Oracle 9i Features*
(* The ones they DIDN’T tell you about)

Presented by
CONNOR MCDONALD
Senior Consultant
DMR Consulting, Asia Pacific

Sep 2003
Advertisements

- Performance / Configuration
- Trouble making shooting
- http://www.oracledba.co.uk
- 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
Avoiding the headlines

- Automatic undo
- Memory resizing
- Multiple block sizes
- Stored parameter file
- List partitioning
- External tables
- Flashback
- ETL (Merge/Insert-All)
- ANSI joins
- Resumable space alloc.
- Data Guard
- Segment space management
- Skip-scan indexes
- PL/SQL native compile
- LogMiner
- Oracle managed files
- Index monitoring
- Workspace management
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
Lets start with SQLPlus

- **SQLPLUS -L**
  - one attempt at user/pass only

- **Connection details available**
  ```sql
  SQL> SET SQLPROMPT '&_CONNECT_IDENTIFIER > '
  mysid >
  ```

- **URL’s for scripts**
  ```sql
  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
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

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>NESTED LOOPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 2</td>
<td>TABLE ACCESS FULL</td>
<td>EMP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 3</td>
<td>INDEX UNIQUE SCAN</td>
<td>PK_DEPT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>T1</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>* 2</td>
<td>INDEX RANGE SCAN</td>
<td>T1X_1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

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

Index (I_COL,I_CHAR) but datatype mismatch
Explain Plan – the new

- Automatically adjusts its output (format=TYPICAL)

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

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>...</th>
<th>Pstart</th>
<th>Pstop</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PARTITION RANGE ITERATOR</td>
<td></td>
<td>...</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>TABLE ACCESS FULL</td>
<td>PAR_TABLE</td>
<td>...</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
```
Explain Plan – the new

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

<table>
<thead>
<tr>
<th>Id</th>
<th>Pstart</th>
<th>Pstop</th>
<th>TQ</th>
<th>IN-OUT</th>
<th>PQ Distrib</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>90</td>
<td>PCWP</td>
<td>QC (RANDOM)</td>
</tr>
<tr>
<td>* 2</td>
<td>1</td>
<td>2</td>
<td>90</td>
<td>P-&gt;S</td>
<td>QC (RANDOM)</td>
</tr>
</tbody>
</table>
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  /
```

<table>
<thead>
<tr>
<th>DEPTNO</th>
<th>MIN(SAL)</th>
<th>EMPNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1300</td>
<td>7934</td>
</tr>
<tr>
<td>20</td>
<td>800</td>
<td>7369</td>
</tr>
<tr>
<td>30</td>
<td>950</td>
<td>7900</td>
</tr>
</tbody>
</table>

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))))

AVG(SAL) -------
  1550
```

Cartesian product

“What the ?”
Analytics – the new

- “What is the median salary?”

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

<table>
<thead>
<tr>
<th>DEPTNO</th>
<th>JOB</th>
<th>MGR</th>
<th>SUM(SAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLERK</td>
<td>7698</td>
<td>950</td>
<td></td>
</tr>
<tr>
<td>CLERK</td>
<td>7902</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>MANAGER</td>
<td>7839</td>
<td>8275</td>
<td></td>
</tr>
<tr>
<td>SALESMAN</td>
<td>7698</td>
<td>5600</td>
<td></td>
</tr>
<tr>
<td>PRESIDENT</td>
<td></td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>8750</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>9400</td>
<td></td>
</tr>
</tbody>
</table>

29025
WITH clause

- WITH clause for complex subqueries

```sql
WITH dept_salaries AS (  
    SELECT dname, SUM(sal) dept_sal  
    FROM emp e, dept d  
    WHERE e.deptno = d.deptno  
    GROUP BY dname),  
  avg_sal AS ( SELECT AVG(dept_sal) asal  
                FROM dept_salaries)  
  
  SELECT * FROM dept_salaries d, avg_sal a  
  WHERE d.dept_sal > a.asal  
  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
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

<table>
<thead>
<tr>
<th>FULL_TREE</th>
<th>EMPNO</th>
<th>DNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-KING</td>
<td>7839</td>
<td>ACCOUNTING</td>
</tr>
<tr>
<td>-KING-BLAKE</td>
<td>7698</td>
<td>SALES</td>
</tr>
<tr>
<td>-KING-BLAKE-ALLEN</td>
<td>7499</td>
<td>SALES</td>
</tr>
<tr>
<td>-KING-BLAKE-JAMES</td>
<td>7900</td>
<td>SALES</td>
</tr>
<tr>
<td>-KING-BLAKE-MARTIN</td>
<td>7654</td>
<td>SALES</td>
</tr>
<tr>
<td>-KING-BLAKE-TURNER</td>
<td>7844</td>
<td>SALES</td>
</tr>
<tr>
<td>-KING-BLAKE-WARD</td>
<td>7521</td>
<td>SALES</td>
</tr>
<tr>
<td>-KING-CLARK</td>
<td>7782</td>
<td>ACCOUNTING</td>
</tr>
<tr>
<td>-KING-CLARK-MILLER</td>
<td>7934</td>
<td>ACCOUNTING</td>
</tr>
</tbody>
</table>

- 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
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 ≠ 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;

<table>
<thead>
<tr>
<th>CONSTRAINT_NAME</th>
<th>OBJECT_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP_MGR_HIRE_UQ</td>
<td>EMP_MGR_HIRE_SAL_IX</td>
</tr>
</tbody>
</table>
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';

<table>
<thead>
<tr>
<th>CONSTRAINT_NAME</th>
<th>INDEX_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP_MGR_HIRE_UQ</td>
<td>EMP_MGR_HIRE_SAL_IX</td>
</tr>
</tbody>
</table>
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

\[\text{begin} \]
\[
\text{DBMS_METADATA.SET_TRANSFORM_PARAM(}
\text{DBMS_METADATA.SESSION_TRANSFORM,}
\text{'STORAGE',}
\text{false)};
\]
\[\text{end;}\]

\text{SQL}\geq\text{ select dbms_metadata.get_ddl('TABLE','EMP','SCOTT')}

2 from dual;
DDL Extraction – the new

```sql
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

<table>
<thead>
<tr>
<th>ASSOCIATION</th>
<th>LIBRARY</th>
<th>ROLLBACK_SEGMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIT</td>
<td>MATERIALIZED_VIEW</td>
<td>SEQUENCE</td>
</tr>
<tr>
<td>AUDIT_OBJ</td>
<td>MATERIALIZED_VIEW_LOG</td>
<td>SYNONYM</td>
</tr>
<tr>
<td>CLUSTER</td>
<td>OBJECT_GRANT</td>
<td>SYSTEM_GRANT</td>
</tr>
<tr>
<td>COMMENT</td>
<td>OPERATOR</td>
<td>TABLE</td>
</tr>
<tr>
<td>CONSTRAINT</td>
<td>OUTLINE</td>
<td>TABLESPACE</td>
</tr>
<tr>
<td>CONTEXT</td>
<td>PACKAGE</td>
<td>TABLESPACE_QUOTA</td>
</tr>
<tr>
<td>DB_LINK</td>
<td>PACKAGE_SPEC</td>
<td>TRIGGER</td>
</tr>
<tr>
<td>DEFAULT_ROLE</td>
<td>PACKAGE_BODY</td>
<td>TRUSTED_DB_LINK</td>
</tr>
<tr>
<td>DIMENSION</td>
<td>PROCEDURE</td>
<td>TYPE</td>
</tr>
<tr>
<td>DIRECTORY</td>
<td>PROFILE</td>
<td>TYPE_SPEC</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>PROFILE</td>
<td>TYPE_BODY</td>
</tr>
<tr>
<td>INDEX</td>
<td>REF_CONSTRAINT</td>
<td>USER</td>
</tr>
<tr>
<td>INDEXTYPE</td>
<td>ROLE</td>
<td>VIEW</td>
</tr>
<tr>
<td>JAVA_SOURCE</td>
<td>ROLE_GRANT</td>
<td>XMLSCHEMA</td>
</tr>
</tbody>
</table>
Online Operations
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 then 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$$, I_SNAP$$_ttt)
  - 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
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>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>PCT_FREE</th>
<th>BLOCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC</td>
<td>10</td>
<td>3567</td>
</tr>
<tr>
<td>C1</td>
<td>0</td>
<td>1742</td>
</tr>
<tr>
<td>C2</td>
<td>0</td>
<td>1682</td>
</tr>
</tbody>
</table>
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
    - Serial 05:07
    - Parallel 2 05:17
    - Parallel 4 05:48
  - 200,000 IOT update
    - Serial 00:34
    - Parallel 2 00:27
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

```sql
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')));

create "global" index fin_dept_ix
    on fin_results (dept);
```

Unusable
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
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
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
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
SQL> function WORK.WRAPPER return strlist pipelined
   2  begin
   3  pipe row('Start');
   4  ... Some work ...
   5  pipe row('Up to here');
   6  ... Some more work ...
   7  pipe row('Done');
   8  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 ➤ important!
  - flush__monitoring_info
  - statistics on external tables and GTT
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_PROCEDURES**

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>NOT NULL</td>
<td>VARCHAR2 (30)</td>
</tr>
<tr>
<td>OBJECT_NAME</td>
<td>NOT NULL</td>
<td>VARCHAR2 (30)</td>
</tr>
<tr>
<td>PROCEDURE_NAME</td>
<td>NOT NULL</td>
<td>VARCHAR2 (30)</td>
</tr>
<tr>
<td>AGGREGATE</td>
<td></td>
<td>VARCHAR2 (3)</td>
</tr>
<tr>
<td>PIPELINED</td>
<td></td>
<td>VARCHAR2 (3)</td>
</tr>
<tr>
<td>IMPLTYPEOWNER</td>
<td></td>
<td>VARCHAR2 (30)</td>
</tr>
<tr>
<td>IMPLTYPENAME</td>
<td></td>
<td>VARCHAR2 (30)</td>
</tr>
<tr>
<td>PARALLEL</td>
<td></td>
<td>VARCHAR2 (3)</td>
</tr>
<tr>
<td>INTERFACE</td>
<td></td>
<td>VARCHAR2 (3)</td>
</tr>
<tr>
<td>DETERMINISTIC</td>
<td></td>
<td>VARCHAR2 (3)</td>
</tr>
<tr>
<td>AUTHID</td>
<td></td>
<td>VARCHAR2 (12)</td>
</tr>
</tbody>
</table>
LOBs
LOB’s really are LOB’s

- (In most cases) normal operators now apply

```sql
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**
  
  ```sql
  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
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$

<table>
<thead>
<tr>
<th>SNAME</th>
<th>PNAME</th>
<th>PVAL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSSTATS_MAIN</td>
<td>SREADTIM</td>
<td>1.754</td>
</tr>
<tr>
<td>SYSSTATS_MAIN</td>
<td>MREADTIM</td>
<td>5.295</td>
</tr>
<tr>
<td>SYSSTATS_MAIN</td>
<td>CPUSPEED</td>
<td>89</td>
</tr>
<tr>
<td>SYSSTATS_MAIN</td>
<td>MBRC</td>
<td>16</td>
</tr>
<tr>
<td>SYSSTATS_MAIN</td>
<td>MAXTHR</td>
<td>6957056</td>
</tr>
<tr>
<td>SYSSTATS_MAIN</td>
<td>SLAVETHR</td>
<td>-1</td>
</tr>
</tbody>
</table>
“Analyze” is dead

- “Smart” histograms
  ```sql
  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
  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
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
```
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSQSTTYP</td>
<td></td>
<td>VARCHAR2 (2)</td>
</tr>
<tr>
<td>KSQSTGET</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>KSQSTWAT</td>
<td></td>
<td>NUMBER</td>
</tr>
</tbody>
</table>

gets  
waits
## Locks – the new

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

```sql
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

SQL> desc V$SQL_PLAN
...
   DEPTH
   CPU_COST
   IO_COST
   TEMP_SPACE
   ACCESS_PREDICATES
   FILTER_PREDICATES

v$latch_children
auto or manual pga
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, maxquerylen, maxconcurrency
    2 from V$UNDOSTAT;

<table>
<thead>
<tr>
<th>BEGIN_TIME</th>
<th>END_TIME</th>
<th>UNDOBLKS</th>
<th>MAXQUERYLEN</th>
<th>MAXCONCURRENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>19/11-15:05</td>
<td>19/11-15:16</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19/11-14:55</td>
<td>19/11-15:05</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>19/11-14:25</td>
<td>19/11-14:55</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19/11-14:15</td>
<td>19/11-14:25</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>19/11-14:05</td>
<td>19/11-14:15</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19/11-13:55</td>
<td>19/11-14:05</td>
<td>8417</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td><strong>19/11-13:45</strong></td>
<td><strong>19/11-13:55</strong></td>
<td><strong>8842</strong></td>
<td><strong>17</strong></td>
<td><strong>4</strong></td>
</tr>
<tr>
<td>19/11-13:35</td>
<td>19/11-13:45</td>
<td>7975</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>19/11-13:25</td>
<td>19/11-13:35</td>
<td>8728</td>
<td>58</td>
<td>1</td>
</tr>
<tr>
<td>19/11-13:15</td>
<td>19/11-13:25</td>
<td>9596</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
## Segment usage

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

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

<table>
<thead>
<tr>
<th>OBJECT_NAME</th>
<th>STATISTIC_NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURCE$</td>
<td>logical reads</td>
<td>11216</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>buffer busy waits</td>
<td>210</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>db block changes</td>
<td>32</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>physical reads</td>
<td>10365</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>physical writes</td>
<td>0</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>physical reads direct</td>
<td>0</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>physical writes direct</td>
<td>0</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>ITL waits</td>
<td>0</td>
</tr>
<tr>
<td>SOURCE$</td>
<td>row lock waits</td>
<td>0</td>
</tr>
</tbody>
</table>
## Cursors usage

```sql
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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESS</td>
<td></td>
<td>RAW(8)</td>
</tr>
<tr>
<td>HASH_VALUE</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>CHILD_NUMBER</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>OPERATION_ID</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST_OUTPUT_ROWS</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>OUTPUT_ROWS</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>LAST_CR_BUFFER_GETS</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>CR_BUFFER_GETS</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>LAST_ELAPSED_TIME</td>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>ELAPSED_TIME</td>
<td></td>
<td>NUMBER</td>
</tr>
</tbody>
</table>
Monitoring Levels

- statistics_level = basic | typical | all

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

<table>
<thead>
<tr>
<th>STATISTICS_NAME</th>
<th>STATISTICS_VIEW_NAME</th>
<th>ACTIVATION_LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Cache Advice</td>
<td>V$DB_CACHE_ADVICE</td>
<td>TYPICAL</td>
</tr>
<tr>
<td>MTTR Advice</td>
<td>V$MTTR_TARGET_ADVICE</td>
<td>TYPICAL</td>
</tr>
<tr>
<td>Timed Statistics</td>
<td></td>
<td>TYPICAL</td>
</tr>
<tr>
<td>Timed OS Statistics</td>
<td></td>
<td>ALL</td>
</tr>
<tr>
<td>Segment Level Statistics</td>
<td>V$SEGSTAT</td>
<td>TYPICAL</td>
</tr>
<tr>
<td>PGA Advice</td>
<td>V$PGA_TARGET_ADVICE</td>
<td>TYPICAL</td>
</tr>
<tr>
<td>Plan Execution Statistics</td>
<td>V$SQL_PLAN_STATISTICS</td>
<td>ALL</td>
</tr>
<tr>
<td>Shared Pool Advice</td>
<td>V$SHARED_POOL_ADVICE</td>
<td>TYPICAL</td>
</tr>
</tbody>
</table>
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

- “EXEMPLARY ACCESS POLICY” privilege
- DBA_POLICY_GROUPS / DBA_POLICY_CONTEXTS
- Change in execution of policy ( _dynamic_rls_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
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:

<table>
<thead>
<tr>
<th>Event waited on</th>
<th>Times</th>
<th>Max. Wait</th>
<th>Total Waited</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL*Net message to client</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>db file sequential read</td>
<td>4</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>db file scattered read</td>
<td>7</td>
<td>0.02</td>
<td>0.11</td>
</tr>
<tr>
<td>SQL*Net message from client</td>
<td>10</td>
<td>2.96</td>
<td>4.01</td>
</tr>
</tbody>
</table>
Backup / Recovery

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 ≠ CONNECTED
    - Queueing timeouts
  - Max. allowed undo
    - Entire group tally
    - Reset at transaction end
Resource Manager

```sql
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...