHILD_LATCH
```

```
SQL> desc V$LATCH (and V$LATCH_CHILDREN, V$LATCH_PARENT)
```

```
...
WAIT_TIME
```

```
SQL> desc INDEX_STATS
```

```
...
OPT_CMPR_COUNT
OPT_CMPR_PCTSAVE
```

Advisories

- **Advisory / “historical” information**
 - **V\$SHARED_POOL_ADVICE**
 - **V\$DB_CACHE_ADVICE**
 - **V\$MTTR_TARGET_ADVICE**
 - **V\$UNDOSTAT**

 - **V\$SQL_WORKAREA**
 - **V\$SQL_WORKAREA_ACTIVE**
 - **V\$SQL_WORKAREA_HISTOGRAM**

 - **V\$PGASTAT**
 - **V\$PGA_TARGET_ADVICE**
 - **V\$PGA_TARGET_ADVICE_HISTOGRAM**

Advisories

```
SQL> select begin_time, end_time, undoblks,
2          maxquerylen, maxconcurrency
3  from    V$UNDOSTAT;
```

BEGIN_TIME	END_TIME	UNDOBLKS	MAXQUERYLEN	MAXCONCURRENCY
19/11-15:05	19/11-15:16	0	0	0
19/11-14:55	19/11-15:05	1	0	2
19/11-14:25	19/11-14:55	0	0	0
19/11-14:15	19/11-14:25	0	3	0
19/11-14:05	19/11-14:15	0	0	0
19/11-13:55	19/11-14:05	8417	21	1
19/11-13:45	19/11-13:55	8842	17	4
19/11-13:35	19/11-13:45	7975	10	2
19/11-13:25	19/11-13:35	8728	58	1
19/11-13:15	19/11-13:25	9596	2	1

Segment usage

```
SQL> select count(*) from sys.source$;      (x3)
```

```
SQL> select object_name, statistic_name, value
2   from V$SEGMENT_STATISTICS      (or V$SEGSTAT)
3   where object_name = 'SOURCE$';
```

OBJECT_NAME	STATISTIC_NAME	VALUE
-----	-----	-----
SOURCE\$	logical reads	11216
SOURCE\$	buffer busy waits	210
SOURCE\$	db block changes	32
SOURCE\$	physical reads	10365
SOURCE\$	physical writes	0
SOURCE\$	physical reads direct	0
SOURCE\$	physical writes direct	0
SOURCE\$	ITL waits	0
SOURCE\$	row lock waits	0

Cursors usage

SQL> desc V\$SQL_SHARED_CURSOR* (* 8.1.7)

Name	Null?	Type

ADDRESS		RAW (8)
KGLHDPAR		RAW (8)
UNBOUND_CURSOR		VARCHAR2 (1)
...		
OPTIMIZER_MISMATCH		VARCHAR2 (1)
OUTLINE_MISMATCH		VARCHAR2 (1)
LITERAL_MISMATCH		VARCHAR2 (1)
SEC_DEPTH_MISMATCH		VARCHAR2 (1)
EXPLAIN_PLAN_CURSOR		VARCHAR2 (1)
BUFFERED_DML_MISMATCH		VARCHAR2 (1)
PDML_ENV_MISMATCH		VARCHAR2 (1)
...		

Plans

SQL> desc V\$SQL_PLAN_STATISTICS

Name	Null?	Type
-----	-----	-----
ADDRESS		RAW (8)
HASH_VALUE		NUMBER
CHILD_NUMBER		NUMBER
OPERATION_ID		NUMBER
...		
LAST_OUTPUT_ROWS		NUMBER
OUTPUT_ROWS		NUMBER
LAST_CR_BUFFER_GETS		NUMBER
CR_BUFFER_GETS		NUMBER
LAST_ELAPSED_TIME		NUMBER
ELAPSED_TIME		NUMBER



Monitoring Levels

- `statistics_level = basic | typical | all`

```
SQL> select STATISTICS_NAME, STATISTICS_VIEW_NAME,
2          ACTIVATION_LEVEL from V$STATISTICS_LEVEL;
```

STATISTICS_NAME	STATISTICS_VIEW_NAME	ACTIVAT
-----	-----	-----
Buffer Cache Advice	V\$DB_CACHE_ADVICE	TYPICAL
MTTR Advice	V\$MTTR_TARGET_ADVICE	TYPICAL
Timed Statistics		TYPICAL
Timed OS Statistics		ALL
Segment Level Statistics	V\$SEGSTAT	TYPICAL
PGA Advice	V\$PGA_TARGET_ADVICE	TYPICAL
Plan Execution Statistics	V\$SQL_PLAN_STATISTICS	ALL
Shared Pool Advice	V\$SHARED_POOL_ADVICE	TYPICAL

VPD / RLS / FGAC – the old

```
SQL> conn scott/tiger
```

```
SQL> select count(*) from SCOTT.SECURE_EMP;
```

```

COUNT (*)
-----
          7

```

```
SQL> conn peter/smith
```

```
SQL> select count(*) from SCOTT.SECURE_EMP;
```

```

COUNT (*)
-----
         14

```

- Something's going on...event 10730

VPD / RLS / FGAC – the new

- V\$VPD_POLICY

```
SQL> select OBJECT_NAME, POLICY, PREDICATE
       2 from V$VPD_POLICY
```

```
OBJECT_NAME POLICY      PREDICATE
-----
SECURE_EMP  OWN_EMPS    usr = USER
```

- “EXEMPT ACCESS POLICY” privilege
- DBA_POLICY_GROUPS / DBA_POLICY_CONTEXTS
- Change in execution of policy (_dynamic_qls_policies)

Other Monitoring

- **ORA-1555's now appear in the alert log**
- **Statspack has a new levels**
 - Level 6: Captures SQL plan information
 - Level 7: Captures segment level statistics
- **Tkprof show wait statistics**

```
SQL> alter session set
      2   events='10046 trace name context forever, level 8';
```

```
SQL> select * from sec_test where z < 123;
```

TKPROF output

```
select * from sec_test where z < 123
```

call	count	cpu	elapsed	disk	query	current	rows
-----	-----	-----	-----	-----	-----	-----	-----
Parse	1	0.01	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	10	0.04	0.19	109	126	0	122
-----	-----	-----	-----	-----	-----	-----	-----
total	12	0.05	0.19	109	126	0	122

Elapsed times include waiting on following events:

Event waited on	Times	Max. Wait	Total Waited
-----	Waited	-----	-----
SQL*Net message to client	10	0.00	0.00
db file sequential read	4	0.01	0.01
db file scattered read	7	0.02	0.11
SQL*Net message from client	10	2.96	4.01

Backup / Recovery

www.dmr.com.au / www.oracledba.co.uk



Controlfiles – the past

```
SQL> alter database backup controlfile to trace;
```

(Wonder / Wander about the user dump dest...)

```
CREATE CONTROLFILE REUSE DATABASE "FAM" NORESETLOGS  
    ARCHIVELOG
```

```
...
```

Controlfiles – the new

```
SQL> alter database backup controlfile to trace
      2      AS FILE '/tmp/my_trace_file.trc' RESETLOGS;
```

(list of archival related parameters)

```
CREATE CONTROLFILE REUSE DATABASE "XYZ"
      RESETLOGS ARCHIVELOG
```

...

- Specifying no log mode gives TWO 'create controlfile' commands in the trace file
- Beware tempfiles and OMF

Instance Recovery

■ The past

- Instance crash during backup
- Recovery “required” on restart
- Switch off hot backup on appropriate files

■ The new

```
SQL> alter database end backup;
```

- No error if redundant

Archiver Control

■ The past

- `00,30 * * * * /usr/local/bin/switch_logfile.sh`

■ The new

- `archive_lag_target = 1800` (30 mins)
- Dynamic
- No “database up?” worries
- Reset by activity-based log switch

Archiver Control

■ The past

- `grep 'ORA-00257' alert.log`
- `00,30 * * * * /usr/local/bin/check_disk_free.sh`
- `00,30 * * * * /usr/local/bin/move_to_remote.sh`

■ The new

- `log_archive_dest_1 =`
`'LOCATION=location=...`
`QUOTA_SIZE=nnn (512)`
`ALTERNATE=log_archive_dest_2'`
- `log_archive_dest_state_2=ALTERNATE`
- Archive repository (standby with no datafiles)

Standby Database

■ The past

- NOLOGGING grief
- db_file_name_convert = ('src', 'dst')
- Messy SQLPLUS session

■ The new

- alter tablespace|database **force logging**
- db_file_name_convert = ('src1', 'dst2', '**src2**', '**dst2**',...)
- recover managed standby **disconnect**

■ DataGuard (even 8i) very under-rated

Standby Database

- **Drop datafiles automatically**
 - On Primary
 - drop tablespace T1 including contents **and datafiles**
 - And on the standby
 - **standby_file_management = auto**

Cloning

■ The past

- Clone outside RMAN
- Restrictions on backup that database (RMAN)
 - DBID issues

■ The new

- DBNEWID utility
- nid target=user/pass
 - Then backup!!!

Resource Manager

- **(Finally?) some useful controls**
 - Estimated query execution time
 - Max. allowed concurrent sessions
 - ACTIVE $\neq$ CONNECTED
 - Queueing timeouts
 - Max. allowed undo
 - Entire group tally
 - Reset at transaction end

Resource Manager

```
DRM.create_pending_area();  
DRM.create_plan ('MY_PLAN');  
DRM.create_consumer_group ('ALL');
```

```
DRM.create_plan_directive(  
    PLAN=>'MY_PLAN',  
    GROUP_OR_SUBPLAN=>'ALL',  
    MAX_EST_EXEC_TIME=>10);
```

```
DRM.create_plan_directive('MY_PLAN', 'OTHER_GROUPS');  
DRM.validate_pending_area;  
DRM.submit_pending_area;  
DRM_PRIVS.grant_switch_consumer_group('SCOTT', 'ALL', true)  
DRM.set_initial_consumer_group('SCOTT', 'ALL');
```

Resource Manager

```
SQL> exec DRM.set_initial_consumer_group('SCOTT','ALL');
```

```
SQL> alter system set resource_manager_plan = MY_PLAN;
```

```
SQL> conn SCOTT/TIGER
```

Connected.

```
SQL> select count(*) from MY_BIG_TAB;
```

```
select count(*) from MY_BIG_TAB;
```

```
*
```

ERROR at line 1:

ORA-07455: estimated execution time (17 secs), exceeds limit (10 secs)

■ Scepticism on Resource Manager?

References

- **www.oaktable.net members**
- **Oracle 9.2 Documentation**
 - Administrator's Guide
 - Application Developer's Guide - Large Objects (LOBs)
 - New Features
 - Performance Tuning Guide and Reference
 - PL/SQL User's Guide and Reference
 - SQL Reference
 - Supplied PL/SQL Packages and Types Reference
 - User-Managed Backup and Recovery Guide
 - Utilities
- **Oracle 9i New Features – Howard Rogers**

References

■ Metalink

- 190077.1
- 212201.1
- 189929.1
- 175434.1
- 70067.1
- 177407.1
- 201342.1
- 191577.1
- 174504.1
- 50875.1
- 147806.1
- 62172.1
- 149016.1
- 149118.1
- 199419.1
- 190284.1
- 196939.1
- 198120.1
- 170649.1
- 149560.1
- 45042.1
- 61998.1
- 148518.1

What we have covered...



- Let someone else invent the wheel
- Your job to make sure it will roll...