



오렌지를 이용한 SQL모니터링 및 튜닝 가이드

DBMS Application Development &
Performance Management Tool

Orange 2010 소개

- Orange 2010 제품군
 - Orange for Oracle 2010
 - Orange for Altibase 2010
 - Orange for Tiberio 2010
 - Orange for DB2 2010
 - Orange for Sybase IQ 2010
 - Orange for Telcobase 2010
 - Orange for MS-SQL 2010 (예정)
 - Orange for Sybase ASE 2010 (예정)
- 새로운 툴/기능 추가
 - Wait Event Monitor
 - AWR Manager
 - Data Generation Tool
 - Export / Import Tool 에 Data Pump 기능 추가
 - Stats Manager에 DBMS_STATS 패키지를 이용한 통계정보 수집 기능 추가

SQL Monitor를 이용한 악성 쿼리 찾기

- 검색 조건 입력하기
 - How long does it take?
실행시간이 긴 SQL문 찾기
 - How many read blocks?
 - Memory 에서 읽은 블록 개수
 - 단위 실행당 Memory 에서 읽은 블록 개수
 - Disk 에서 읽은 블록 개수
 - What times was executed?
 - 빈번하게 사용되는 쿼리 찾기
 - Literal SQL 찾기
 - Which word is included?
특정 단어가 포함된 SQL문 찾기
- 검색 조건 입력 후 Search 버튼 클릭

The screenshot shows a 'Search' dialog box with the following sections:

- How long does it take?**
 - Don't consider
 - Critical (more than 0.1 sec)
 - OLTP (more than 1 sec)
 - Online Batch (more than 10 sec)
 - Specify Time (more than) [] sec
- How many read blocks?**
 - Don't consider
 - Small (more than 1000)
 - Medium (more than 10000)
 - Large (more than 100000)
 - Specify blocks (more than) [] blocks
- What times was executed?**
 - Don't consider
 - Literal (= 1)
 - Rare (less than 10)
 - Frequent (more than 100)
 - Specify Executions [] times
- Which word is included?**
[]

Search

SQL Monitor를 이용한 악성 쿼리 찾기

- 검색된 SQL 리스트에 보여줄 SQL 문장의 길이를 옵션에서 정할 수 있다.
- SQL 문장을 파란색으로 Preview 형태로 볼 수 있는 옵션 제공
- 팝업메뉴에서 Explain Plan 선택하거나 SQL 문장을 더블클릭 시 Plan Tool 과 연동

The screenshot displays the Oracle SQL Monitor interface. The main window shows a table of SQL queries with columns for SQL ID, DB User, Address, Hash Value, Child Number, Buffer Gets, Executions, Buffer Gets per Execution, Disk Reads, Rows per Execution, and Elapsed Time. The queries listed include various SELECT statements with complex joins and subqueries.

The 'Orange Options' dialog box is open, showing a tree view of tool categories. The 'General' option under 'SQL Monitor' is selected. The 'Query Length' is set to 256, and the 'Preview Mode' checkbox is checked. The dialog also includes buttons for 'Default', 'OK', 'Cancel', and 'Apply'.

Plan Tool 에서 튜닝하기

- SQL Monitor 에서 연동된 경우 SQL 문장이 복사되고 자동으로 Explain Plan 정보 제공
- Oracle 9i 이상인 경우 Runtime Plan 제공
 - Grid Plan 탭의 Plan 정보와 비교 용이하도록 ID 제공
- Oracle 10g 이상인 경우 Bind Data 정보 제공.
 - 실행 결과를 바로 볼 수 있도록 Bind 변수 값을 Bind 변수 창에 자동 입력.

The screenshot shows the Oracle Plan Tool interface. The top pane displays the original SQL query. The bottom pane shows the execution plan, which is a tree structure of operations. A yellow circle highlights the 'Bind Variable List' window, which is a small table with the following data:

#	Variable	Value	Type
1	:SYS_NUMB...	5	String
2	:SEG_FILE	5	String
3	:SEG_BLO...	179	String
4			String

The execution plan in the bottom pane shows the following operations:

- 1. SELECT STATEMENT Optimizer=ALL_ROWS (Cost=3K)
- 2. HASH JOIN (Cost=3K Card=1 Bytes=340)
- 3. VIEW (Cost=3K Card=1 Bytes=69)
- 4. HASH (GROUP BY) (Cost=3K Card=1 Bytes=315)
- 5. HASH JOIN (Cost=760 Card=1 Bytes=216)
- 6. TABLE ACCESS (CLUSTER) OF 'SYS.TS#' (Cost=1 Card=1 Bytes=11)
- 7. INDEX (UNIQUE SCAN) OF 'SYS.L.TS#' (Cost=0 Card=1)
- 8. VIEW OF 'SYS.DBA.SEGMENTS' (Cost=786 Card=3 Bytes=615)
- 9. VIEW OF 'SYS.SYS.DBA.SEGS' (Cost=786 Card=3 Bytes=339)
- 10. UNION-ALL
- 11. NESTED LOOPS (Cost=774 Card=1 Bytes=290)
- 12. NESTED LOOPS (OUTER) (Cost=773 Card=1 Bytes=268)
- 13. NESTED LOOPS (Cost=772 Card=1 Bytes=265)

Plan Tool 에서 튜닝하기

- ▶ 두 쿼리의 실행계획을 비교하기 용이하도록 한 화면에서 제공
- ▶ 플랜 정보를 상하 혹은 좌우로 위치 변경이 가능.
- ▶ 플랜 정보를 더블 클릭하면 해당 단계의 테이블 혹은 인덱스 상세 정보 제공

The screenshot displays the Oracle Plan Tool interface with two execution plans side-by-side for comparison. The top window shows the 'Original' plan, and the bottom window shows the 'Turning' plan. Both plans are for the same SQL query: `select * from emp where deptno in (select deptno from dept where dname = 'SALES');`

Original Plan:

- SELECT STATEMENT OPTIMIZER=ALL_ROWS
- MERGE JOIN (Cost=6 Card=5 Bytes=250)
- TABLE ACCESS (BY INDEX ROWID) OF 'DEPT' (TABLE) (Cost=2 Card=1 Bytes=13)
- INDEX (FULL SCAN) OF 'PK_DEPT' (INDEX (UNIQUE)) (Cost=1 Card=4)
- SORT (JOIN) (Cost=4 Card=15 Bytes=555)
- TABLE ACCESS (FULL) OF 'EMP' (TABLE) (Cost=3 Card=15 Bytes=555)

Turning Plan:

- SELECT STATEMENT OPTIMIZER=ALL_ROWS
- MERGE JOIN (Cost=6 Card=5 Bytes=250)
- SORT (UNIQUE) (Cost=2 Card=1 Bytes=13)
- TABLE ACCESS (BY INDEX ROWID) OF 'DEPT' (TABLE) (Cost=2 Card=1 Bytes=13)
- INDEX (FULL SCAN) OF 'PK_DEPT' (INDEX (UNIQUE)) (Cost=1 Card=4)
- SORT (JOIN) (Cost=4 Card=15 Bytes=555)
- TABLE ACCESS (FULL) OF 'EMP' (TABLE) (Cost=3 Card=15 Bytes=555)

The interface includes a 'Text Plan' view, a 'Grid Plan' view, and a 'Schema' view. The 'Output' window shows the execution plan details. The status bar at the bottom indicates 'Ln 5, Col 24 0.17 sec.' for the original plan and 'Ln 5, Col 24 0.08 sec.' for the turning plan.

Plan Tool 에서 튜닝하기

- ▶ 플랜정보 뿐 아니라 실행 결과도 비교가 용이하도록 상/하 혹은 좌/우 배치 가능
- ▶ Plan Tool 뿐만 아니라 SQL Tool 에서도 동일한 기능 제공

Original

```

1 select * from emp
2 where deptno in (select deptno
3 from dept
4 where dname = 'SALES')
    
```

Turning1

```

1 select * from emp
2 where deptno in (select deptno
3 from dept
4 where dname = 'SALES')
    
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	
1	7499	ALLEN	SALESMAN	7698	1981/02/20 00:00:00	1600	300	30
2	7521	WARD	SALESMAN	7698	1981/02/22 00:00:00	1250	500	30
3	7654	MARTIN	SALESMAN	7698	1981/09/28 00:00:00	1250	1400	30
4	7698	BLAKE	MANAGER	7839	1981/05/01 00:00:00	2850		30
5	7844	TURNER	SALESMAN	7698	1981/09/08 00:00:00	1500	0	30
6	7900	JAMES	CLERK	7698	1981/12/03 00:00:00	950		30

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	
1	7499	ALLEN	SALESMAN	7698	1981/02/20 00:00:00	1600	300	30
2	7521	WARD	SALESMAN	7698	1981/02/22 00:00:00	1250	500	30
3	7654	MARTIN	SALESMAN	7698	1981/09/28 00:00:00	1250	1400	30
4	7698	BLAKE	MANAGER	7839	1981/05/01 00:00:00	2850		30
5	7844	TURNER	SALESMAN	7698	1981/09/08 00:00:00	1500	0	30
6	7900	JAMES	CLERK	7698	1981/12/03 00:00:00	950		30

SQL1

```

1 select * from emp;
2
3 select * from dept;
    
```

SQL2

```

1 select * from emp;
2
3 select * from dept;
    
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	
1	7369	SMITH	CLERK	7902	1980/12/17 00:00:00	800		20
2	7499	ALLEN	SALESMAN	7698	1981/02/20 00:00:00	1600	300	30
3	7521	WARD	SALESMAN	7698	1981/02/22 00:00:00	1250	500	30
4	7566	JONES	MANAGER	7839	1981/04/02 00:00:00	2975		20
5	7654	MARTIN	SALESMAN	7698	1981/09/28 00:00:00	1250	1400	30
6	7698	BLAKE	MANAGER	7839	1981/05/01 00:00:00	2850		30
7	7782	CLARK	MANAGER	7839	1981/06/09 00:00:00	2450		10
8	7788	SCOTT	ANALYST	7566	1987/04/19 00:00:00	3000		20
9	7839	KING	PRESIDENT		1981/11/17 00:00:00	5000		10
10	7844	TURNER	SALESMAN	7698	1981/09/08 00:00:00	1500	0	30
11	7876	ADAMS	CLERK	7788	1987/05/23 00:00:00	1100		20
12	7900	JAMES	CLERK	7698	1981/12/03 00:00:00	950		30
13	7902	FORD	ANALYST	7566	1981/12/03 00:00:00	3000		20
14	7934	MILLER	CLERK	7782	1982/01/23 00:00:00	1300		10

DEPTNO	DNAME	LOC
1	60	MANAGEMENT SEATTLE
2	10	ACCOUNTING NEW YORK
3	20	RESEARCH DALLAS
4	30	SALES CHICAGO
5	40	OPERATIONS BOSTON
6	50	SUPPORT DETROIT

Plan Tool 에서 튜닝하기

- ▶ 튜닝에 필요한 정보를 편리하게 제공
 - Schema 탭을 통하여 쿼리에 사용된 테이블/인덱스 상세 정보 제공
 - Trace 탭을 통하여 빠르고 편리하게 Trace 정보 제공

Schema Tab Information:

Column Name	Nullable	Column Type	Distinct	Buckets
EMPNO	NOT NULL	NUMBER(4)	15	1
ENAME		VARCHAR2(10)	15	1
JOB		VARCHAR2(9)	5	5
MGR		NUMBER(4)	6	1
HIREDATE		DATE	14	1
SAL		NUMBER(7,2)	13	1
COMM		NUMBER(7,2)	4	1
DEPTNO		NUMBER(2)	3	1

Index Information:

Index Name	Index Type	Distinct	Rows	Last Analyzed
PK_EMP	NORMAL	15	15	2010/08/21 09:39:30

Trace Tab Execution Statistics:

Call	Count	CPU Time	Elapsed Time	Disk	Query	Current	Rows
Parse	1	0.000	0.001	0	0	0	0
Execute	1	0.000	0.000	0	0	0	0
Fetch	1	0.000	0.016	8	9	0	1
Total	3	0.000	0.017	8	9	0	1

Execution Plan:

```

1 MERGE JOIN (cr=9 pr=8 pw=8 time=0 us cost=6 size=250 card=5)
1 TABLE ACCESS BY INDEX ROWID DEPT (cr=2 pr=2 pw=2 time=0 us cost=2 size=13 card=1)
3 INDEX FULL SCAN PK_DEPT (cr=1 pr=1 pw=1 time=0 us cost=1 size=0 card=4)
1 SORT JOIN (cr=7 pr=6 pw=6 time=0 us cost=4 size=555 card=15)
15 TABLE ACCESS FULL EMP (cr=7 pr=6 pw=6 time=1 us cost=3 size=555 card=15)
    
```


Plan Tool 에서 튜닝하기

➤ Trace 정보

통계 정보	설 명
Count	각 처리 단계 별 실행된 수
CPU	각 처리 단계 별 CPU 소모 시간(초)
Elapsed	각 처리 단계의 시작에서 종료까지 총 경과 시간(초)
Disk	각 처리 단계 별 물리적인 디스크 블록 접근한 횟수
Query	각 처리 단계 별 읽은 변경된 버퍼 블록 수(Consistent Read) 읽기 일관성 관련/주로 SELECT 문
Current	각 처리 단계 별 현 세션에만 유효한 버퍼 블록을 접근한 수(Current Read) 주로 INSERT,UPDATE,DELETE 작업 시 발생
Rows	각 처리 단계 별 읽은 총 행수 Fetch 단계 : SELECT에 의해 질의된 행 수 Execute 단계 : INSERT, UPDATE, DELETE 문에서 처리된 행수

Trace Tool 의 활용

- trc 파일에 존재하는 모든 대기 이벤트 정보 조회.
- 특정 대기 이벤트를 선택하면 하단에서 해당 대기 이벤트가 발생한 쿼리 리스트 제공

The screenshot shows the Oracle Trace Tool interface. The main window displays a 'WAIT SUMMARY' table with columns: Wait Event Name, Count, Wait(sec), and Max. Wait. Below this is a 'Query List' table with columns: Statement, Total Wait Count, Total Wait Time, Wait Count for Selected Wait, and Total Wait Time for Selected Wait.

Wait Event Name	Count	Wait(sec)	Max. Wait
'log file sync'	52	0.073	0.009
'gc current grant 2-way'	11	0.005	0.002
'gc cr grant 2-way'	1742	0.682	0.599
'read by other session'	2	0.001	0.001
'latch: shared pool'	2	0.003	0.003
'SQL*Net more data to client'	12739	0.323	0.292
'latch free'	4	0.003	0.003
'row cache lock'	31	0.031	0.012
'db file sequential read'	20252	30.015	7.462
'db file scattered read'	4136	5.198	1.084
'db file parallel read'	312	0.805	0.116
'latch: cache buffers chains'	3	0.000	0.000
'library cache lock'	14	0.012	0.002
'gc cr multi block request'	1323	1.174	0.233
'SQL*Net message from client'	16850	1461.930	621.578
'gc current block 2-way'	5814	7.289	2.059
'SQL*Net message to client'	16850	0.010	0.005
'direct path read'	3087	0.008	0.003

Statement	Total Wait Count	Total Wait Time	Wait Count for Selected Wait	Total Wait Time for Selected Wait
SELECT t.id, s.sectionid, s.siteid, s.priorit... 20097	23,024,895	1	0,000,487	
SELECT COUNT(*) FROM GO... 423	2,247,383	1	0,000,922	

Trace Tool 의 활용

- trc 파일에 존재하는 모든 쿼리 문장을 Hierarchy 관계로 제공.
- 특정 SQL문장을 선택하면 FULL TEXT QUERY와 바인드 변수 값을 제공

The screenshot shows the Oracle Trace Tool interface. The main window displays a list of SQL statements with columns for Parse C., Exec C., Fetch C., CPU, Elapsed, Disk, Query, Current, Rows, Parse Mi., Exec Mis, and Opti Goal. Below this, the 'Detail Information' section shows the 'SQL Statement' and 'Bind' variables.

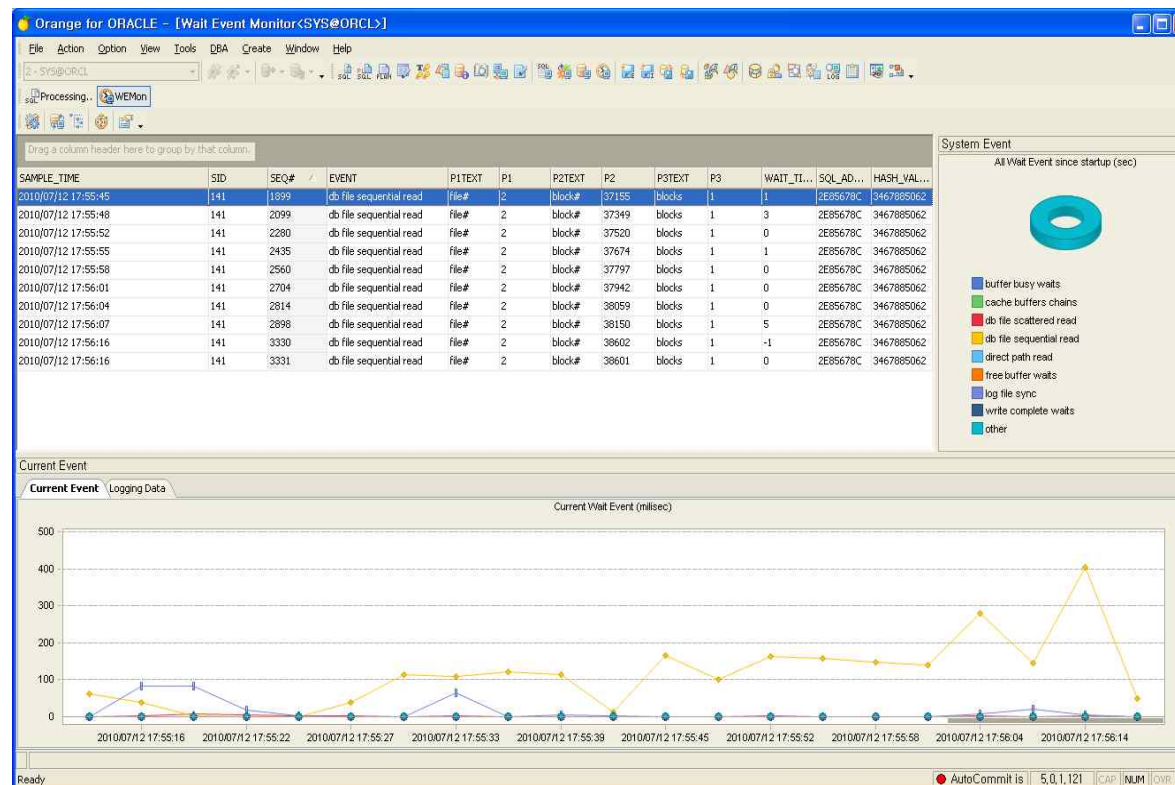
#	SQL	Parse C...	Exec C...	Fetch C...	CPU	Elapsed	Disk	Query	Current	Rows	Parse Mi...	Exec Mis	Opti Goal
11	SELECT SITE_ID SITEID, NAME, HOME_URL HOMEU...	46	46	0	0,020	0,010	0	92	0	46	0	0	ALL_ROWS
12	SELECT SITE_ID AS siteid, TE...	71	87	0	0,040	0,061	0	326	0	628	0	0	ALL_ROWS
13	SELECT TABLE_ID AS tableid, ...	83	349	0	0,040	0,092	3	728	0	2992	0	0	ALL_ROWS
14	SELECT tid.sectionid,siteid,priority,status,modifyid,ap...	33	33	0	1,790	3,113	352	38867	0	133	0	0	ALL_ROWS
15	SELECT TABLE_ID AS tableid, ...	0	159	0	0,080	0,065	0	505	0	159	0	0	ALL_ROWS
16	SELECT tid.sectionid,siteid,priority,status,modifyid,ap...	30	30	0	0,070	0,054	0	1830	0	150	0	0	ALL_ROWS
17	SELECT tid.sectionid,siteid,priority,status,modifyid,ap...	33	33	0	0,060	0,056	0	2277	0	165	0	0	ALL_ROWS
18	SELECT id, sectionid, siteid, title FROM (...)	31	31	0	0,130	0,283	15	8596	0	31	0	0	ALL_ROWS
19	SELECT tid.sectionid,siteid,priority,status,modifyid,ap...	8	8	0	0,020	0,037	0	1384	0	40	0	0	ALL_ROWS
20	SELECT tid.sectionid,siteid,priority,status,modifyid,ap...	9	9	0	0,000	0,010	0	288	0	45	0	0	ALL_ROWS
21	SELECT PROPERTY_ID as propertyid FROM GO_CA...	39	40	0	0,010	0,008	0	79	0	148	0	0	ALL_ROWS
22	SELECT tid.sectionid,siteid,priority,status,modifyid,ap...	32	32	0	0,250	0,296	0	10944	0	160	0	0	ALL_ROWS
23	SELECT SITE_ID AS siteid, BANNER_KIND AS...	10	10	0	0,000	0,002	0	40	0	20	0	0	ALL_ROWS
24	SELECT SITE_ID AS siteid, TEMP...	2	2	0	0,000	0,000	0	6	0	2	0	0	ALL_ROWS
25	SELECT TABLE_ID AS tableid, ...	1	1	1	0,010	0,005	0	10	0	1	1	0	ALL_ROWS
26	SELECT ID id, SECTION_ID...	50	50	0	0,010	0,119	11	201	0	50	0	0	ALL_ROWS
27	UPDATE GO_NEWS_DATASET HIT = NVL(HIT,0) + 1W...	47	0	0	0,060	0,103	13	141	426	47	0	0	ALL_ROWS
28	SELECT LEVEL as depth,oid.edatid,ename, opa...	48	437	0	0,560	8,509	1559	4744	0	3997	0	0	ALL_ROWS

Bind	Data Type	Value
0	VARCHAR2	'nts'
1	NUMBER	0
2	NUMBER	3

Wait Event Monitor 활용하기

➤ Wait Event Monitor 기능

- 세션 별로 최근 발생한 대기 이벤트를 최대 10까지 리스트로 제공
- 리스트 더블 클릭 시 플랜 툴 연계
- DB Startup 이후 발생한 대기 이벤트 빈도를 도넛 모양의 차트로 제공
- 현재 발생 중인 대기 이벤트 정보를 선 그래프로 제공



Wait Event Monitor 활용하기

➤ Wait Event Monitor 기능

- Logging Data 탭에서 구간을 지정하여 logging 데이터 조회 가능
- 조회된 데이터를 더블 클릭 시 플랜 툴과 연계

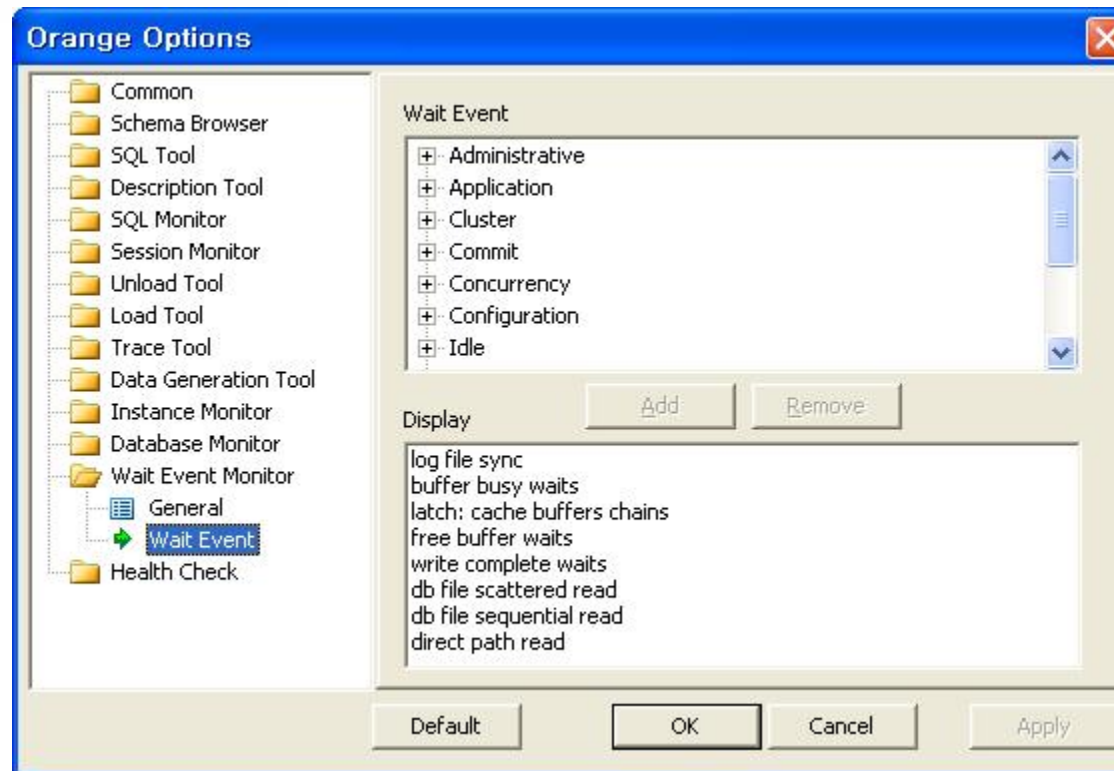
The screenshot displays the Oracle SQL Developer interface. The 'Current Event' window is open to the 'Logging Data' tab. The 'FROM' field is set to '2010/06/14 23:23:52.-' and the 'TO' field is '2010/07/14 23:23:56.-'. A table of logging data is shown with columns: SAMPLE_TIME, SID, SERIAL#, WAIT_TIME, SECONDS_IN_WAIT, and TEXT. The table contains 9 rows of data. To the right, a SQL query is displayed in a text area, which is a PL/SQL procedure to insert wait event data into the 'orange.wait_event_log' table. The query uses a cursor to select data from the 'V\$SESSION_WAIT' view and inserts it into the logging table.

	SAMPLE_TIME	SID	SERIAL#	WAIT_TIME	SECONDS_IN_WAIT	TEXT
1	20100713134227	157	17137	1	0	INSERT INTO orange.orange_wait_event_log SELECT * FROM
2	20100712174846	141	34693	4	0	INSERT INTO VALUES (:B1
3	20100712174810	141	34693	3	0	INSERT INTO VALUES (:B1
4	20100712174722	137	414	0	2	select count(*
5	20100712174016	137	414	0	2	select count(*
6	20100712173840	141	34693	5	0	declare v_cnt begin loop ir (id
7	20100712173831	137	414	0	1	select count(*
8	20100712172831	137	414	0	3	select count(*
9	20100712172501	137	414	1	0	select count(*

```
1  -- wait event name : db file sequential read
2  INSERT
3  INTO orange.wait_event_log
4  SELECT *
5  FROM (SELECT TO_CHAR(SYSDATE, 'YYYYMMDDHH24MISS') DT,
6          W.WAIT_TIME,
7          W.SECONDS_IN_WAIT,
8          N.EVENT#,
9          S.SID,
10         S.SERIAL#,
11         T.ADDRESS,
12         T.HASH_VALUE,
13         T.PIECE,
14         T.SQL_TEXT
15        FROM V$SESSION_WAIT W,
16         V$SESSION S,
17         V$SQLTEXT T,
18         V$EVENT_NAME N
19        WHERE (N.WAIT_TIME = W.WAIT_TIME))
```

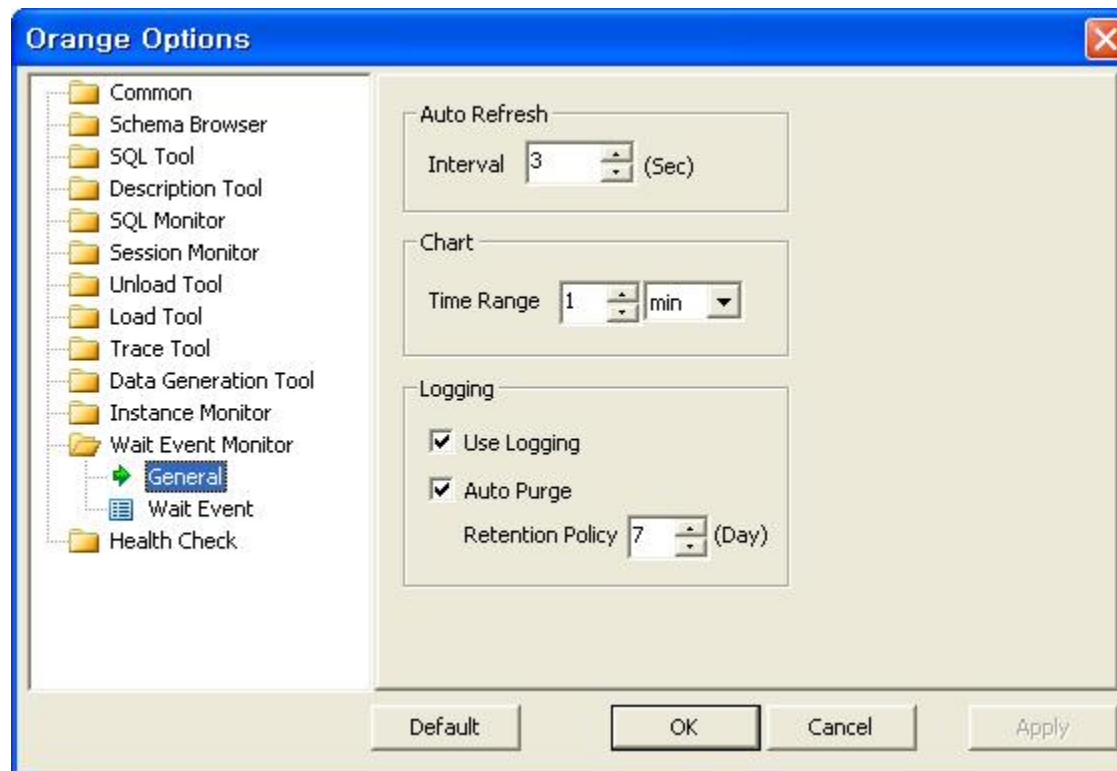
Wait Event Monitor 활용하기

- 옵션 대화상자에서 감시할 대기 이벤트를 최대 8개 까지 선택 가능
- 선택한 대기 이벤트가 1초 이상 대기 하고 있는 경우 해당 세션과 쿼리를 DB에 저장



Wait Event Monitor 활용하기

- 옵션 대화상자에서 logging 여부 설정
- logging 데이터를 얼마 동안 보관할지 여부 설정



AWR Manager를 이용한 악성 쿼리 찾기

❖ Snapshot Manager 사용하기

➤ 수집 관련 설정

- Snapshot Interval : 간격
- Retention : 보유 시간
- Top N SQL : Top SQL 개수 지정
- 설정 변경 후 툴바 에서 체크버튼 클릭

➤ Overall Snapshot

- 시작/마지막 스냅샷 ID
- 시작/마지막 스냅샷 시간
- AWR Table과 인덱스 개수
- AWR 사이즈
- 전체 스냅샷 개수

➤ Snapshot Information

- Add / Remove 버튼 클릭하여 추가 또는 삭제
- 스냅샷 번호를 선택하면 스냅샷 정보 제공

Snapshot Manager

Collection Setting

Snapshot Interval	+00000 01:00:00,0
Retention	+00007 10:00:00,0
Top N SQL	DEFAULT

Overall Snapshot

First Snapshot ID	125
Last Snapshot ID	651
First Snapshot Time	2010/08/16 22:00:07
Last Snapshot Time	2010/09/09 12:00:41
Total AWR Tables	136
Total AWR Indexes	141
AWR Size	69MB
Total # of Snapshots	206

Snapshot Information

Snapshot List Add / Remove

Snapshot

Snapshot ID	
DB ID	
Instance #	
Startup Time	
Begin Interval Time	
End Interval Time	
Flush Elapsed	
Error Count	

Snapshot Interval

The new interval setting between each snapshot, in units of minutes. The specified value must be in the range MIN_RETENTION (10 minutes) to MAX_RETENTION (1 year).

If ZERO is specified, automatic and manual snapshots will be disabled. A large system-defined value will be used as the retention setting.

If NULL is specified, the current value is preserved.

AWR Manager를 이용한 악성 쿼리 찾기

- Oracle 에서 제공하는 5가지 유형의 Report 를 Text 혹은 Html 파일 형태로 제공
 - ADDM (Auto Database Diagnostic Monitor) Report
 - AWR (Automatic Workload Repository) Report
 - AWR SQL Report
 - AWR Diff Report
 - ASH (Active Session History) Report

Report Manager

Report Configuration

Report Mode: ADDM Report
Report Format: Html

Scope

Snapshot: Load Baseline

Instance	1
Starting Sanpshot	...
Ending Sanpshot	...

Report Mode

This table function displays the ADDM(Automatic Database Diagnostic Monitor) Report

Report Manager

Report Configuration

Report Mode: AWR Report
Report Format: Html

Scope

Snapshot: Load Baseline

Instance	1
Starting Sanpshot	...
Ending Sanpshot	...

Report Mode

This table function displays the AWR report

Report Manager

Report Configuration

Report Mode: AWR Diff Report
Report Format: Html

Scope

Compare: Load Baseline

Instance	1
Starting Sanpshot	...
Ending Sanpshot	...

To: Load Baseline

Instance	1
Starting Sanpshot	...
Ending Sanpshot	...

Report Mode

This table function displays the AWR Compare Periods report

Report Manager

Report Configuration

Report Mode: AWR SQL Report
Report Format: Html

Scope

Snapshot: Load Baseline

Instance	1
Starting Sanpshot	...
Ending Sanpshot	...
SQL ID	...

Report Mode

This table function displays the AWR SQL Report

Report Manager

Report Configuration

Report Mode: ASH Report
Report Format: Html

Scope

Time set

Starting Time	...
Ending Time	...

Report Mode

This table function displays the ASH Spot report

AWR Manager를 이용한 악성 쿼리 찾기

- Report를 생성하기 위해서 스냅샷 번호 구간을 선택하거나 베이스 라인 선택 필요
- Baseline Manager를 통하여 새로운 베이스라인을 등록하거나 삭제 가능

The screenshot displays the Oracle AWR Manager application window. The main interface is divided into several panes: Report Manager, Snapshot Manager, and Overall Snapshot. The Report Manager pane shows configuration options for report mode and scope. The Snapshot Manager pane shows collection settings and a list of snapshots. The Overall Snapshot pane provides summary statistics. A 'Baseline Manager' dialog box is open in the foreground, displaying a table of existing baselines and an 'Add Baseline' sub-dialog box.

BASLINE_ID	BASLINE_NAME	START_SNAP_ID	END_SNAP_ID	START_SNAP_TIME	END_SNAP_TIME
13	a	389	392	2010/08/29 15:00:33	2010/08/29 16:00:29
12	asd	125	135	2010/08/16 23:00:48	2010/08/17 09:00:22

The 'Add Baseline' dialog box contains the following fields:

- Baseline Name:
- Starting Snapshot: ...
- Ending Snapshot: ...

Buttons: OK, Cancel, Close

AWR Manager를 이용한 악성 쿼리 찾기

- AWR SQL Report 에서 제공하는 항목
 - Snapshot Period Summary, SQL Summary, SQL ID, Plan Statistics,
 - Execution Plan, Full SQL Text

The screenshot displays the Oracle AWR Manager interface with several report views open:

- WORKLOAD REPOSITORY SQL Report - Snapshot Period Summary:** Shows a table with columns: DB Name, DB Id, Instance, Inst num, Release, RAC, Host. Below it, a table shows Snap Id, Snap Time, Sessions, and Cursors/Session.
- SQL Summary:** Shows a table with columns: SQL Id, Elapsed Time (ms), Module, Action, SQL. Below it, a table shows Plan Hash Value, Total Elapsed Time(ms), Executions, 1st Capture Snap ID, and Last Capture Snap ID.
- Plan Statistics:** Shows a table with columns: Stat Name, Statement Total, Per Execution, % Snap Total. Below it, a table shows Plan Statistics with columns: Id, Operation, Name, Rows, Bytes, Cost (%CPU), Time.
- Full SQL Text:** Shows the SQL text for a specific query.

Health Check 이용한 악성 쿼리 찾기

- Access Type 항목을 체크하고 Health Check 실행
- Orange 4.0 과는 달리 메인 화면에서는 SQL 문장을 제외하고 나머지 정보만 제공
- 표 상단의 별도의 Html 링크를 클릭한 경우에 Full Text Query를 제공

Top-20 SQL(by Buffer Gets)

USERNAME	SHARABLE_MEM	EXECUTIONS	PARSE_CALLS	DISK_READS	BUFFER_GETS	RowsPerExec	Elapsed Time (Sec)	OPTIMIZER_MODE	SQL_TEXT
YHKIM	182030	34	34	1	1190513	1	0.12	ALL_ROWS	select a.segment_space_management from dba_tablespaces a, dba_segments b where a.tablespace_name = b.tablespace_name and b.owner = 'owner' and b.segment_type = 'seg_type' and b.partition_name = 'part_name'
YHKIM	526815	24	24	9871	451337	419	0.74	ALL_ROWS	select /*+ ordered use_hash(a) */ a.owner, a.segment_name, a.partition_name, a.segment_type, b.extents, b.blocks, b.bytes, a.initial_extent, a.next_extent, a.min_extents, a.max_extents, a.pct_increase, a.relative_fno, a.header_block, a.tablespace_id, 5 file_id from sys.sys_dba_segs a, (select owner, segment_name, segment_type, nvl(partition_name, 1) partition_name, count(segment_name) as extents, sum(bytes), sum(blocks) blocks from sys.dba_extents where tablespace_name = 'tsname' and file_id = 'fid1' group by owner, segment_name, segment_type, partition_name) b where a.owner = b.owner and a.segment_name = b.segment_name and a.segment_type = b.segment_type and nvl(a.partition_name, 1) = b.partition_name and a.tablespace_name = 'tsname union all select ' owner, 'FREE_SPACE' segment_name, ' ' partition_name, 'FREE_SPACE' segment_type, tot_bc, tot_block, tot_block_size from v\$datafile where file#=fid1) as bytes, TO_NUMBER('') initial_extent, TO_NUMBER('') next_extent, TO_NUMBER('') min_extents, TO_NUMBER('') max_extents, TO_NUMBER('') pct_increase, TO_NUMBER('') relative_fno, TO_NUMBER('') header_block, TO_NUMBER('') tablespace_id, 5 file_id from (select sum(bc) as tot_bc, sum(sum_block) as tot_block from (select count (block_id) as bc, sum(dtp.blocks) as sum_block from sys.dba_dmt_free_space dtp where file_id=fid1 union all select count (block_id) as bc, sum(blocks) as sum_block from sys.dba_inmemory_space where file_id=fid1 union all select count (block#) as bc, sum(space) as sum_block from sys.recyclebin\$ where file#=fid1)) order by 1
YHKIM	137343	24	24	0	254435	7	1.97	ALL_ROWS	
YHKIM	67433	23	23	0	189666	1030	1.86	ALL_ROWS	
YHKIM	149394	24	24	0	118760	7	0.04	ALL_ROWS	
SYSMAN	12660	919							
SYSMAN	12673	292							
SYSMAN	25018	395							
SYSMAN	18068	9094							
SYSMAN	91195	3257							
SYSMAN	22028	140							
SYSMAN	91631	2294							
CSLEE	153322	2							
SYSMAN	17089	2565							
SYSMAN	26529	891							
SYSMAN	12686	110							
SYSMAN	22297	13695							
SYSMAN	18084	104							
SYSMAN	18116	727							

Top-20 SQL (by Buffer Gets)

Thursday September 09 2010 14:38:24

HASH_VALUE	ADDRESS	SQL_TEXT
1637746790	2EDB2A0C	select a.segment_space_management from dba_tablespaces a, dba_segments b where a.tablespace_name = b.tablespace_name and b.owner = 'owner' and b.segment_type = 'seg_type' and b.partition_name = 'part_name'
1909424644	2A4E4298	select /*+ ordered use_hash(a) */ a.owner, a.segment_name, a.partition_name, a.segment_type, b.extents, b.blocks, b.bytes, a.initial_extent, a.next_extent, a.min_extents, a.max_extents, a.pct_increase, a.relative_fno, a.header_block, a.tablespace_id, 5 file_id from sys.sys_dba_segs a, (select owner, segment_name, segment_type, nvl(partition_name, 1) partition_name, count(segment_name) as extents, sum(bytes), sum(blocks) blocks from sys.dba_extents where tablespace_name = 'tsname' and file_id = 'fid1' group by owner, segment_name, segment_type, partition_name) b where a.owner = b.owner and a.segment_name = b.segment_name and a.segment_type = b.segment_type and nvl(a.partition_name, 1) = b.partition_name and a.tablespace_name = 'tsname union all select ' owner, 'FREE_SPACE' segment_name, ' ' partition_name, 'FREE_SPACE' segment_type, tot_bc, tot_block, tot_block_size from v\$datafile where file#=fid1) as bytes, TO_NUMBER('') initial_extent, TO_NUMBER('') next_extent, TO_NUMBER('') min_extents, TO_NUMBER('') max_extents, TO_NUMBER('') pct_increase, TO_NUMBER('') relative_fno, TO_NUMBER('') header_block, TO_NUMBER('') tablespace_id, 5 file_id from (select sum(bc) as tot_bc, sum(sum_block) as tot_block from (select count (block_id) as bc, sum(dtp.blocks) as sum_block from sys.dba_dmt_free_space dtp where file_id=fid1 union all select count (block_id) as bc, sum(blocks) as sum_block from sys.dba_inmemory_space where file_id=fid1 union all select count (block#) as bc, sum(space) as sum_block from sys.recyclebin\$ where file#=fid1)) order by 1



Session Monitor 활용하기

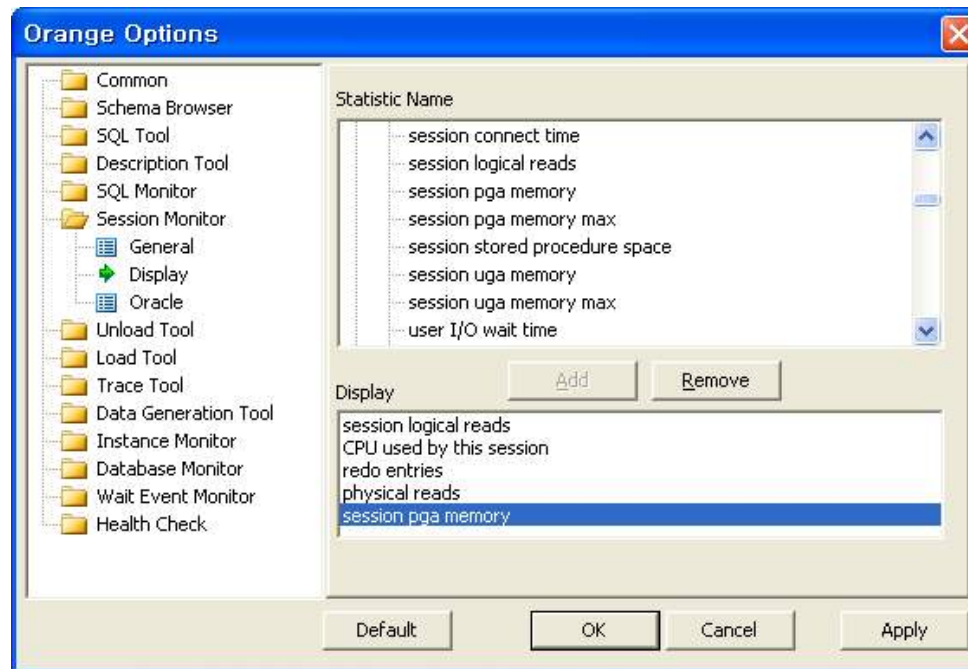
- 새로운 세션 모니터 기능
 - 칼럼 순서 변경 가능
 - 특정 칼럼을 기준으로 그룹핑 가능
 - 여러 개 칼럼으로 정렬 가능
 - 여러 개 세션 선택 가능. 동시에 여러 세션을 kill 하거나 trace on/off 가능
 - Coordinate Process와 Slave Process의 관계를 트리 형태로 제공.
 - 세션 선택 시 하단 창에서 상세 정보 제공
 - 서버 버전이 10g 이상인 경우 Active Session History 조회 가능

Inst#	Server#	SID	Serial#	SPID	User Name	O/S User	Event	Command	Machine	Module	Action	Prt.	Loc.	Ses...
10	135	1272228	SCOTT	CSLEE-374...	SOL-Net message from cli...	UNKNOWN	WORKGR...	Orange for ORACLE DBA	5.6.2 (Eu...	Or...	201...			3670
10	137	52432	SYSTEM	Streams AQ: qmn slave idl...	UNKNOWN	WORKGR...	STREAMS	OMON S...	OR...	201...				88
10	142	12820	SYS	CSLEE-374...	Streams AQ: waiting for f...	UNKNOWN	CSLEE-37...	STREAMS	OMON S...	Or...	201...			12
10	148	453020	SYSTEM	Space Manager: slave idle...	UNKNOWN	CSLEE-37...	KTSJ	KTSJ Co...	OR...	201...				0
10	149	12384	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...	KTSJ	KTSJ Co...	OR...	201...				0
10	152	32280	SYSTEM	lbar timer	UNKNOWN	CSLEE-37...			OR...	201...				2182
10	154	132932	SYSTEM	Streams AQ: qmn coordin...	UNKNOWN	CSLEE-37...	STREAMS	OMON ...	OR...	201...				6
10	155	11248	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				0
10	156	11929	SYSTEM	DIAG idle wait	UNKNOWN	CSLEE-37...			OR...	201...				0
10	157	11254	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				168
10	158	11900	SYSTEM	VKTM Logical Idle Wait	UNKNOWN	CSLEE-37...			OR...	201...				0
10	160	11454	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				268
10	161	11316	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				11409
10	162	11296	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				132
10	163	11288	SYSTEM	smon timer	UNKNOWN	CSLEE-37...			OR...	201...				27513
10	164	11308	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				0
10	165	11180	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				0
10	166	11272	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				0
10	167	11280	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				0
10	167	11280	SYSTEM	rdcms ipc message	UNKNOWN	CSLEE-37...			OR...	201...				0
10	168	31278	SYSTEM	DIAG idle wait	UNKNOWN	CSLEE-37...			OR...	201...				0

```
1 SELECT /*+ ordered */sql_text,
2         piece
3 FROM v$session s,
4       v$sql_text_with_newlines t
5 WHERE s.sid = t.sid
6       AND s.serial# = t.serial
7       AND s.sql_address = t.address
8       AND s.sql_hash_value = t.hash_value
9       AND s.sql_hash_value <> 0
10 UNION ALL
11 SELECT /*+ ordered */sql_text,
12        piece
13 FROM v$session s,
14       v$sql_text_with_newlines t
15 WHERE s.sid = t.sid
16       AND s.serial# = t.serial
17       AND s.prev_sql_addr = t.address
18       AND s.prev_hash_value = t.hash_value
19       AND s.sql_hash_value = 0
```

Session Monitor 활용하기

- 옵션에서 Session Statistics 정보를 최대 5개 까지 추가 가능
 - Session logical reads – 메모리에서 데이터를 읽고 있는 세션
 - CPU used by this session – CPU 사용량이 많은 세션
 - Redo entries – DML 작업을 하고 있는 세션
 - Physical reads – 디스크에서 데이터를 많이 읽고 있는 세션
 - Session pga memory – pga 메모리를 많이 사용하고 있는 세션 (ex. 정렬)





➤ 오렌지 2010 의 새로운 기능을 확인해 보시기 바랍니다.

감사합니다.