



# Oracle 10G Features for Developers

**SAGE Computing Services**  
Customised Oracle Training Workshops  
and Consulting  
[www.sagecomputing.com.au](http://www.sagecomputing.com.au)

**Penny Cookson - Managing Director**

[www.sagecomputing.com.au](http://www.sagecomputing.com.au)

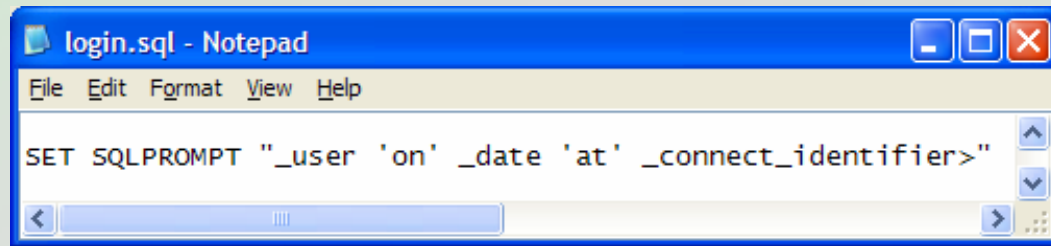
# Agenda

- **SQL\*Plus**
- **Recycle Bin**
- **Merge Syntax**
- **Connect By**
- **Numeric datatypes**
- **Flashback**
- **Regular Expressions**
- **The MODEL Clause**
- **Commit Processing**
- **DML Error Handling**
- **PL/SQL Warnings**
- **Conditional Compilation**
- **Bulk Binding**
- **Auditing**
- **Scheduler**
- **Optimisation**

# SQL\*Plus New Features

- **GLOGIN.sql and LOGIN.sql** execute for each new session
- **New variables for SET SQLPROMPT**

**Example:**

A screenshot of a Notepad window titled "login.sql - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the SQL command: `SET SQLPROMPT "_user 'on' _date 'at' _connect_identifier>"`. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
SYS on 28-AUG-04 at ora10g> CONNECT train/train@ora10g
Connected.
```

```
TRAIN on 28-AUG-04 at ora10g>
```

# SQL\*Plus – File Saving

```
SQL>SPOOL myresults
```

```
SQL>SELECT * FROM resource_types
```

```
2 WHERE CODE LIKE 'C%';
```

```
CODE NAME
```

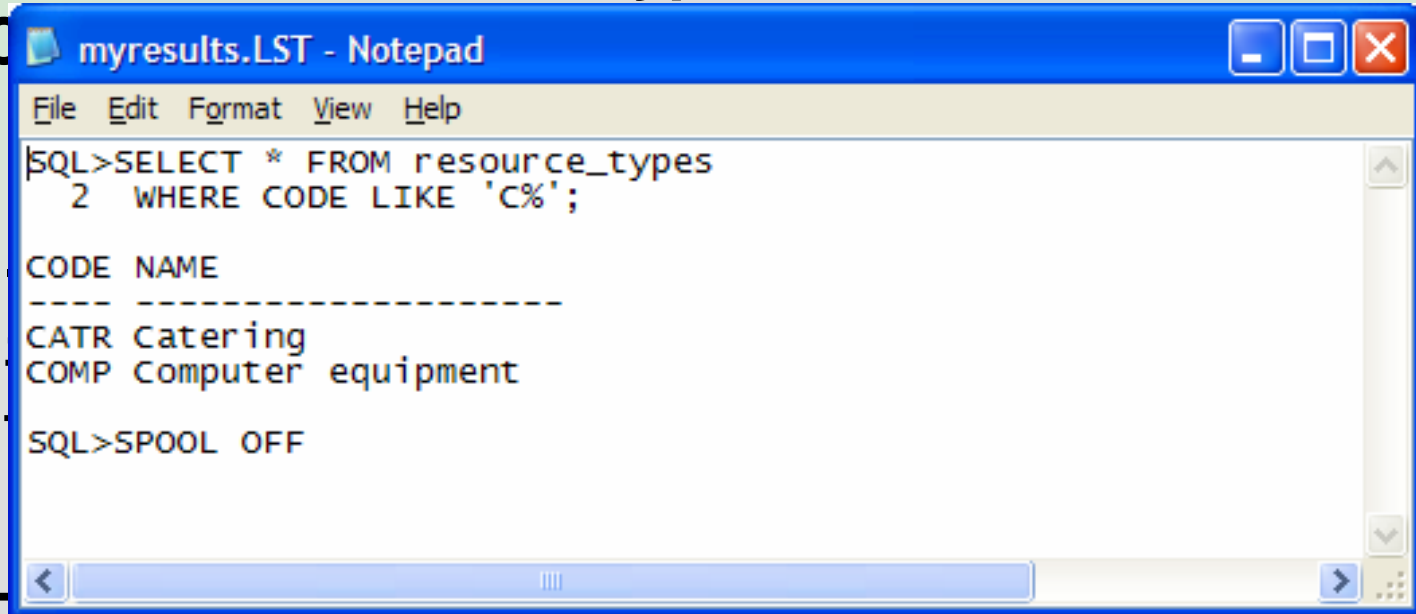
```
-----
```

```
CATR Catering
```

```
COMP Computer equipment
```

```
SQL>SPOOL OFF
```

```
SQL>ed myresults.lst
```

A screenshot of a Notepad window titled "myresults.LST - Notepad". The window contains the following text:

```
File Edit Format View Help
SQL>SELECT * FROM resource_types
2 WHERE CODE LIKE 'C%';

CODE NAME
-----
CATR Catering
COMP Computer equipment

SQL>SPOOL OFF
```

# SQL\*Plus – File Saving

SQL>SPOOL

SQL>SELEC

2 WHERE

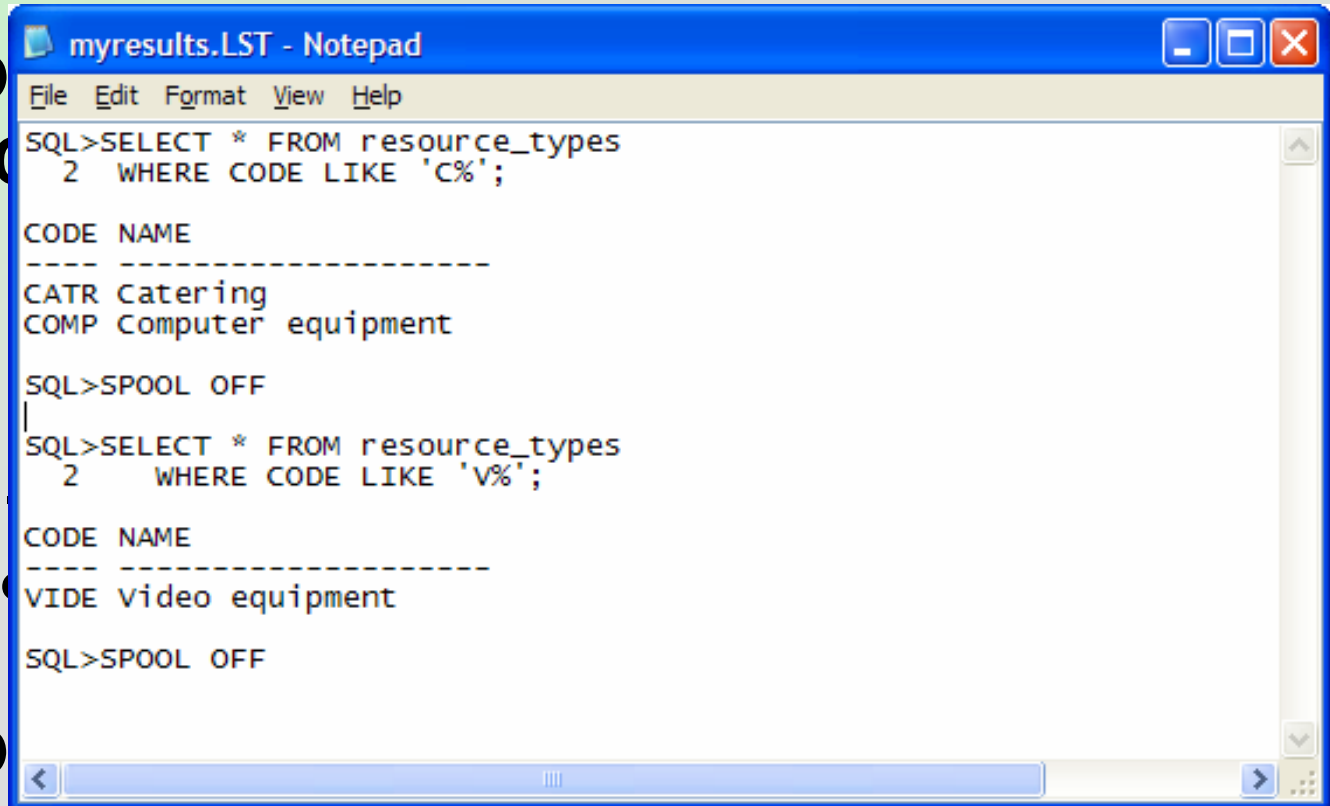
CODE NAME

-----

VIDE Vide

SQL>SPOOL

SQL>ed myresults.lst



```

myresults.LST - Notepad
File Edit Format View Help
SQL>SELECT * FROM resource_types
  2  WHERE CODE LIKE 'C%';

CODE NAME
-----
CATR Catering
COMP Computer equipment

SQL>SPOOL OFF
|
SQL>SELECT * FROM resource_types
  2  WHERE CODE LIKE 'V%';

CODE NAME
-----
VIDE Video equipment

SQL>SPOOL OFF
  
```



# DUAL Table 9i

```
ORA92>SELECT booking_seq.nextval FROM dual;
```

```
      NEXTVAL
```

```
-----  
      11283
```

## Execution Plan

```
-----  
   0          SELECT STATEMENT Optimizer=ALL_ROWS (Cost=2 Card=1)  
   1   0      SEQUENCE OF 'BOOKING_SEQ'  
   2   1      TABLE ACCESS (FULL) OF 'DUAL' (Cost=2 Card=1)
```

## Statistics

```
-----  
      0  recursive calls  
      0  db block gets  
      3  consistent gets  
      0  physical reads  
      0  redo size  
    380  bytes sent via SQL*Net to client  
    499  bytes received via SQL*Net from client  
      2  SQL*Net roundtrips to/from client  
      0  sorts (memory)  
      0  sorts (disk)  
      1  rows processed
```



# DUAL Table 10g

```
SQL>SELECT booking_seq.nextval FROM dual;
```

```
      NEXTVAL
```

```
-----  
      11282
```

## Execution Plan

```
-----  
   0          SELECT STATEMENT Optimizer=ALL_ROWS (Cost=2 Card=1)  
   1   0      SEQUENCE OF 'BOOKING_SEQ' (SEQUENCE)  
   2   1          FAST DUAL (Cost=2 Card=1)
```

## Statistics

```
-----  
      0  recursive calls  
      0  db block gets  
      0  consistent gets  
      0  physical reads  
      0  redo size  
   394  bytes sent via SQL*Net to client  
   508  bytes received via SQL*Net from client  
      2  SQL*Net roundtrips to/from client  
      0  sorts (memory)  
      0  sorts (disk)  
      1  rows processed
```

# Recycle Bin



- DROP TABLE TEST;
- SELECT \* FROM RECYCLEBIN;

➤ FLASH

RENAME

Flash

➤ DESCRIBE

Oracle SQL\*Plus

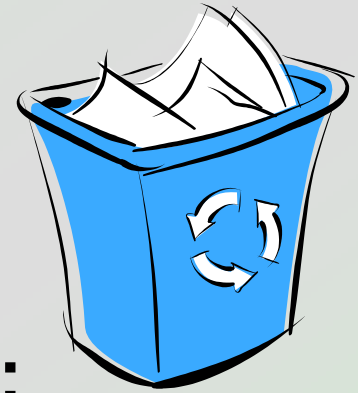
OBJECT_NAME	ORIGINAL_NAME	OPERATION
TYPE	TS_NAME	CREATETIME
DROPTIME	DROPSCN	PARTITION_NAME
RELATED	BASE_OBJECT	PURGE_OBJECT
SPACE		
BIN\$0mpnahyRSah11C2vkywLA==\$0	TEST	DROP
TABLE	USERS	2004-08-19:00:10:02
2004-08-19:00:11:24	500090	YES YES
51645	51645	51645
		8

Name

COL1

NUMBER





- Oracle SQL\*Plus

File Edit Search Options Help

show user

OBJECT_NAME	ORIGINAL_NAME	OPERATION
TYPE	TS_NAME	CREATETIME
DROPTIME	DROPSCN	PARTITION_NAME
RELATED	BASE_OBJECT	PURGE_OBJECT
SPACE		
BIN\$naHDW2izQx6Mbvn1/z/USQ==\$0	TEST_IX	DROP
INDEX	USERS	2004-09-17:12:10:50
2004-09-17:12:10:50	956548	NO YES
52375	52375	52376
8		
BIN\$u6hEnuI3Rp6sWY1/LAR03Q==\$0	TEST	DROP
TABLE	USERS	2004-09-17:12:10:50
2004-09-17:12:10:50	956549	YES YES
52375	52375	52375
8		

BIN\$na

The screenshot shows the Oracle SQL\*Plus interface. The command prompt displays two DROP statements followed by their execution results.

```

SQL> DROP INDEX BIN$naHDW2izQx6Mbvn1/z/USQ==$0 TEST_IX
      2004-09-17:12:10:50          956548          NO YES
            52375             52375           52376              8

SQL> DROP TABLE BIN$u6hEnuI3Rp6sWY1/LAR03Q==$0 TEST
      2004-09-17:12:10:50          956549          YES YES
            52375             52375           52375              8

```

# Recycle Bin



- **DROP TABLE TEST; -- send to recycle bin**
- **DROP TABLE TEST PURGE; -- drop completely**
- **PURGE TABLE test; -- remove from recycle bin**
- **PURGE TABLE**  
**BIN\$n0mpnahyRSah1ICZvkywLA==\$0;**
- **PURGE RECYCLEBIN; -- purge your objects**
- **PURGE DBA\_RECYCLEBIN; -- purge all objects**
- **Objects will be automatically purged if space is required**

# MERGE Statement

```
MERGE INTO employees e
  USING (SELECT id, name, status, salary
        FROM employee_update) n
  ON (e.id = n.id)
  WHEN MATCHED THEN UPDATE
    SET e.name = n.name, e.status = n.status,
    e.salary = n.salary
  DELETE WHERE (e.status = 'D')
  WHEN NOT MATCHED THEN INSERT
    (id, name, status, salary)
  VALUES (n.id, n.name, n.status, n.salary)
  WHERE n.status != 'P';
```

# MERGE Statement

```
SQL> SELECT * FROM employee
```

ID	NAME	S	SALARY
1	SMITH	C	50000
12	SMITH	D	50000
2	HARRIS	C	50000

← Updated and Deleted  
← Not Deleted

```
SQL> SELECT * FROM employee_update
```

ID	NAME	S	SALARY
1	JONES	D	50000
3	COOKSON	C	50000
4	MARSHALL	P	50000
2	HARRIS	C	60000

← Inserted  
← Not Inserted

```
SQL> @t1
```

```
3 rows merged.
```

```
SQL> SELECT * FROM employee;
```

ID	NAME	S	SALARY
12	SMITH	D	50000
2	HARRIS	C	60000
3	COOKSON	C	50000

# CONNECT\_BY\_ISLEAF

- Returns 1 for a leaf row
- Returns 0 for a row which has child rows
- Example

```
SELECT org_id, name, level, sys_connect_by_path(org_id,'/') path,
CONNECT_BY_ISLEAF Leaf
FROM organisations
CONNECT BY PRIOR org_id = parent_org_id
START WITH parent_org_id IS NULL
```

ORG_ID	NAME	LEVEL	PATH	LEAF
1000	Sage Computing Services	1	/1000	0
1010	Consulting Services	2	/1000/1010	0
1030	Database Administration	3	/1000/1010/1030	1
1040	Systems Development	3	/1000/1010/1040	1
2264	Australian Medical Systems	1	/2264	1
3210	Conservation Society	1	/3210	0
3214	Forests Division	2	/3210/3214	1
3216	Rivers Division	2	/3210/3216	1
3842	Institute of Business Services	1	/3842	0
3843	Marketing Services	2	/3842/3843	1
3844	Financial Services	2	/3842/3844	1



# CONNECT\_BY\_ISCYCLE

## ➤ CONNECT\_BY\_ISCYCLE

Returns 1 if the row is part of a circular reference

Returns 0 if row links are not circular

## ➤ Example

```
UPDATE organisations set parent_org_id = 1010 WHERE org_id = 1000;
```

```
SELECT org_id, name, level, sys_connect_by_path(org_id,'/') path,  
CONNECT_BY_ISCYCLE Cycle  
FROM organisations  
WHERE org_id IN (1000,1010)  
CONNECT BY NOCYCLE PRIOR org_id = parent_org_id
```

ORG_ID	NAME	LEVEL	PATH	CYCLE
1010	Consulting Services	1	/1010	0
1000	Sage Computing Services	2	/1010/1000	1
1000	Sage Computing Services	1	/1000	0
1010	Consulting Services	2	/1000/1010	1

# CONNECT\_BY\_ROOT

➤ Returns the value of the expression in the root row

➤ Example:

```
SELECT org_id, name, level, sys_connect_by_path(org_id, '/') path,
       CONNECT_BY_ROOT org_id parent
FROM   organisations
WHERE  level > 1
CONNECT BY PRIOR org_id = parent_org_id;
```

ORG_ID	NAME	LEVEL	PATH	PARENT
1030	Database Administration	2	/1010/1030	1010
1040	Systems Development	2	/1010/1040	1010
1010	Consulting Services	2	/1000/1010	1000
1030	Database Administration	3	/1000/1010/1030	1000
1040	Systems Development	3	/1000/1010/1040	1000
3843	Marketing Services	2	/3842/3843	3842
3844	Financial Services	2	/3842/3844	3842
3214	Forests Division	2	/3210/3214	3210
3216	Rivers Division	2	/3210/3216	3210

# CONNECT\_BY\_ROOT



## Example:

Find the number of organisations that ultimately report to each organisation

```
SELECT parent, count(cnt)
FROM
    (SELECT CONNECT_BY_ROOT org_id parent, 1 cnt
     FROM organisations
     WHERE level >1
     CONNECT BY PRIOR org_id = parent_org_id)
GROUP BY parent
```

PARENT	COUNT ( CNT )
1000	3
1010	2
3210	2
3842	2





# Floating Point Numbers

- **BINARY\_FLOAT**
  - 5 bytes including length byte
- **BINARY\_DOUBLE**
  - 9 bytes including length bytes
- **Faster Arithmetic**
- **Binary Precision**
- **Some IEEE754 Conformance**
- **1.2e-308 to 3.4e308**

# Floating Point Numbers

```
DECLARE
```

```
  v_1 NUMBER := 10;
```

```
  v_2 BINARY_FLOAT := 10;
```

```
  v_3 BINARY_DOUBLE := 10;
```

```
BEGIN
```

```
  FOR count in 1..100 LOOP
```

```
    v_1 := v_1 * v_1;
```

```
    v_2 := v_2 * v_2;
```

```
    v_3 := v_3 * v_3;
```

```
    dbms_output.put_line('number is '||v_1);
```

```
    dbms_output.put_line('binary float is '||v_2);
```

```
    dbms_output.put_line('binary double is '||v_3);
```

```
  END LOOP;
```

```
END;
```

```
/
```

# Floating Point Numbers

```
number is 100
binary float is 1.0E+002
binary double is 1.0E+002
```

```
number is 10000
binary float is 1.0E+004
binary double is 1.0E+004
```

```
number is 100000000
binary float is 1.0E+08
binary double is 1.0E+08
```

```
number is 10000000000000000
binary float is 1.00000003E+016
binary double is 1.0E+016
```

```
number is 10000000000000000000000000000000  
binary float is 1.00000003E+032  
binary double is 1.0000000000000001E+032
```

```
number is 1000000000000000000000000000000000000000000000000000000000000000  
binary float is Inf  
binary double is 1.0000000000000002E+064
```

```
number is ~
binary float is Inf
binary double is 1.0000000000000003E+128
```

```
number is ~
binary float is Inf
binary double is 1.0000000000000005E+256
```

number is ~  
binary float is Inf  
binary double is Inf

# NAN and INFINITY

```
CREATE TABLE testnm (col1 binary_float);  
INSERT INTO testnm (col1) VALUES ('INFINITY');  
INSERT INTO testnm (col1) VALUES ('NAN');  
SELECT * FROM testnm;
```

COL1
-----
Nan
Inf

```
SELECT * FROM testnm WHERE col1 IS NAN;  
SELECT * FROM testnm WHERE col1 IS INFINITE;  
SELECT NANVL(col1,null) FROM testnm;
```



# Version 9i Flashback

```
SELECT * FROM resources WHERE code = 'VCR1';
```

CODE	TYPE	DESCRIPTION	DAILY_RATE	E
VCR1	VIDE	Video recorder No 02378	100	

```
ORA92>SELECT systimestamp FROM dual;
```

```
SYSTIMESTAMP
```

```
28/AUG/04 10:04:17.515000 PM +08:00
```

```
ORA92>UPDATE RESOURCES SET daily_rate = 900 WHERE code = 'VCR1';
```

```
1 row updated.
```

```
ORA92>commit;
```

```
Commit complete.
```

```
ORA92>SELECT * FROM resources AS OF TIMESTAMP
```

```
TO_TIMESTAMP('28/AUG/2004 22:04:17','dd/mon/yyyy hh24:mi:ss');
```

```
WHERE code = 'VCR1';
```

CODE	TYPE	DESCRIPTION	DAILY_RATE	E
VCR1	VIDE	Video recorder No 02378	100	



# Version 10G Flashback

```
SELECT * FROM resources WHERE code = 'VCR1';
```

CODE	TYPE	DESCRIPTION	DAILY_RATE	E
VCR1	VIDE	Video recorder No 02378	100	

```
ORA10G>SELECT systimestamp FROM dual;
```

SYSTIMESTAMP
28-AUG-04 10.15.14.437000 PM +08:00

```
ORA10G>UPDATE RESOURCES SET daily_rate = 900 WHERE code = 'VCR1';
```

```
1 row updated.
```

```
ORA10G>commit;
```

```
Commit complete.
```

```
ORA10G>UPDATE RESOURCES SET daily_rate = 1000 WHERE code = 'VCR1';
```

```
1 row updated.
```

```
ORA10G>commit;
```

```
Commit complete.
```



# Version 10G Flashback

```
SELECT * FROM resources  
VERSIONS BETWEEN TIMESTAMP  
TO_TIMESTAMP('28/aug/2004 22:15:14', 'dd/mon/yyyy hh24:mi:ss')  
AND MAXVALUE  
WHERE code = 'VCR1';
```

CODE	TYPE	DESCRIPTION	DAILY_RATE	E
VCR1	VIDE	Video recorder No 02378	1000	
VCR1	VIDE	Video recorder No 02378	900	
VCR1	VIDE	Video recorder No 02378	100	



# Version 10G Flashback

```
SELECT code, daily_rate,  
       VERSIONS_STARTTIME, VERSIONS_STARTSCN,  
       VERSIONS_ENDTIME, VERSIONS_ENDSCN,  
       VERSIONS_XID, VERSIONS_OPERATION  
FROM resources  
VERSIONS BETWEEN TIMESTAMP  
TO_TIMESTAMP('28/aug/2004 22:15:14', 'dd/mon/yyyy hh24:mi:ss')  
AND MAXVALUE  
WHERE code = 'VCR1';
```

CODE	DAILY_RATE	VERSIONS_STARTTIME	VERSIONS_STARTSCN	VERSIONS_ENDTIME	VERSIONS_ENDSCN	VERSIONS_XID	V
VCR1	1000	28-AUG-04 10.22.00 PM	599525			04002400D8000000	U
VCR1	900	28-AUG-04 10.15.32 PM	599354	28-AUG-04 10.22.00 PM	599525	04002000D8000000	U
VCR1	100			28-AUG-04 10.15.32 PM	599354		





# Version 10G Flashback

```
SELECT undo_sql  
FROM FLASHBACK_TRANSACTION_QUERY  
WHERE XID = '04002000D8000000'
```

UNDO\_SQL

-----  
update "TRAIN"."RESOURCES" set "DAILY\_RATE" = '1000' where ROWID = 'AAAMnYAAEAAA  
AH+AAA';



# Regular Expressions

- **Advanced pattern matching**
- **Character literals and metacharacters**
- **Supports POSIX extended Regular Expression Syntax**
- **REGEXP\_LIKE**
- **REGEXP\_INSTR**
- **REGEXP\_REPLACE**
- **REGEXP\_SUBSTR**



# Regular Expressions – Example(1)

```
SELECT * FROM testexp;
```

```
COL1
```

```
-----
```

```
colour
```

```
color
```

```
grey
```

```
gray
```

```
the grey colour
```

```
SELECT * FROM testexp  
WHERE REGEXP_LIKE(col1,'gr(e|a)y');
```

```
COL1
```

```
-----
```

```
grey
```

```
gray
```

```
the grey colour
```

# Regular Expressions – Example(2)

```
SELECT * FROM testexp;
```

```
COL1
```

```
-----
```

```
colour
```

```
color
```

```
grey
```

```
gray
```

```
the grey colour
```

```
SELECT * FROM testexp
```

```
WHERE REGEXP_LIKE(col1,'^gr(e|a)y$')
```

```
COL1
```

```
-----
```

```
grey
```

```
gray
```



# Regular Expressions – Example(3)

```
SELECT postcode FROM address;
```

```
POST
```

```
----
```

```
6020
```

```
2030
```

```
203a
```

```
abcd
```

```
-- Find strings which contain a character that is not a numeric digit
```

```
SELECT postcode
```

```
FROM address
```

```
WHERE REGEXP_LIKE(postcode, '^[[:digit:]]');
```

```
POST
```

```
----
```

```
203a
```

```
abcd
```

# Regular Expressions – Functions

-- Find the position of the first character that is not a numeric digit

```
SELECT postcode, REGEXP_INSTR(postcode, '^[[:digit:]]') pos
FROM address
WHERE REGEXP_LIKE(postcode, '^[[:digit:]]');
```

POST	POS
-----	-----
203a	4
abcd	1

-- Replace Grey or Gray with Green

```
SELECT col1, REGEXP_REPLACE(col1, 'gr(e|a)y','green') newcol
FROM testexp
WHERE REGEXP_LIKE(col1,'gr(e|a)y');
```

COL1	NEWCOL
-----	-----
grey	green
gray	green
the grey colour	the green colour

# Row SCN

```
SELECT code, daily_rate, ORA_ROWSCN,
       SCN_TO_TIMESTAMP(ORA_ROWSCN)
FROM   resources;
```

CODE	DAILY_RATE	ORA_ROWSCN	SCN_TO_TIMESTAMP(ORA_ROWSCN)			
----	-----	-----	-----	-----	-----	-----
VCR1	1000	599525	28-AUG-04	10.22.00.0000000000	PM	
VCR2	120	599525	28-AUG-04	10.22.00.0000000000	PM	
BRLG		599525	28-AUG-04	10.22.00.0000000000	PM	
BRSM		599525	28-AUG-04	10.22.00.0000000000	PM	
CONF	1000	599525	28-AUG-04	10.22.00.0000000000	PM	
CHRS		599525	28-AUG-04	10.22.00.0000000000	PM	
FLPC		599525	28-AUG-04	10.22.00.0000000000	PM	
FILE	8	599525	28-AUG-04	10.22.00.0000000000	PM	
OHP1	100	599525	28-AUG-04	10.22.00.0000000000	PM	
DATS	400	599525	28-AUG-04	10.22.00.0000000000	PM	
TAP1	90	599525	28-AUG-04	10.22.00.0000000000	PM	
LNCH	21	599525	28-AUG-04	10.22.00.0000000000	PM	
PC1	140	599525	28-AUG-04	10.22.00.0000000000	PM	
LT1	120	599525	28-AUG-04	10.22.00.0000000000	PM	

➤ This is actually the SCN of the last row changed in the block



# Row SCN

- Store Row level SCN information
- Add 6 bytes to each row
- CREATE TABLE resources\_new ROWDEPENDENCIES  
AS SELECT \* FROM resources;

```
SELECT code, daily_rate, ORA_ROWSCN,  
       SCN_TO_TIMESTAMP(ORA_ROWSCN)  
FROM   resources_new;
```

CODE	DAILY_RATE	ORA_ROWSCN	SCN_TO_TIMESTAMP(ORA_ROWSCN)
VCR1	1500	601977	28-AUG-04 11.44.36.000000000 PM
VCR2	180	601983	28-AUG-04 11.44.48.000000000 PM
BRLG		602012	28-AUG-04 11.45.10.000000000 PM
BRSM		601941	28-AUG-04 11.43.53.000000000 PM
CONF	1000	601941	28-AUG-04 11.43.53.000000000 PM
CHRS		601941	28-AUG-04 11.43.53.000000000 PM
FLPC		601941	28-AUG-04 11.43.53.000000000 PM
FILE	8	601941	28-AUG-04 11.43.53.000000000 PM
OHP1	100	601941	28-AUG-04 11.43.53.000000000 PM
DATS	400	601941	28-AUG-04 11.43.53.000000000 PM
TAP1	90	601941	28-AUG-04 11.43.53.000000000 PM
LNCH	21	601941	28-AUG-04 11.43.53.000000000 PM
PC1	140	601941	28-AUG-04 11.43.53.000000000 PM
LT1	120	601941	28-AUG-04 11.43.53.000000000 PM





# The Model Clause

- **Use multiple rows as input to calculations**
- **Perform projections**
- **Estimate missing data according to rules**
- **Calculations on array**
- **Similar to spreadsheet processing**



# The Model Clause – An Example

-- Create a table to record storage used by tables in a database

```
CREATE TABLE table_storage  
(database_name      VARCHAR2(10),  
table_name          VARCHAR2(30),  
storage_year        NUMBER(4),  
storage_used        NUMBER,  
CONSTRAINT stor_pk PRIMARY KEY  
(database_name, table_name, storage_year));
```

# The Model Clause – An Example

- Populate the table**
- Notice values for 1999 are missing**

```
insert into table_storage values ('PROD','ORGANISATIONS',1998,10);  
insert into table_storage values ('PROD','ORGANISATIONS',2000,40);  
insert into table_storage values ('PROD','ORGANISATIONS',2001,50);  
insert into table_storage values ('PROD','ORGANISATIONS',2002,60);
```

```
insert into table_storage values ('DEV','ORGANISATIONS',1998,5);  
insert into table_storage values ('DEV','ORGANISATIONS',2000,10);  
insert into table_storage values ('DEV','ORGANISATIONS',2001,15);  
insert into table_storage values ('DEV','ORGANISATIONS',2002,20);
```

```
insert into table_storage values ('PROD','EVENTS',1998,100);  
insert into table_storage values ('PROD','EVENTS',2000,300);  
insert into table_storage values ('PROD','EVENTS',2001,400);  
insert into table_storage values ('PROD','EVENTS',2002,500);
```

```
insert into table_storage values ('DEV','EVENTS',1998,50);  
insert into table_storage values ('DEV','EVENTS',2000,100);  
insert into table_storage values ('DEV','EVENTS',2001,150);  
insert into table_storage values ('DEV','EVENTS',2002,200);
```



# The Model Clause – An Example

- Decide on the Partitions, Dimensions and Measures for the MODEL clause
- Think of this as a multidimensional array with the measure in the cell
- Partitions are used for independent data and allow parallelisation

## MODEL

**PARTITION BY (database\_name db)**

**DIMENSION BY (table\_name tn,  
storage\_year yr)**

**MEASURES (s.storage\_used st)**

# The Model Clause – An Example

- Add rules to project future data
- The storage for 2003 will increase by the same proportion as it did in 2002

```
SELECT db, tn, yr, st
FROM table_storage s
MODEL
  PARTITION BY (database_name db)
  DIMENSION BY (table_name tn, storage_year yr)
  MEASURES (s.storage_used st)
  RULES (
    st['ORGANISATIONS',2003] = ROUND((((st['ORGANISATIONS',2002]-
    st['ORGANISATIONS',2001])
    /st['ORGANISATIONS',2001]) * st['ORGANISATIONS',2002]
    + st['ORGANISATIONS',2002]),
    st['EVENTS',2003] = ROUND((((st['EVENTS',2002]-st['EVENTS',2001])
    /st['EVENTS',2001]) * st['EVENTS',2002]
    + st['EVENTS',2002]))
  )
ORDER BY db, tn, yr;
```

# The Model Clause – An Example

DB	TN	YR	ST
DEV	EVENTS	1998	50
DEV	EVENTS	2000	100
DEV	EVENTS	2001	150
DEV	EVENTS	2002	200
DEV	EVENTS	2003	267
DEV	ORGANISATIONS	1998	5
DEV	ORGANISATIONS	2000	10
DEV	ORGANISATIONS	2001	15
DEV	ORGANISATIONS	2002	20
DEV	ORGANISATIONS	2003	27
PROD	EVENTS	1998	100
PROD	EVENTS	2000	300
PROD	EVENTS	2001	400
PROD	EVENTS	2002	500
PROD	EVENTS	2003	625
PROD	ORGANISATIONS	1998	10
PROD	ORGANISATIONS	2000	40
PROD	ORGANISATIONS	2001	50
PROD	ORGANISATIONS	2002	60
PROD	ORGANISATIONS	2003	72

# The Model Clause – An Example

- Add rules to fill in missing data
- The missing data will be the average of the year below and above

```
SELECT db, tn, yr, st, nst
FROM table_storage s
MODEL
  PARTITION BY (database_name db)
  DIMENSION BY (table_name tn, storage_year yr)
  MEASURES (s.storage_used st, CAST(NULL AS NUMBER) nst)
RULES (
  -- generate figures for years that have none
  nst[FOR tn IN (SELECT table_name FROM table_storage),
    FOR yr FROM 1998 TO 2002 INCREMENT 1]
    = CASE WHEN st[cv(), cv()] IS PRESENT
      THEN st[cv(), cv()]
      ELSE ROUND(AVG(st)[cv(),yr BETWEEN cv()-1 and cv() +1])
    END
)
ORDER BY db, tn, yr;
```

# The Model Clause – An Example

DB	TN	YR	ST	NST
DEV	EVENTS	1998	50	50
DEV	EVENTS	1999		75
DEV	EVENTS	2000	100	100
DEV	EVENTS	2001	150	150
DEV	EVENTS	2002	200	200
DEV	ORGANISATIONS	1998	5	5
DEV	ORGANISATIONS	1999		8
DEV	ORGANISATIONS	2000	10	10
DEV	ORGANISATIONS	2001	15	15
DEV	ORGANISATIONS	2002	20	20
PROD	EVENTS	1998	100	100
PROD	EVENTS	1999		200
PROD	EVENTS	2000	300	300
PROD	EVENTS	2001	400	400
PROD	EVENTS	2002	500	500
PROD	ORGANISATIONS	1998	10	10
PROD	ORGANISATIONS	1999		25
PROD	ORGANISATIONS	2000	40	40
PROD	ORGANISATIONS	2001	50	50
PROD	ORGANISATIONS	2002	60	60





# Commit Processing

- **COMMIT WRITE WAIT**
- **COMMIT WRITE NOWAIT**
- **COMMIT WRITE BATCH**
- **ALTER SYSTEM SET COMMIT\_WRITE NOWAIT;**
- **ALTER SESSION SET COMMIT\_WRITE NOWAIT;**



# DML Error Handling - Example

**Create an error log table for BOOKINGS**

**BEGIN**

**dbms\_errlog.create\_error\_log('BOOKINGS');**

**END;**

**Insert using error logging**

**INSERT INTO bookings**

**SELECT \* FROM bookings\_new**

**LOG ERRORS INTO err\$\_bookings**

**REJECT LIMIT 50;**



# DML Error Handling - Example

List the rows that had errors

```
SELECT  ora_err_mesg$, booking_no,  
        resource_code, cost, quantity  
FROM    err$_bookings;
```

ORA_ERR_MESG\$	BOOKI	RESO	COST	QUANT
ORA-00001: unique constraint (TRAIN.BOOK_PK) violated	1310	TAP1	0	1
ORA-00001: unique constraint (TRAIN.BOOK_PK) violated	1311	BRSM	0	1
ORA-00001: unique constraint (TRAIN.BOOK_PK) violated	1312	CONF	3630	1
ORA-00001: unique constraint (TRAIN.BOOK_PK) violated	1313	VCR2	435.6	1
ORA-00001: unique constraint (TRAIN.BOOK_PK) violated	1214	LNCH	889.3	35



# Aggregate Functions

- **MEDIAN**
- **STATS\_MODE**
- **STATS\_ONE\_WAY\_ANOVA**
- **STATS\_T\_TEST\_ONE**
- **STATS\_T\_TEST\_PAIRRED**
- **STATS\_BINOMIAL\_TEST**
- **STATS\_WSR\_TEST**
- **STATS\_MW\_TEST**

# PL/SQL Warnings

➤ **Produce compile time warnings**

➤ **Set at:**

➤ **Instance level**

**PLSQL\_WARNINGS='ENABLE:SEVERE';**

➤ **System level**

**ALTER SYSTEM SET PLSQL\_WARNINGS='ENABLE:ALL';**

➤ **Session level**

**ALTER SESSION SET PLSQL\_WARNINGS='DISABLE:ALL';**

**ALTER SESSION SET PLSQL\_WARNINGS='ENABLE:SEVERE',  
'DISABLE:PERFORMANCE','ERROR:06002';**

➤ **Program unit**

**ALTER PROCEDURE proc1 COMPILE PLSQL\_WARNINGS=  
'ENABLE:PERFORMANCE';**



# Enable PL/SQL Warnings

```
SQL> SELECT dbms_warning.get_warning_setting_string()  
FROM dual;
```

```
DBMS_WARNING.GET_WARNING_SETTING_STRING( )
```

```
-----  
DISABLE:ALL
```

```
SQL> CALL DBMS_WARNING.SET_WARNING_SETTING_STRING  
('ENABLE:ALL' , 'SESSION');
```

```
SQL> SELECT dbms_warning.get_warning_setting_string()  
FROM dual;
```

```
DBMS_WARNING.GET_WARNING_SETTING_STRING( )
```

```
-----  
ENABLE:ALL
```



# Test PL/SQL Warning

```
CREATE OR REPLACE PROCEDURE proc2 AS
  v_postcode NUMBER;
  CURSOR c_org IS
  SELECT * FROM organisations
  WHERE postcode = v_postcode;
  BEGIN
    v_postcode := 6020; -- postcode column is VARCHAR2(4)
    FOR r_org IN c_org LOOP
      null;
    END LOOP;
  END;
```

**SP2-0804: Procedure created with compilation warnings**



# Test PL/SQL Warning

**SQL> show errors**

**Errors for PROCEDURE PROC2:**

**LINE/COL ERROR**

-----  
**4/1 PLW-07204: Message 7204 not found; No message file for  
product=plsql, facility=PLW**

➤ **Message File plwus.msb not included in installation media**

➤ **Use Database Error Messages documentation**

**PLW-07204: conversion away from column type may result in sub-optimal query plan**

**Cause:** The column type and the bind type do not exactly match. This may result in the column being converted to the type of the bind variable. This type conversion may prevent the SQL optimizer from using any index the column participates in. This may adversely affect the execution performance of this statement.

**Action:** To make use of any index for this column, make sure the bind type is the same type as the column type.





# DBMS\_OUTPUT

- **Default buffer size UNLIMITED**
- **Line buffer up to 32767**

**SQL> SET SERVEROUTPUT ON**

**SQL> SHOW SERVEROUTPUT**

**serveroutput ON SIZE UNLIMITED  
FORMAT WORD\_WRAPPED**

# Nested Table Operations

Operation	Description
<b>MULTISET UNION</b>	The equivalent of a UNION ALL operation. Rows that are in either set are combined. Duplicates are not removed.
<b>MULTISET UNION DISTINCT</b>	The equivalent of a UNION operation. Rows that are in either set are combined. Duplicates are removed.
<b>MULTISET INTERSECT</b>	Returns rows that exist in both sets. Duplicates are not removed.
<b>MULTISET INTERSECT DISTINCT</b>	Returns rows that exist in both sets. Duplicates are removed.
<b>MULTISET EXCEPT</b>	Returns rows that exist in the first set but not the second. Duplicates are not removed.
<b>MULTISET EXCEPT DISTINCT</b>	Returns rows that exist in the first set but not the second. Duplicates are removed.

# Nested Table Operations

```

DECLARE
  TYPE table_type IS TABLE OF VARCHAR2(1);
  t1 table_type;
  t2 table_type;
  t3 table_type;
  t4 table_type;
  t5 table_type;
  t6 table_type;
  PROCEDURE show_table (p_table IN table_type)
  IS BEGIN
    FOR v_counter IN p_table.first..p_table.last LOOP
      dbms_output.put (p_table(v_counter));
    END LOOP;
    dbms_output.new_line;
  END;
BEGIN
  t1 := table_type('A','B','C','D');
  t2 := table_type('C','D','E','F');
  t3 := t1 MULTiset UNION t2;
  dbms_output.put_line ('t3 is ');
  show_table(t3);
  t4 := t1 MULTiset UNION DISTINCT t2;
  dbms_output.put_line ('t4 is ');
  show_table(t4);
  t5 := t1 MULTiset INTERSECT t2;
  dbms_output.put_line ('t5 is ');
  show_table(t5);
  t6 := t1 MULTiset EXCEPT t2;
  dbms_output.put_line ('t6 is ');
  show_table(t6);
END;
/

```

```

t3 is
ABCD CDEF
t4 is
ABCDEF
t5 is
CD
t6 is
AB

```



# More Nested Table Operations

- **MEMBER OF**
- **SUBMULTISET OF**
- **CARDINALITY**
- **SET**
- **IS A SET**



# Conditional Compilation

- **Code included in the source code**
- **Compiled into the stored program unit conditionally**
- **Useful for debugging information**
- **Code to support multiple versions**
- **DBMS\_DB\_VERSION provides version and release constants**



# Conditional Compilation Directives

**\$IF boolean\_static\_expression \$THEN**

**PL/SQL code**

**\$ELSIF boolean\_static\_expression \$THEN**

**PL/SQL code**

**\$ELSE**

**PL/SQL code**

**\$END**

**\$ERROR varchar2\_static\_expression \$END**

# Conditional Compilation - Example

- **Set the value of a pre-processor variable**  
**ALTER SESSION**  
**SET plsql\_ccflags = 'DEBUGON:TRUE';**
- **Create a procedure containing conditionally compiled code**

```
CREATE OR REPLACE PROCEDURE proc1 AS
BEGIN
    -- some processing

    $IF $$DEBUGON $THEN
        INSERT INTO LOGTABLE (col1)
        VALUES ('some debugging information');
    $END
END;
```

# Conditional Compilation

- **Pre-processor variables can also be defined in the ALTER PROCEDURE command**  
**ALTER PROCEDURE proc1 COMPILE**  
**plsql\_ccflags = 'DEBUGON:FALSE'**  
**REUSE SETTINGS;**
- **Persistent packaged variables can also be used in conditional compilation**



# Improvements in Bulk Processing

- In version 9i sparse PL/SQL tables used in FORALL processing produced errors

```
CREATE TABLE TESTBULK (col1 number, col2 VARCHAR2(4));
```

```
CREATE OR REPLACE PROCEDURE ins_bulk AS  
  TYPE test_tab IS TABLE OF VARCHAR2(4) INDEX BY BINARY_INTEGER;  
  t_test test_tab;  
  v_sql VARCHAR2(200) := 'INSERT INTO TESTBULK (col1, col2)  
                        VALUES (test_seq.nextval,:1)';
```

```
BEGIN  
  t_test(1) := 'A';  t_test(2) := 'B';  t_test(3) := 'C';  
  t_test(4) := 'D';  t_test(5) := 'E';  
  t_test.delete(3);  
  FORALL v_counter IN 1..t_test.count SAVE EXCEPTIONS  
    EXECUTE IMMEDIATE v_sql USING t_test(v_counter);  
END;
```

```
SQL> exec ins_bulk
```

```
ERROR:
```

```
ORA-24381: error(s) in array DML
```

```
ORA-06512: at "TRAIN.INS_BULK", line 12
```

```
ORA-06512: at line 1
```

# Improvements in Bulk Processing

- In version 10i sparse PL/SQL tables can use the INDICES OF clause

```
CREATE OR REPLACE PROCEDURE ins_bulk AS
  TYPE test_tab IS TABLE OF VARCHAR2(4) INDEX BY BINARY_INTEGER;
  t_test test_tab;
  v_sql VARCHAR2(200) := 'INSERT INTO TESTBULK (col1, col2)
                        VALUES (test_seq.nextval,:1)';

BEGIN
  t_test(1) := 'A';  t_test(2) := 'B';  t_test(3) := 'C';
  t_test(4) := 'D';  t_test(5) := 'E';
  t_test.delete(3);
  FORALL v_counter IN INDICES OF t_test SAVE EXCEPTIONS
    EXECUTE IMMEDIATE v_sql USING t_test(v_counter);

END;
```

SQL> exec ins\_bulk

PL/SQL procedure successfully completed.

SQL> select \* from testbulk;

COL1	COL2
9	A
10	B
11	D
12	E

# Improvements in Bulk Processing

- In version 10i sparse PL/SQL tables can use the **VALUES OF** clause

```
CREATE OR REPLACE PROCEDURE ins
```

```
    TYPE test_tab IS TABLE OF VARCHAR2(200);
```

```
    TYPE test_ind IS TABLE OF PLS_INTEGER;
```

```
    t_test test_tab;
```

```
    t_ind test_ind := test_ind();
```

```
    v_sql VARCHAR2(200) := 'INSERT INTO testbulk  
VALUES
```

```
BEGIN
```

```
    t_test(1) := 'A'; t_test(2) := 'B'; t_test(3) := 'C';
```

```
    t_test(4) := 'D'; t_test(5) := 'E';
```

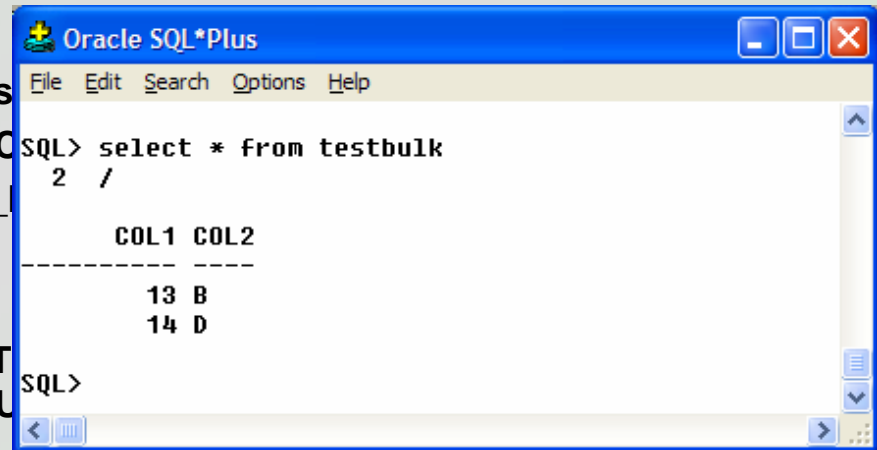
```
    t_ind.extend; t_ind(t_ind.last) := 2;
```

```
    t_ind.extend; t_ind(t_ind.last) := 4;
```

```
    FORALL v_counter IN VALUES OF t_ind SAVE EXCEPTIONS
```

```
    EXECUTE IMMEDIATE v_sql USING t_test(v_counter);
```

```
END;
```



Oracle SQL\*Plus

File Edit Search Options Help

```
SQL> select * from testbulk
2 /
```

COL1	COL2
13	B
14	D

SQL>



# Automatic Bulk Processing

```
FOR r_rec IN (SELECT.....) LOOP  
  .. Do some processing  
END LOOP
```

**Automatic read ahead of 100 rows**

# Fine Grained Auditing 9i

## ➤ Audit SELECT statements

```
-- Create an audit policy
-- for the BOOKINGS table
BEGIN
    dbms_fga.add_policy
        ( object_schema => 'TRAIN',
          object_name => 'BOOKINGS',
          policy_name => 'BOOK_AUD',
          audit_condition=>'cost > 500');
END;
```

# Fine Grained Auditing 9i

➤ The following statements would all be audited:

```
SELECT *  
FROM bookings;
```

```
SELECT cost  
FROM bookings;
```

```
SELECT *  
FROM bookings  
WHERE booking_no = 1207;
```

```
SELECT event_no  
FROM bookings  
WHERE booking_no = 1207;
```

```
SELECT *  
FROM bookings  
WHERE cost = 505;
```



# View the Audit Information 9i

```
SELECT db_user, timestamp, sql_text, sql_bind
FROM    dba_fga_audit_trail;
```

```
DB_USER                                TIMESTAMP
-----                                -
```

```
SQL_TEXT
-----
```

```
SQL_BIND
-----
```

```
TRAIN                                09/SEP/03
```

```
SELECT booking_no, cost
```

```
FROM bookings
```

```
WHERE resource_code = :v1
```

```
#1(4):CONF
```

# Fine Grained Auditing 10G

➤ Audit INSERT, UPDATE and DELETE statements

-- Create an audit policy

-- for the BOOKINGS table

**BEGIN**

dbms\_fga.add\_policy

(object\_schema => 'TRAIN',

object\_name => 'BOOKINGS',

policy\_name => 'BOOK\_AUD',

audit\_condition=>'cost > 500',

statement\_types => 'INSERT, UPDATE,  
DELETE, SELECT')

**END;**





# Fine Grained Auditing 10G

```
SELECT resource_code code, cost
FROM bookings
WHERE booking_no = 1207;
```

CODE	COST
LNCH	420

```
SELECT db_user, timestamp, sql_text, sql_bind
FROM dba_fga_audit_trail;
```

no rows selected



# Fine Grained Auditing 10G

```
UPDATE bookings SET cost = 501 where booking_no = 1207;  
1 row updated.
```

```
DELETE FROM bookings WHERE booking_no = 1207;  
1 row deleted.
```

```
SELECT db_user, timestamp, sql_text, sql_bind  
FROM    dba_fga_audit_trail;
```

```
DB_USER                                TIMESTAMP  
-----
```

```
SQL_TEXT  
-----
```

```
SQL_BIND  
-----
```

```
TRAIN                                17-SEP-04  
UPDATE bookings SET cost = 501 where booking_no = 1207
```

```
TRAIN                                17-SEP-04  
DELETE FROM bookings WHERE booking_no = 1207
```

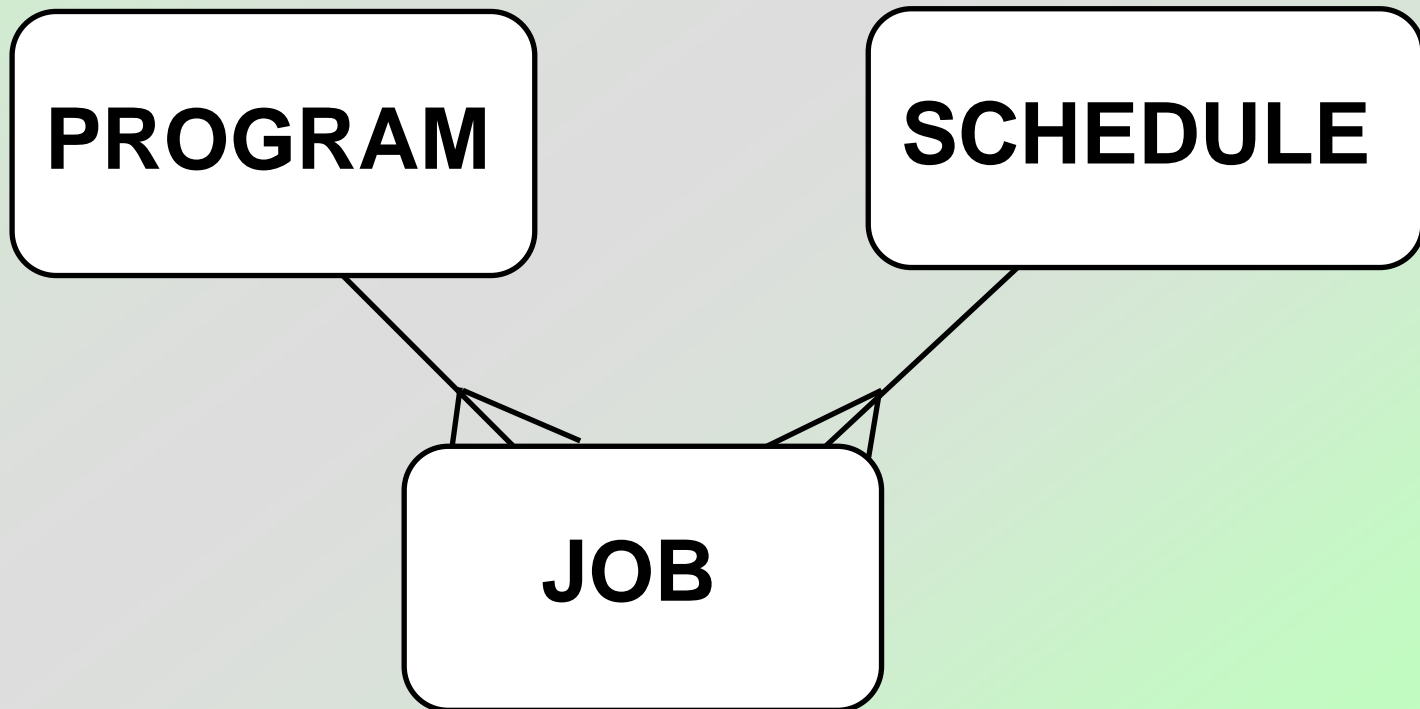


# Fine Grained Auditing 10G

- **Audit information uses autonomous transaction**
- **Audit information saved even if rollback**
- **AUDIT triggers fire per row**
- **FGA fires per statement.**
- **audit\_trail = DB\_EXTENDED stores bind variables in AUD\$**

# Scheduler

## ➤ New DBMS\_SCHEDULER package



# Scheduler

- Supports the execution of PL/SQL, C, Java and OS scripts
- Jobs can be scheduled at a specific time or at intervals
- Dynamic shrinking and growth of the pool of slave processes
- Allows prioritisation of jobs
- Allows chaining of jobs
- Monitoring views can be displayed from Enterprise Manager or using SQL
- Jobs can be scheduled in remote databases
- Jobs can be grouped into classes
- Event based scheduling



# Scheduler

```
dbms_scheduler.create_schedule (  
    schedule_name IN VARCHAR2, -- A unique name for the schedule  
    start_date      IN TIMESTAMP WITH TIMEZONE DEFAULT NULL,  
                    -- Start time of the job  
    repeat_interval IN VARCHAR2,  
                    -- Interval at which the job will be repeated  
    end_date        IN TIMESTAMP WITH TIMEZONE DEFAULT NULL,  
                    -- The date the job is no longer valid  
    comments        IN VARCHAR2 DEFAULT NULL) ;  
                    -- textual comments
```

# Scheduler

```
dbms_scheduler.create_program (  
    program_name => 'check_group_calls',  
    program_action => 'check_calls',  
    program_type  => 'stored_procedure');
```

```
dbms_scheduler.create_job (  
    job_name          =>    'group_calls_job1',  
    program_name      =>    'check_group_calls',  
    schedule_name     =>    'daily_midnight_job',  
    auto_drop         =>    FALSE);
```



# New Performance Features

- **Optimizer mode**
- **Statistics gathering**
- **New optimiser behaviour**
- **Trcsess utility**
- **New hints**
- **New tuning tools**
  - **SQL\*Profile**
  - **SQL\*Access**





# No Rule Based Optimiser

## ➤ OPTIMIZER\_MODE

### ➤ ALL\_ROWS

Throughput

### ➤ FIRST\_ROWS

Response

### ➤ FIRST\_ROWS\_N

Response + I hate full scans

# Statistics Gathering

- Number of INSERT, UPDATE, and DELETE operations recorded
- **SELECT \* FROM DBA\_TAB\_MODIFICATIONS**
- Automatic statistics gathered on STALE objects
  - Tables and indexes
  - Job Name = GATHER\_STATS\_JOB
  - Program =  
DBMS\_STATS.GATHER\_DATABASE\_STATS\_JOB\_PROC
  - Schedule = MAINTENANCE\_WINDOW\_GROUP
  - Nights and weekend
- Can be disabled using EXECUTE  
DBMS\_SCHEDULER.DISABLE  
( 'GATHER\_STATS\_JOB' );

# Statistics Management

- **Statistics history**
- **Stored by default for a period of 31 days**
- **If statistics\_level is TYPICAL or ALL they are purged automatically**
  - **DBA\_OPTSTAT\_OPERATIONS**
  - **USER\_TAB\_STATS\_HISTORY**
  - **DBMS\_STATS.ALTER\_STATS\_HISTORY\_RETENTION**
  - **DBMS\_STATS.PURGE\_STATS**
  - **DBMS\_STATS.RESTORE**
  - **DBMS\_STATS.GET\_STATS\_HISTORY\_AVAILABILITY**

# Statistics Management

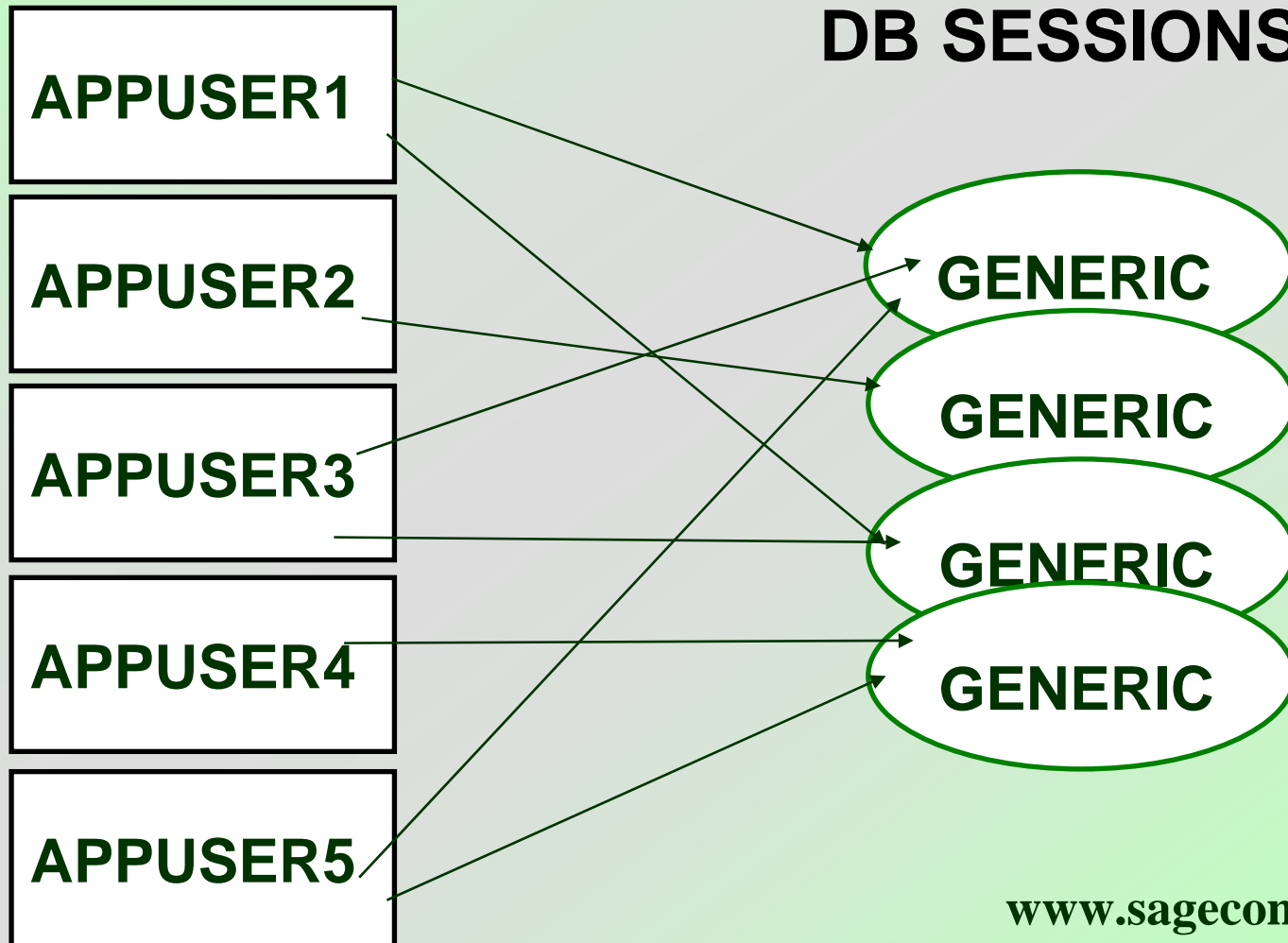
- **Ability to lock statistics**
  - **DBMS\_STATS.LOCK\_SCHEMA\_STATS**
  - **DBMS\_STATS.LOCK\_TABLE\_STATS**
- **Statistics can be unlocked**
  - **DBMS\_STATS.UNLOCK\_SCHEMA\_STATS**
  - **DBMS\_STATS.UNLOCK\_TABLE\_STATS**
- **Dynamic sampling**
  - **For very dynamic data or missing statistics**
  - **OPTIMIZER\_DYNAMIC\_SAMPLING**
  - **--+ DYNAMIC\_SAMPLING(tablealias level)**
  - **Levels 0 to 10**



# New Optimizer Behaviour

- **New costing model**
  - **CPU + IO**
  - **Uses MBRC system parameter for full scans**
  - **Statistics with or without workload**
- **Optimizer can run in tuning mode**
- **SQL profiles contain additional statistics**
- **SQL Access Advisor recommends indexes and materialised views**

# End to End Monitoring





# Oracle 10g Tuning - trcsess

- **trcsess used to consolidate a number of trace files**
- **Based on**
  - **Service**
  - **Clientid**
  - **Module**
  - **Action**
- **DBMS\_APPLICATION\_INFO package can be used to set the module and action from within an application**



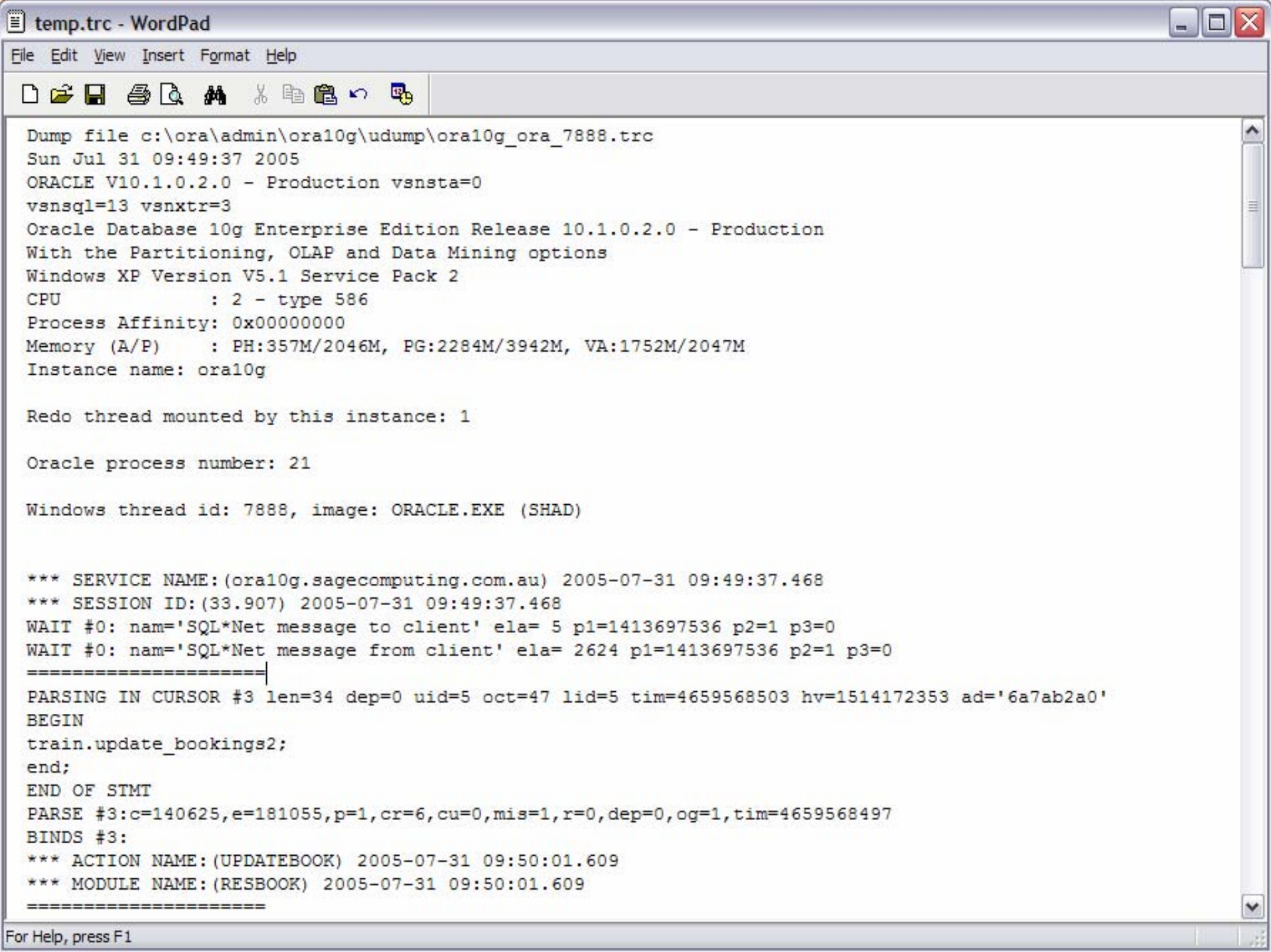
# Setting Module and Action

```
dbms_application_info.set_module  
(module_name=>'RESBOOK',  
action_name=>'UPDATEBOOK');
```

```
SELECT module, action  
FROM    v$session  
WHERE   module IS NOT NULL
```

MODULE	ACTION
-----	-----
SQL*Plus	
RESBOOK	UPDATEBOOK
SQL*Plus	





Dump file c:\ora\admin\ora10g\udump\ora10g\_ora\_7888.trc

Sun Jul 31 09:49:37 2005

ORACLE V10.1.0.2.0 - Production vsnsta=0

vsnsql=13 vsnxtr=3

Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production

With the Partitioning, OLAP and Data Mining options

Windows XP Version V5.1 Service Pack 2

CPU : 2 - type 586

Process Affinity: 0x00000000

Memory (A/P) : PH:357M/2046M, PG:2284M/3942M, VA:1752M/2047M

Instance name: ora10g

Redo thread mounted by this instance: 1

Oracle process number: 21

Windows thread id: 7888, image: ORACLE.EXE (SHAD)

\*\*\* SERVICE NAME:(ora10g.sagecomputing.com.au) 2005-07-31 09:49:37.468

\*\*\* SESSION ID:(33.907) 2005-07-31 09:49:37.468

WAIT #0: nam='SQL\*Net message to client' ela= 5 p1=1413697536 p2=1 p3=0

WAIT #0: nam='SQL\*Net message from client' ela= 2624 p1=1413697536 p2=1 p3=0

=====

PARSING IN CURSOR #3 len=34 dep=0 uid=5 oct=47 lid=5 tim=4659568503 hv=1514172353 ad='6a7ab2a0'

BEGIN

train.update\_bookings2;

end;

END OF STMT

PARSE #3:c=140625,e=181055,p=1,cr=6,cu=0,mis=1,r=0,dep=0,og=1,tim=4659568497

BINDS #3:

\*\*\* ACTION NAME:(UPDATEBOOK) 2005-07-31 09:50:01.609

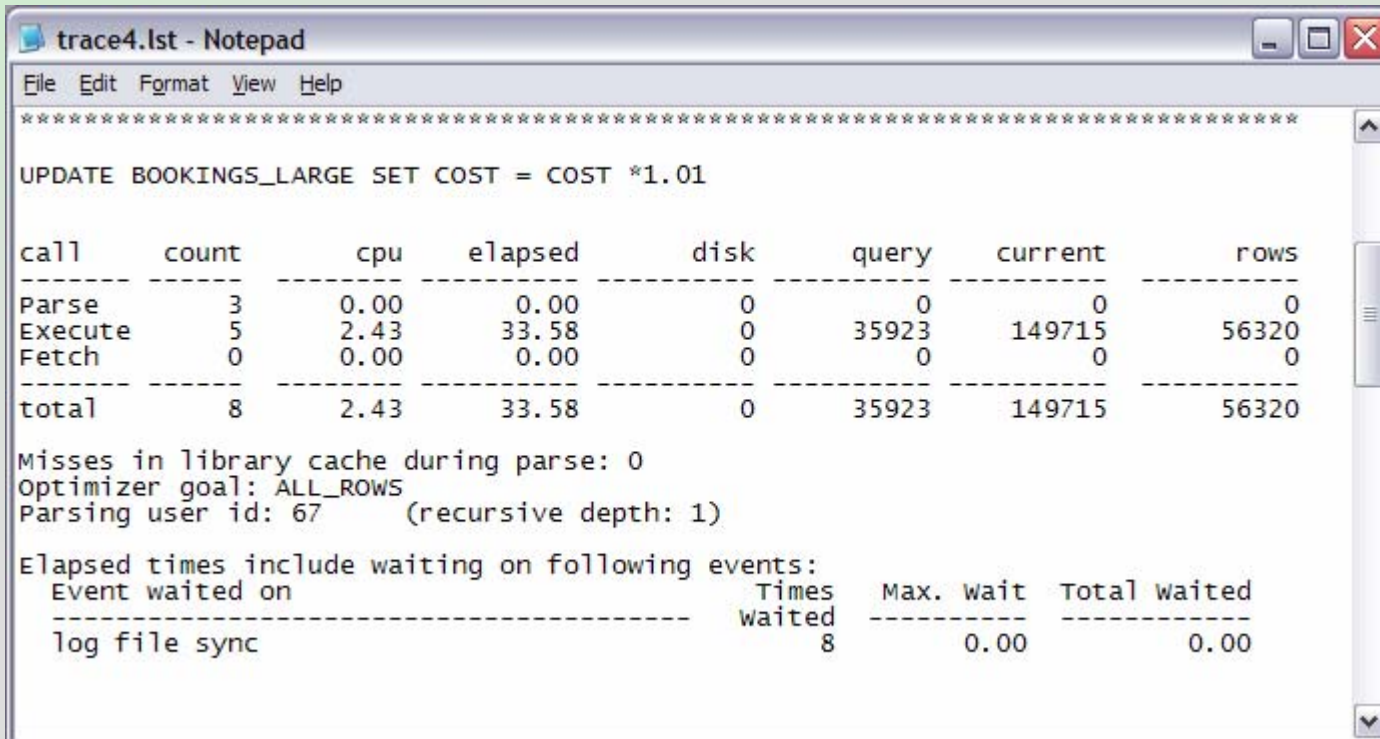
\*\*\* MODULE NAME:(RESBOOK) 2005-07-31 09:50:01.609

=====

# Consolidate Trace Files

```
trcssess output=trace4.trc module=RESBOOK
```

```
tkprof trace4.trc trace4.lst sys=no
```



```

*****
UPDATE BOOKINGS_LARGE SET COST = COST *1.01

call      count          cpu    elapsed        disk    query    current    rows
-----
Parse           3         0.00         0.00          0          0          0          0
Execute         5         2.43        33.58          0       35923      149715      56320
Fetch           0         0.00         0.00          0          0          0          0
total           8         2.43        33.58          0       35923      149715      56320

Misses in library cache during parse: 0
Optimizer goal: ALL_ROWS
Parsing user id: 67      (recursive depth: 1)

Elapsed times include waiting on following events:
Event waited on                      Times    Max. wait    Total waited
-----
log file sync                          8         0.00         0.00
  
```



# DBMS Monitor

## ➤ Trace

- Session
- Service/Module/Action
- Client ID

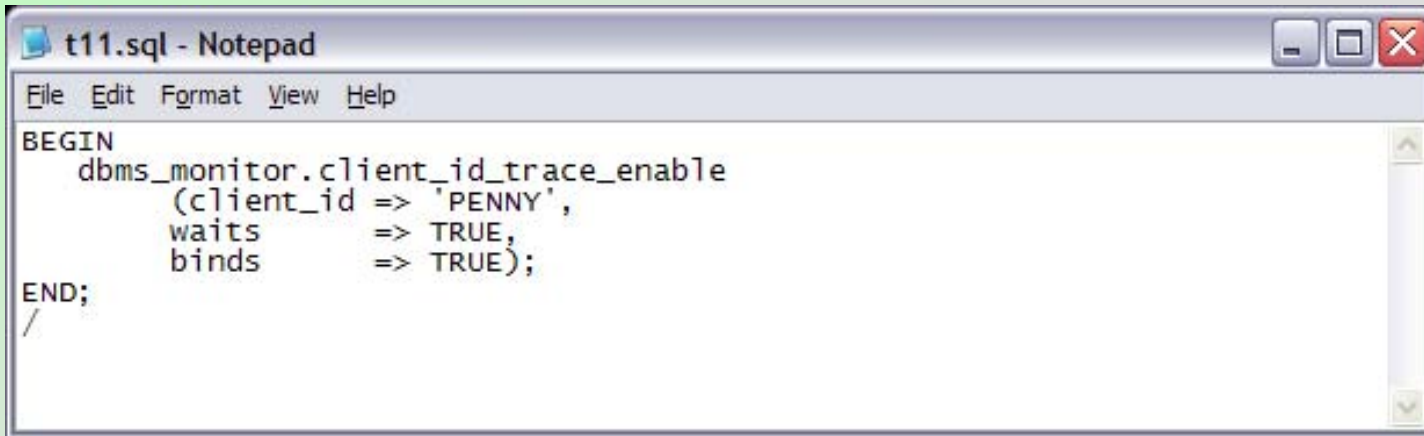
## ➤ Gather statistics for

- Session
- Service/Module/Action
- Client ID

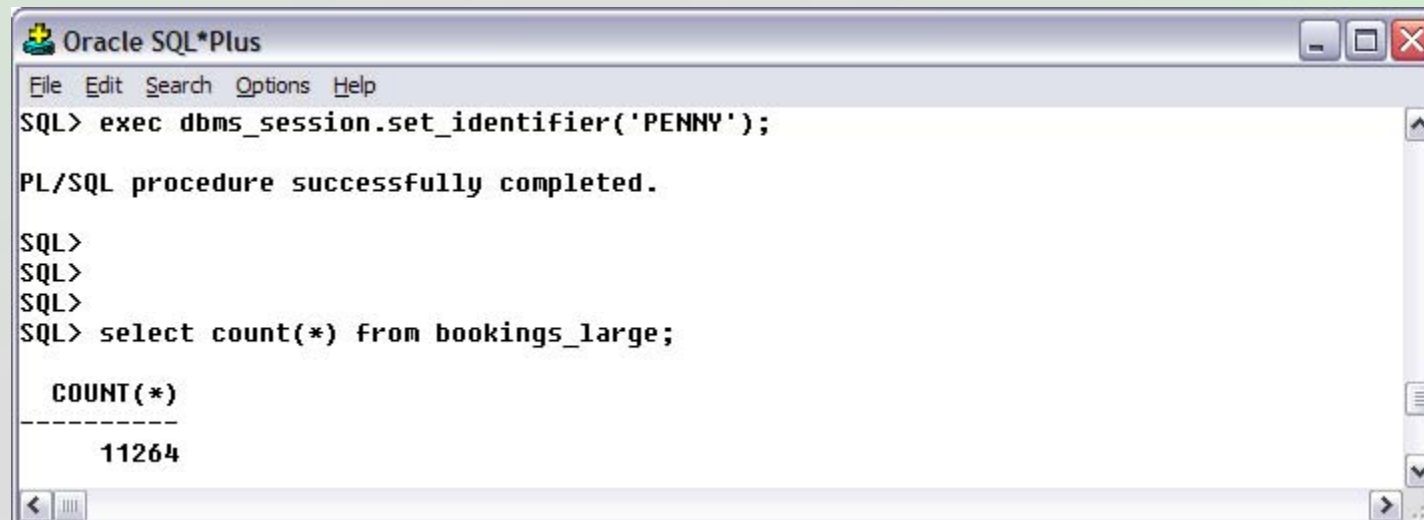
## ➤ Set Module/Action with dbms\_application\_info

## ➤ Set Client Id with dbms\_session

# Trace a Client

A screenshot of a Notepad window titled "t11.sql - Notepad". The window contains the following SQL code:

```
BEGIN
  dbms_monitor.client_id_trace_enable
    (client_id => 'PENNY',
     waits      => TRUE,
     binds      => TRUE);
END;
/
```

A screenshot of an Oracle SQL\*Plus window. The window shows the execution of a PL/SQL procedure and a SQL query. The output of the query is displayed in a table format.

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> exec dbms_session.set_identifier('PENNY');

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL>
SQL> select count(*) from bookings_large;

  COUNT(*)
-----
    11264
```



# Find Active Traces

```
SELECT *  
FROM  
DBA_ENABLED_TRACES
```

```
TRACE_TYPE  
-----  
PRIMARY_ID  
-----  
QUALIFIER_ID1  
-----  
QUALIFIER_ID2          WAITS BINDS INSTANCE_NAME  
-----  
CLIENT_ID  
PENNY  
  
TRUE TRUE
```





# Trace a Client

A screenshot of a Windows WordPad window titled "ora10g\_ora\_7436.trc - WordPad". The window displays the contents of an Oracle trace file. The text is as follows:

```
File Edit View Insert Format Help
Dump file c:\ora\admin\ora10g\udump\ora10g_ora_7436.trc
Sun Jul 31 12:05:49 2005
ORACLE V10.1.0.2.0 - Production vsnsta=0
vsnsql=13 vsnxtr=3
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options
Windows XP Version V5.1 Service Pack 2
CPU : 2 - type 586
Process Affinity: 0x00000000
Memory (A/P) : PH:338M/2046M, PG:2266M/3942M, VA:1756M/2047M
Instance name: ora10g

Redo thread mounted by this instance: 1

Oracle process number: 18

Windows thread id: 7436, image: ORACLE.EXE (SHAD)

*** 2005-07-31 12:05:49.156
*** ACTION NAME: (UPDATEBOOK) 2005-07-31 12:05:49.140
*** MODULE NAME: (RESBOOK) 2005-07-31 12:05:49.140
*** SERVICE NAME: (ora10g.sagecomputing.com.au) 2005-07-31 12:05:49.140
*** CLIENT ID: (PENNY) 2005-07-31 12:05:49.140
*** SESSION ID: (36.43) 2005-07-31 12:05:49.140
```

The window has a standard menu bar (File, Edit, View, Insert, Format, Help) and a toolbar with various icons. The status bar at the bottom indicates "For Help, press F1".

# Consolidate Trace Files

```
trcsess output=trace5.trc clientid='PENNY'
```

```
tkprof trace5.trc trace5.lst sys=no
```

```

trace5.lst - Notepad
File Edit Format View Help
SQL*Net message from client          1          15.39          15.39
*****

select count(*)
from
bookings_large

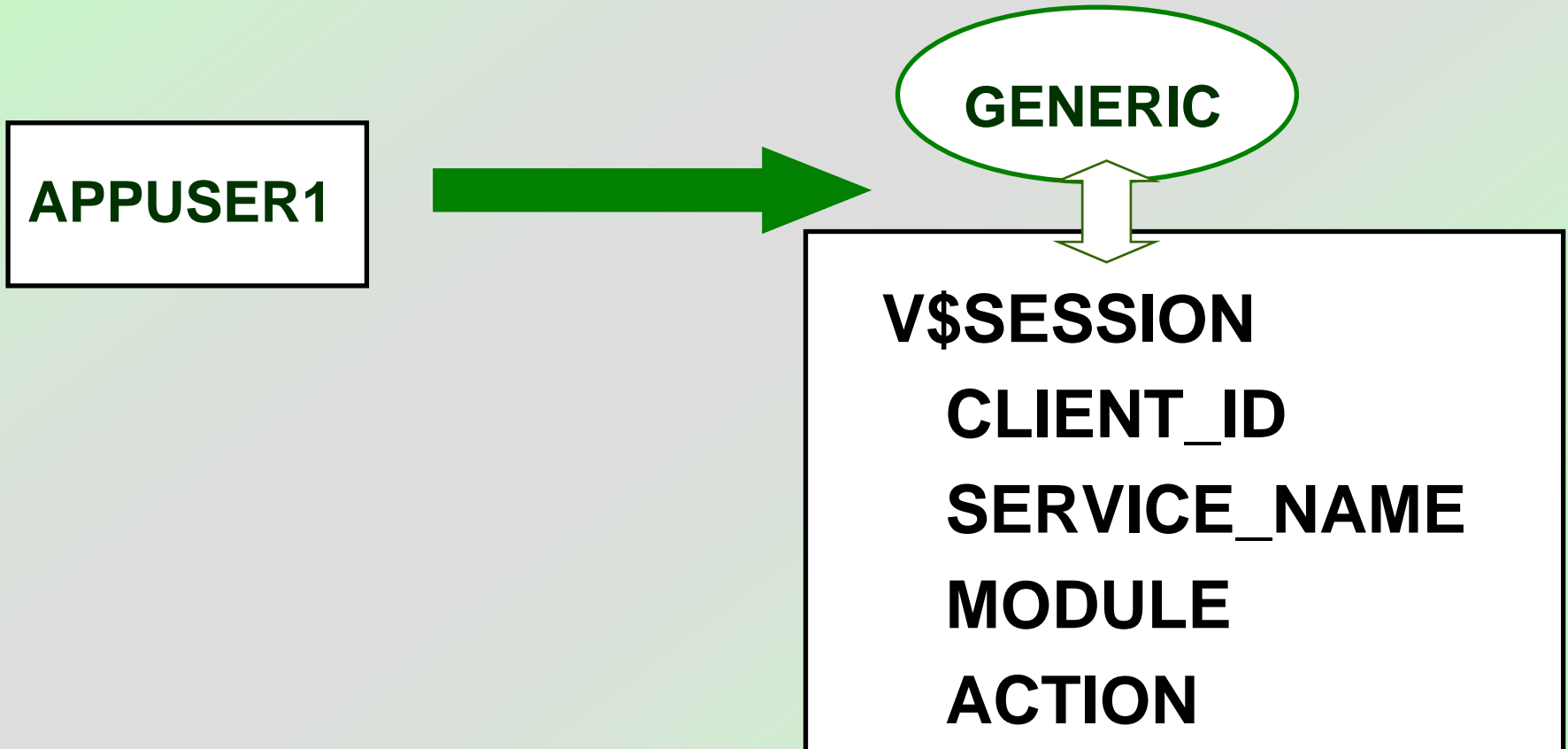
call      count          cpu          elapsed          disk          query          current          rows
-----
Parse      4             0.00           0.00             0             0             0             0
Execute    4             0.01           0.00             0             0             0             0
Fetch      8             0.03           0.05            26            116            0             4
-----
total     16             0.04           0.06            26            116            0             4

Misses in library cache during parse: 1
Optimizer goal: ALL_ROWS
Parsing user id: 67

Rows      Row Source Operation
-----
      1  SORT AGGREGATE (cr=29 pr=26 pw=0 time=49240 us)
    11264  INDEX FAST FULL SCAN BK_EVT5 (cr=29 pr=26 pw=0 time=34861 us)(object id
    54996)

```

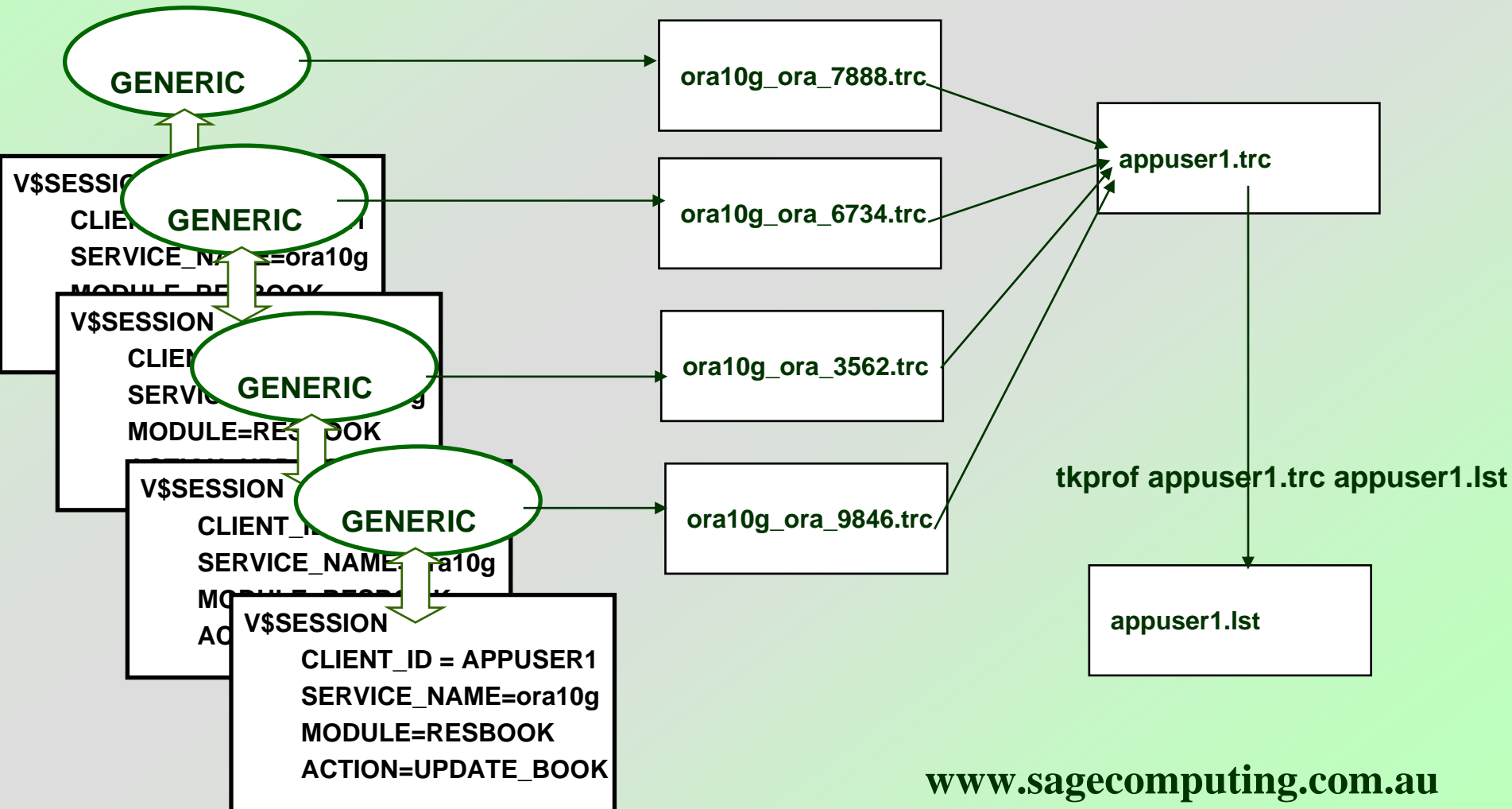
# End to End Monitoring





# End to End Monitoring

trcsess output=appuser1.trc clientid=APPUSER1



# Example - HTMLDB

Employees - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail News RSS Feeds

Address <http://sage4.sagecomputing.com.au:7780/pls/htmldb/f?p=118:6:11105254909161965822::NO> Go Links Norton Internet Security

[USER1] [ ] [ ]

Home Employees Calendar Enter Timesheet Approve Timesheets Reporting Upload files Timesheet2 Parties Search

**Employee Parameters**

Organisation  Go

Name  Go

**Employees** Create

	Name ▲	First Names	Date of Birth	Job Title	Organisation	From	To
	Brunini	Christian	29-DEC-70	System Consultant	SAGE Computing Services	01-MAY-95	
	Cookson	Penny	14-JAN-56	Managing Director	SAGE Computing Services	01-MAY-93	05-FEB-05
	Harris	Eddie	10-NOV-87	System Consultant	SAGE Computing Services	21-JUN-02	
	Marshall	Katie	28-JUN-05	System Consultant	SAGE Training Services	10-JUL-04	
	Njegichddd	Branka		System Consultant	SAGE Training Services	24-AUG-98	
	Smith	Fred	01-JUN-53		Oddbods Professional Services	01-SEP-04	
	Tindallid	Ray		System Consultant	SAGE Computing Services	10-NOV-96	

1 - 7

Internet



# Example HTMLDB

```
SELECT service_name, client_identifier,  
module, action, username  
FROM v$session  
WHERE username = 'HTMLDB_PUBLIC_USER'
```

SERVICE\_NAME

-----

CLIENT\_IDENTIFIER

-----

MODULE

-----

ACTION

USERNAME

-----

ora10g.sagecomputing.com.au

HTML DB

application 118, page 13, sessio HTMLDB\_PUBLIC\_USER

ora10g\_ora\_6436.trc - WordPad

File Edit View Insert Format Help

Dump file c:\ora\admin\ora10g\udump\ora10g\_ora\_6436.trc  
Sun Jul 31 13:22:46 2005  
ORACLE V10.1.0.2.0 - Production vsnsta=0  
vsnsql=13 vsnxtr=3  
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production  
With the Partitioning, OLAP and Data Mining options  
Windows XP Version V5.1 Service Pack 2  
CPU : 2 - type 586  
Process Affinity: 0x00000000  
Memory (A/P) : PH:1067M/2046M, PG:3086M/3942M, VA:1753M/2047M  
Instance name: ora10g

Redo thread mounted by this instance: 1

Oracle process number: 17

Windows thread id: 6436, image: ORACLE.EXE (SHAD)

\*\*\* 2005-07-31 13:22:46.250  
\*\*\* ACTION NAME:(application 118, page 6, session) 2005-07-31 13:22:46.234  
\*\*\* MODULE NAME:(HTML DB) 2005-07-31 13:22:46.234  
\*\*\* SERVICE NAME:(ora10g.sagecomputing.com.au) 2005-07-31 13:22:46.234  
\*\*\* SESSION ID:(29.531) 2005-07-31 13:22:46.234  
=====

PARSING IN CURSOR #54 len=614 dep=0 uid=65 oct=47 lid=65 tim=2059248367 hv=3947649331 ad='6844bef8'  
declare  
rc\_\_ number;  
begin  
owa.init\_cgi\_env(:n\_\_,:nm\_\_,:v\_\_);  
http.HTBUF\_LEN := 255;  
null;  
null;

For Help, press F1

# Example HTMLDB

## Process

\* Page  ▼

\* Name

Type: **PL/SQL anonymous block**

## Process Firing Point

\* Sequence

Process Point  ▼

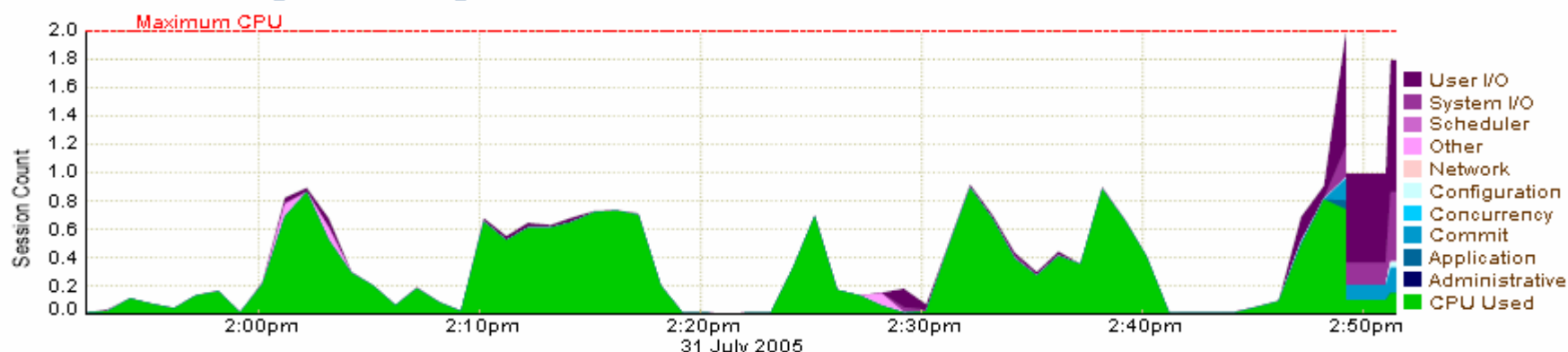
Run Process  ▼

## Source

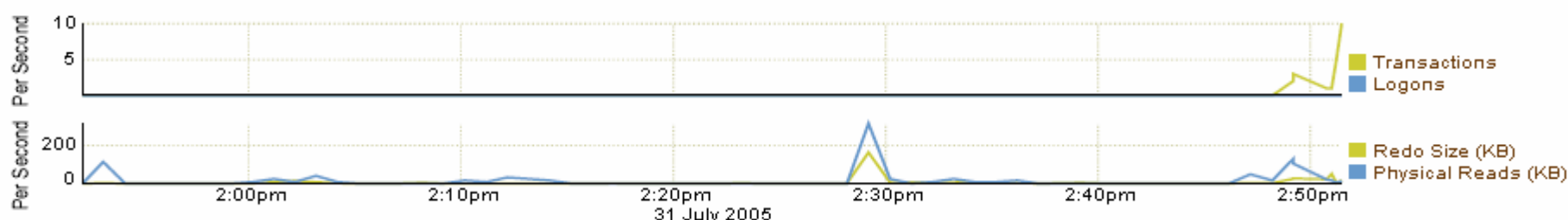
\* Process [\[Download Source\]](#)

```
begin
dbms_session.set_identifier(:APP_USER);
end;
```

## Sessions: waiting and working



## Instance Throughput



View ☒ Per Second ☐ Per Transaction

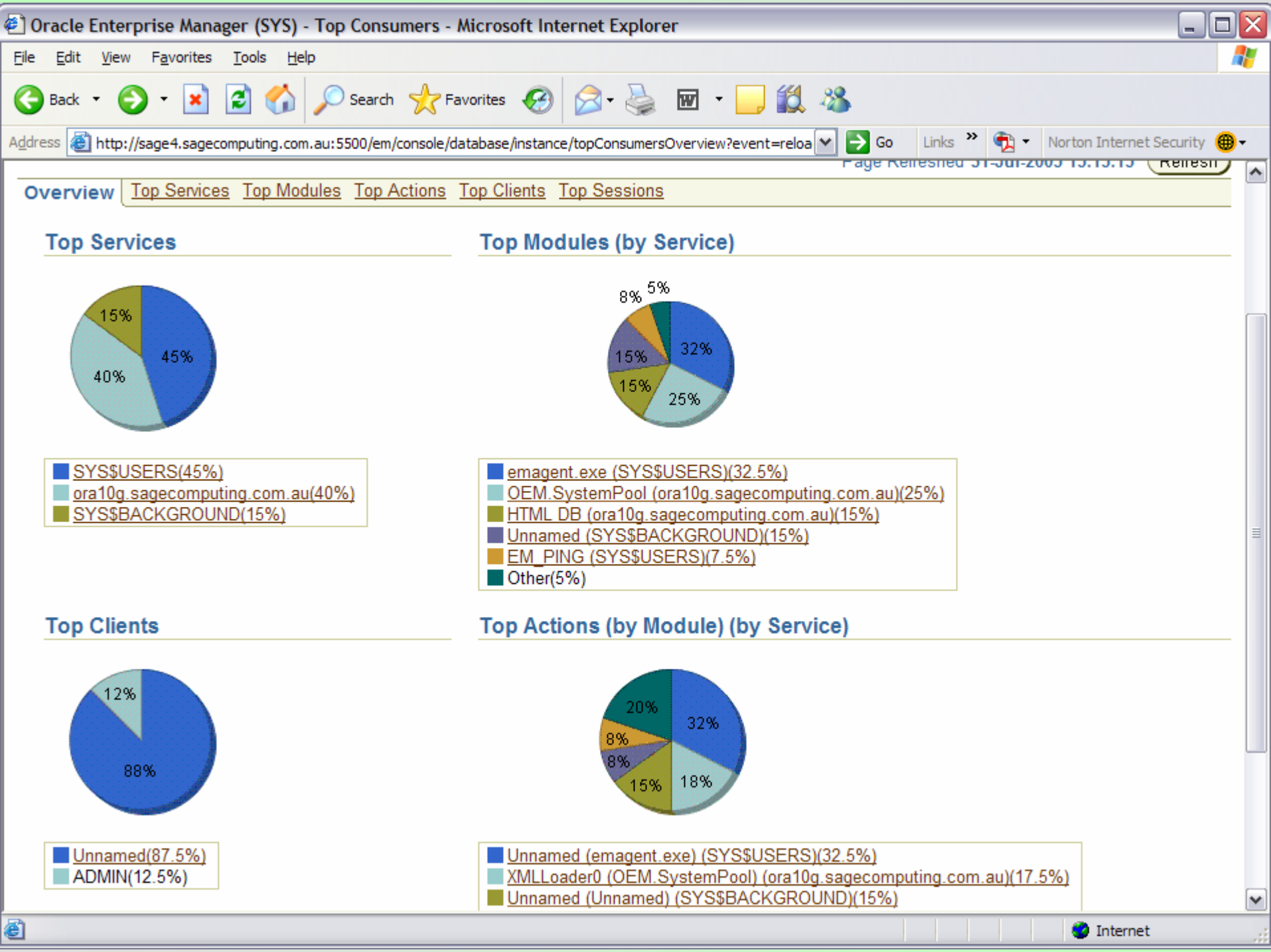
## Additional Monitoring Links

- [Top Sessions](#)
- [Top SQL](#)
- [Blocking Sessions](#)
- [Database Locks](#)
- [Instance Activity](#)
- [Top Consumers](#)
- [Snapshots](#)

[Home](#) [Performance](#) [Administration](#) [Maintenance](#)

## Related Links





# Client: ADMIN

Page Refreshed 31-Jul-2005 15:15:39 [Refresh](#)

## Statistics

Previous 1-25 of 27 Next 2

Name	Delta Value	Cumulative Value
user calls	0	59
DB time	0	2968891
DB CPU	0	1670160
parse count (total)	0	1955
parse time elapsed	0	1369944
execute count	0	7753
sql execute elapsed time	0	2869692
opened cursors cumulative	0	1917
session logical reads	0	11818
physical reads	0	180
physical writes	0	0
redo size	0	20476
user commits	0	16
workarea executions - optimal	0	365
workarea executions - onepass	0	0
workarea executions - multipass	0	0
session cursor cache hits	0	0
user rollbacks	0	0
db block changes	0	246
gc cr blocks received	0	0
gc cr block receive time	0	0
gc current blocks received	0	0



Oracle Enterprise Manager (SYS) - Module: HTML DB - Microsoft Internet Explorer

FileEditViewFavoritesToolsHelp

Back

Search

Favorites

Address

http://sage4.sagecomputing.com.au:5500/em/console/database/instance/SMDetails?event=doLoad&target=ora

Go

Links

Norton Internet Security

ORACLE Enterprise Manager 10g

Database Control

SetupPreferencesHelpLogout

Database

Database: ora10g.sagecomputing.com.au > Top Consumers > Service: ora10g.sagecomputing.com.au > Module: HTML DB

Logged in As SYS

View Data

Real Time: Manual Refresh

Module: HTML DB

Page Refreshed 31-Jul-2005 15:17:00

Refresh

Actions

Statistics

Top Actions

17%

17%

33%

33%

application 4000, page 1, sessi(33.3%)

application 4000, page 4150, se(33.3%)

application 118, page 7, sessio(16.7%)

application 118, page 6, sessio(16.7%)

Active Actions

View

Active Actions

Enable Aggregation

Disable Aggregation

Enable SQL Trace

Disable SQL Trace

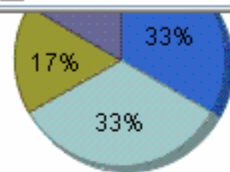
View SQL Trace File

Select All

Select None

			Aggregation	SQL Trace	Delta Elapsed Time	Cumulative Elapsed Time	Delta CPU Time	Cumulative CPU Time	Delta Physical I/O	Cumulative Physical

DoneInternet



- application 4000, page 1, sessio (33.3%)
- application 4000, page 4150, se (33.3%)
- application 118, page 7, sessio (16.7%)
- application 118, page 6, sessio (16.7%)

## Active Actions

View Active Actions

Enable Aggregation

Disable Aggregation

Enable SQL Trace

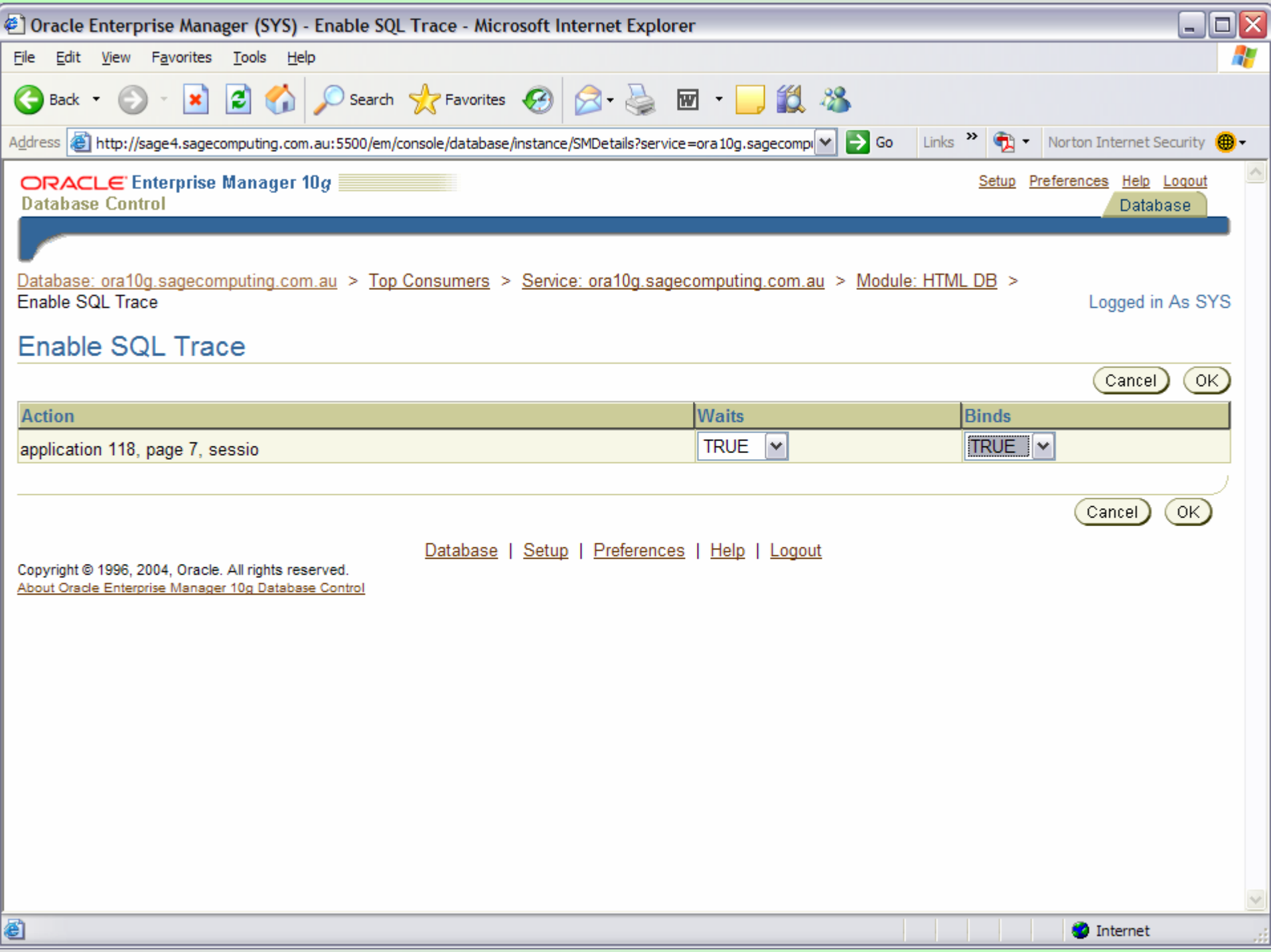
Disable SQL Trace

View SQL Trace File

Select All | Select None

Select	Action	Activity (% for the last 5 minutes)	Aggregation Enabled	SQL Trace Enabled	Delta Elapsed Time (seconds)	Cumulative Elapsed Time (seconds)	Delta CPU Time (seconds)	Cumulative CPU Time (seconds)	Delta Physical I/O (blocks)	Cumulative Physical I/O (blocks)
<input type="checkbox"/>	<a href="#">application 4000, page 1, sessio</a>	33.3	FALSE	FALSE						
<input type="checkbox"/>	<a href="#">application 4000, page 4150, se</a>	33.3	FALSE	FALSE						
<input checked="" type="checkbox"/>	<a href="#">application 118, page 7, sessio</a>	16.7	FALSE	FALSE						
<input type="checkbox"/>	<a href="#">application 118, page 6, sessio</a>	16.7	FALSE	FALSE						

Actions [Statistics](#)



[illegible]

1



--	--

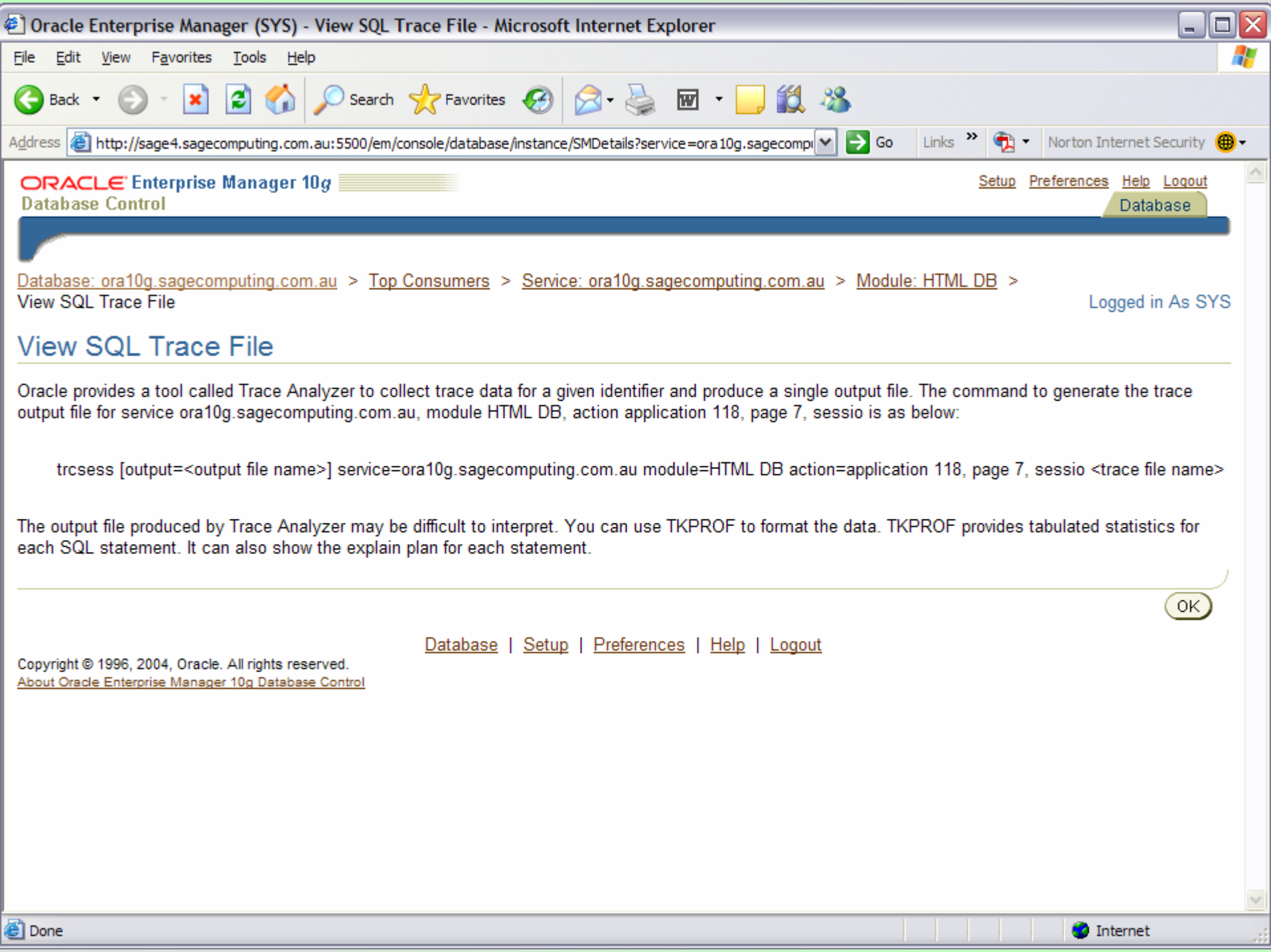
--	--

1

11

--	--

--	--



# New Hints

- **Specify index with name or column**  
--+ INDEX(org (org\_id name))
- **Use nested loop with a specific index**  
--+ USE\_NL\_WITH\_INDEX (alias (indexname))
- **Name a query block**  
--+ QB\_NAME (q1)
- **Do not perform query transformation**  
--+ NO\_QUERY\_TRANSFORMATION
- **Perform index skip scan**  
--+ INDEX\_SS



# New/Changed Hints

## ➤ Prevent operations

NO\_USE\_NL  
NO\_USE\_MERGE  
NO\_USE\_HASH  
NO\_INDEX\_FFS  
NO\_INDEX\_SS

## ➤ Deprecated Hints

AND\_EQUAL  
HASH\_AJ  
MERGE\_AJ  
HASH\_SJ  
MERGE\_SJ  
STAR  
ORDERED\_PREDICATES

## ➤ Renamed Hints

New	Old
NO_PARALLEL	NOPARALLEL
NO_PARALLEL_INDEX	NOPARALLEL_INDEX
NO_REWRITE	NOREWRITE



# New Tuning Tools

- **Oracle Enterprise Manager**
- **SQL Tuning Advisor**
- **SQL Access Advisor**
- **Automatic Database Diagnostics Monitor**
- **Automatic Workload Repository**
- **SQL Tuning Sets**
- **SQL Profiling**





# Questions?

**SAGE Computing Services**  
Customised Oracle Training Workshops  
and Consulting  
[www.sagecomputing.com.au](http://www.sagecomputing.com.au)

**Penny Cookson - Managing Director**

[www.sagecomputing.com.au](http://www.sagecomputing.com.au)