



## All the Free Stuff from Oracle Corp

This paper is associated with a presentation of the same name. I will cover the various cool explain plan technologies that are available with the database. I will illustrate the trace tools (the obvious TKProf and the new Trace Analyzer) and I will conclude with the new SQL Developer (formerly known as Project Raptor), Oracle's new SQL & PL/SQL development tool.

### Explain Plan Free Tools

Oracle has always supplied tools that expose the execution plan or explain plans for SQL statements. All explain plan utilities use the PLAN\_TABLE that is found in the <oracle home>/rdbms/admin/utlxplan.sql file. I recommend that this file gets run for your home directory for each database type you are likely to use.

To use a couple of these tools, you need to populate the PLAN\_TABLE. Use the 'explain plan for' syntax as illustrated below. The Statement\_ID is not necessary but used in my show\_plan.sql script.

```
Oracle SQL*Plus
File Edit Search Options Help

DHOTKAXP:ora9r2:SCOTT>explain plan set statement_id = 'SCOTT1' for
 2 select ename
 3 from emp
 4 where deptno in (select deptno from dept);

Explained.

DHOTKAXP:ora9r2:SCOTT>start show_plan SCOTT1
old  4: where statement_id = '&1'
new  4: where statement_id = 'SCOTT1'

-----
Cost   ID P_ID Access                               Access Path      Object Name
-----
 2     0      SELECT STATEMENT
 2     1     0   NESTED LOOPS
 2     2     1   TABLE ACCESS                               FULL            EMP
        3     1   INDEX                                       UNIQUE SCAN     PK_DEPT

old  1: delete from plan_table where statement_id = '&1'
new  1: delete from plan_table where statement_id = 'SCOTT1'

4 rows deleted.

DHOTKAXP:ora9r2:SCOTT>
```

The dbms\_xplan.display package exposes the where clause predicates associated with each line of the explain plan output. This is only available using the appropriate PLAN\_TABLE and is only available starting with Oracle v9.2. This feature only works for the cost-based optimizer.

```

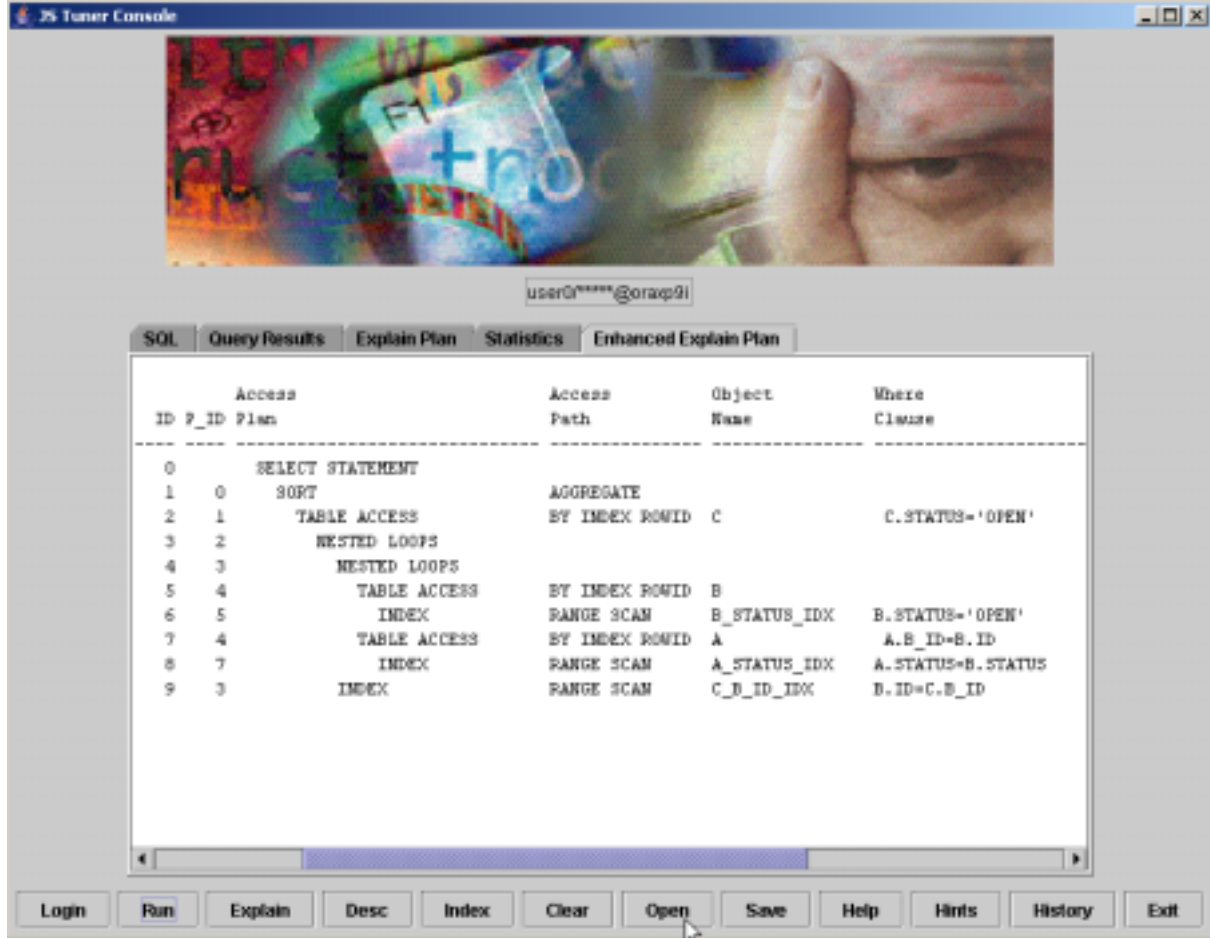
DHOTKAXP:ora9r2:USER0>1
 1* select * from table(dbms_xplan.display)
DHOTKAXP:ora9r2:USER0>/

PLAN_TABLE_OUTPUT
-----
| Id | Operation                                | Name          | Rows  | Bytes | Cost |
-----|-----|-----|-----|-----|-----|
| 0  | SELECT STATEMENT                        |               | 1     | 18    | 8    |
| 1  | SORT AGGREGATE                          |               | 1     | 18    | 8    |
|* 2  | HASH JOIN                                |               | 10000 | 175K  | 8    |
|* 3  | HASH JOIN                                |               | 1000  | 12000 | 5    |
| 4  | TABLE ACCESS BY INDEX ROWID            | B             | 100   | 600   | 2    |
|* 5  | INDEX RANGE SCAN                         | B_STATUS_IDX | 100   |       | 1    |
|* 6  | TABLE ACCESS FULL                       | A             | 1000  | 6000  | 2    |

PLAN_TABLE_OUTPUT
-----
|* 7  | TABLE ACCESS FULL                       | C             | 1000  | 6000  | 2    |

Predicate Information (identified by operation id):
-----
 2 - access("B"."ID"="C"."B_ID")
 3 - access("A"."B_ID"="B"."ID")
 5 - access("B"."STATUS"='OPEN')
 6 - filter("A"."STATUS"='OPEN')
 7 - filter("C"."STATUS"='OPEN')

```



JSTuner has a number of cool features for tuning SELECT SQL statements. It can show the simple explain plan from the first example. It can show both table and index columns, useful index statistical information, has a series of available hints for cut/paste, and a SQL history. This tool installs with a snap and self-configures itself upon first execution. This tool simply front-ends character-mode SQL\*Plus, its scripts and undocumented trace options that makes it special.

The feature displayed uses an Oracle Trace to capture the where clause predicates. This works equally well for both rule and cost-based optimizers. This works at least as far back as Oracle 8i.

This tool is available for download from my website.

## Oracle Trace Free Tools

Oracle has supplied a vehicle for capturing all SQL statements since about Oracle7. This might have been available in Oracle 6 and was un-documented. This is called a 10047 trace (pronounced 10-47). There are many options available such as tripping it on for specific sessions, capturing extended information such as bind variable and Oracle events, etc.

```
Alter session set max_dump_file_size=unlimited;
Alter session set timed_statistics=true;
Alter session set tracefile_identifier='HOTKA';
Alter session set events '10046 trace name context forever, level 12';
Select 'Hello World: Today is ' || sysdate from dual; (run your application here)
Alter session set events '10046 trace name context off'
```

The above example is how I trip this trace on to collect all SQL statements with bind variables and event information.

This trace file is created in the Oracle UDUMP directory. Using the tracefile\_identifier makes it easy to spot your SQL trace file.

TKProf has traditionally used to format a SQL Trace file, order the SQL statements by a variety of parameters, allowing you to drill into the collected SQL statements looking for what ever issue you can get to trickle to the top. TKProf can simply display the contents of the trace file, or, order the SQL statements by a variety of collected statistical options. I typically use PRINT = 10 to display only the first 10 SQL statements it finds. I use the SYS=NO to skip any Oracle-generated SQL. I use the SORT=<options> to find the SQL of interest to me (ie: total elapsed time, top disk i/o, etc).

```
TKPROF <tracefile name> <output file> print=10 sys=no sort=<sort options>
```

Run TKPROF without any options to see all the available options.

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Trace file: ora9r2\_ora\_Level12.trc

Sort options: default

```

*****
count      = number of times OCI procedure was executed
cpu        = cpu time in seconds executing
elapsed    = elapsed time in seconds executing
disk       = number of physical reads of buffers from disk
query      = number of buffers gotten for consistent read
current    = number of buffers gotten in current mode (usually for update)
rows       = number of rows processed by the fetch or execute call
*****

```

alter session set events '10046 trace name context forever, level 12'

call	count	cpu	elapsed	disk	query	current	rows
Parse	0	0.00	0.00	0	0	0	0
Execute	1	0.00	0.03	0	0	0	0
Fetch	0	0.00	0.00	0	0	0	0
total	1	0.00	0.03	0	0	0	0

```

Misses in library cache during parse: 0
Misses in library cache during execute: 1
Optimizer goal: CHOOSE
Parsing user id: 88

```

Elapsed times include waiting on following events:

Event waited on	Times Waited	Max. Wait	Total Waited
SQL*Net message to client	1	0.00	0.00
SQL*Net message from client	1	15.93	15.93

\*\*\*\*\*

This is what the sample output of TKProf displays. This was displayed from a SQL trace that collected both bind variables and Oracle events.

It took years but Oracle Corp finally came out with a better trace file analysis tool: Trace Analyzer. This tool looks at SQL trace files and produces a nice hyperlinked report. This report is a cross between the detailed statistical information of Stats Pack and a nice-enhanced TKProf output. The only downside is that there are no sort options, it only displays what it thinks is the top 20 poorly performing SQL statements according to CPU time, Elapsed and Wait times.

Trace Analyzer is available via MetaLink. Search on Trace Analyzer and download trca.zip. Pay particular attention to the Instructions.txt file, especially when installing the required package.

Trace Analyzer is executed thru SQL\*Plus, passing the script the SQL tracefile name found in the UDUMP directory on the server SQL\*Plus is logged into. The output HTML file is created in both the UDUMP directory and the startup directory of SQL\*Plus.

```

Sqlplus userid/pwd@host @trcanlzl <trace file>
SQL> start trcanlzl <trace file>

```

Trace Analyzer Report 560 6 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Print Mail Stop

Address [C:\TEMP\trcanlzt\\_Oraxp10g.html](C:\TEMP\trcanlzt_Oraxp10g.html)

Links Toshiba Access Customize Links Free Hotmail RealPlayer Windows Windows Market

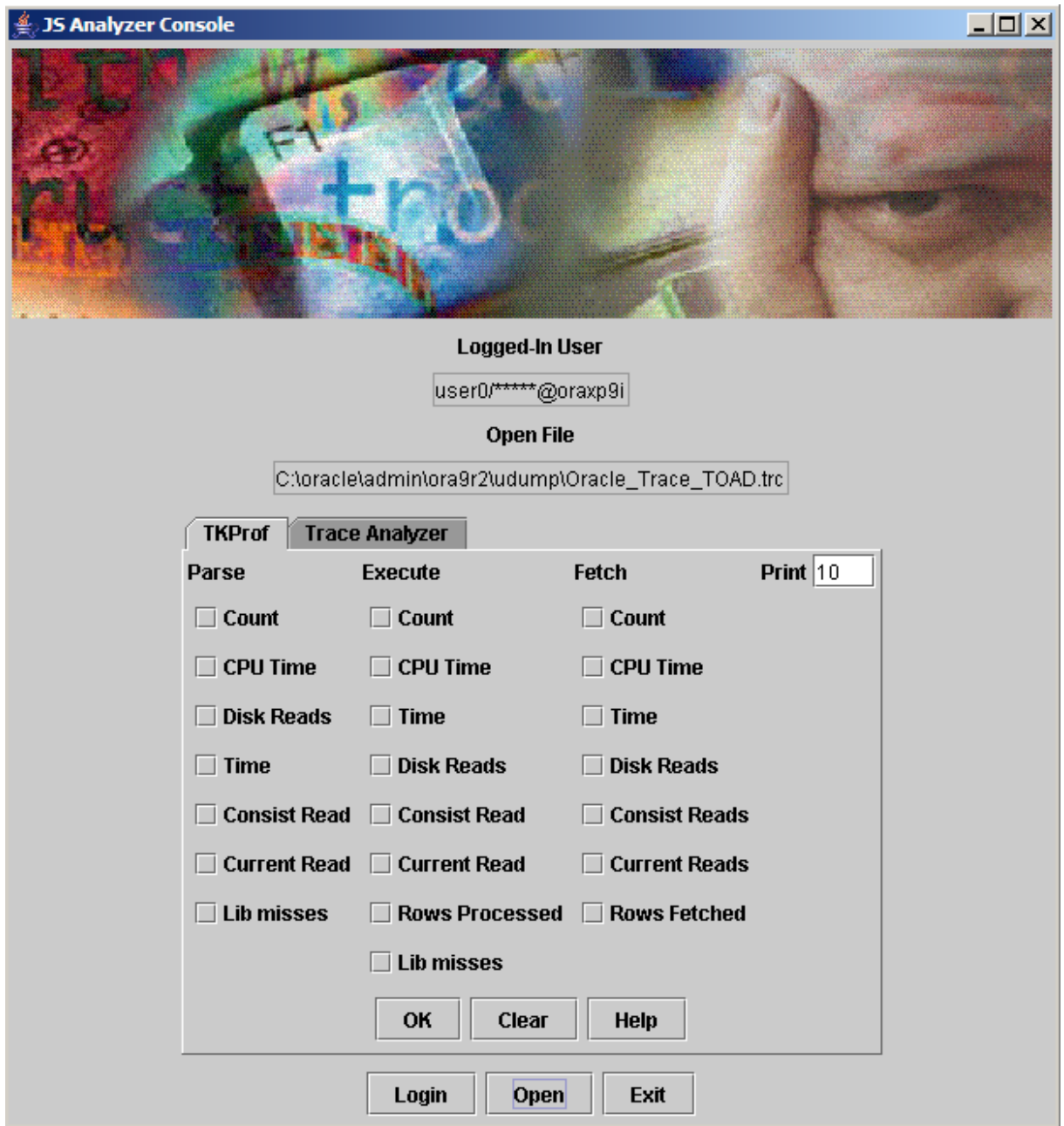
To help protect your security, Internet Explorer has restricted this file from showing active content that could access y

## Trace Analyzer Report: 560 6

- [Trace Identification](#)
- [Trace Header](#)
- [SQL Statements Count](#)
- [Time Summary](#)
- [User Non-Recursive Totals](#)
- [User Recursive Totals](#)
- [Internal Totals](#)
- [Top 20 SQL Statements](#)
- [List of 35 Unique SQL Statements](#)
- [Gaps of no Trace Activity \(4\)](#)
- [Non-Default Initialization Parameters](#)
- [SQL Statements Details](#)

### Trace Identification

Trace Filename:	oraxp10g_ora_560_level12.trc
UDUMP Directory:	C:\ORACLE\PRODUCT\10.1.0\ADMIN\ORAXP10G\UDUMP
Instance Name:	oraxp10g
Host Name:	Unknown
RDBMS Version:	10.1.0.2.0
Trace Size (bytes):	83692
Trace Lines:	1347
Trace Start Time:	24-JAN-06 08:53:05
Trace End Time:	24-JAN-06 08:56:31
Trace Duration (tot secs):	205.500
4 Gaps (tot secs):	131.149
Analyzed on Same Instance?:	Y
Trace Truncated?:	N
Report Generated by User:	USER0



JS Analyzer makes easy work of both TKProf (with its cumbersome command-line syntax) and Trace Analyzer running a package passing it a lengthy-error-prone trace file name.

Simply point JS Analyzer at the SQL trace file using your mouse and select your options for running TKProf from the TKProf tab. Clicking on 'OK' will run TKProf and open the output file in an editor (like Notepad or TextPad) of your choice. The Trace Analyzer tab allows for you to run the Trace Analyzer package and open the output in your computers default web browser. Both tools are executed behind the scenes via their command-line roots.

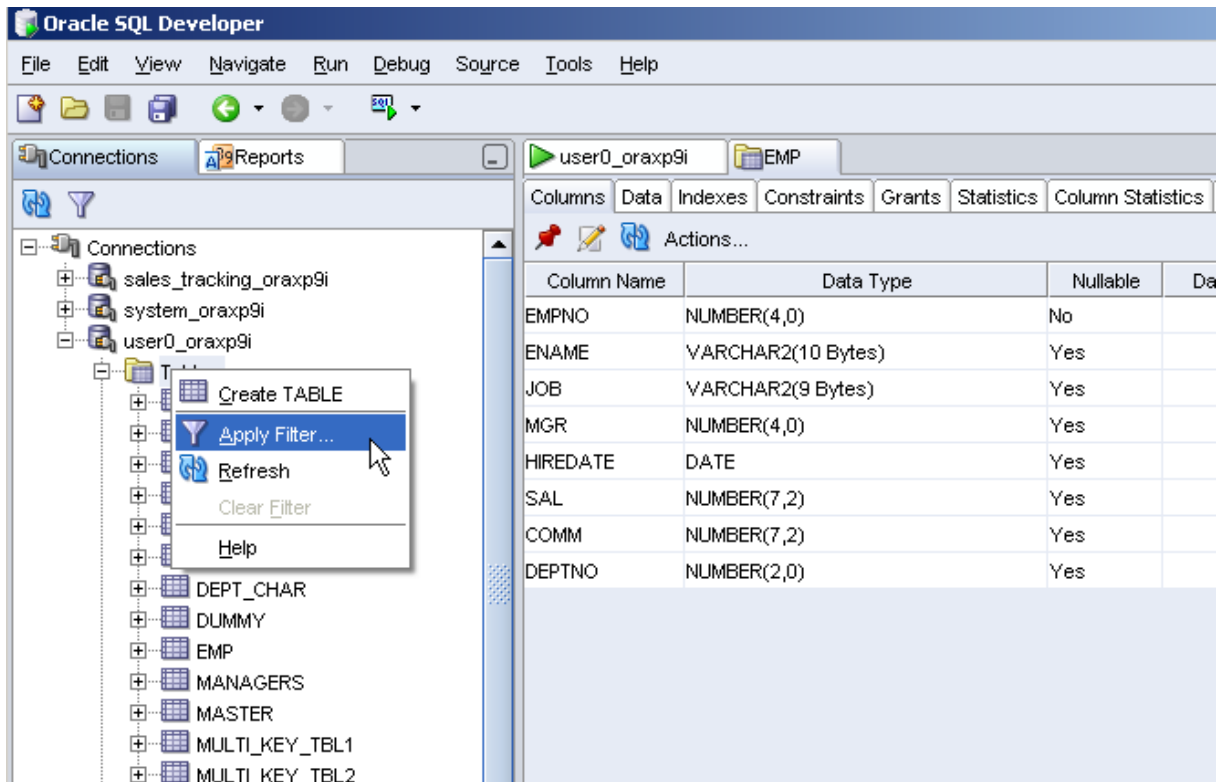
## **Oracle Development Free Tools**

Oracle Corp has released a cool new tool for table/object research, application install/setup/maintenance, SQL, SQL script, and PL/SQL development.

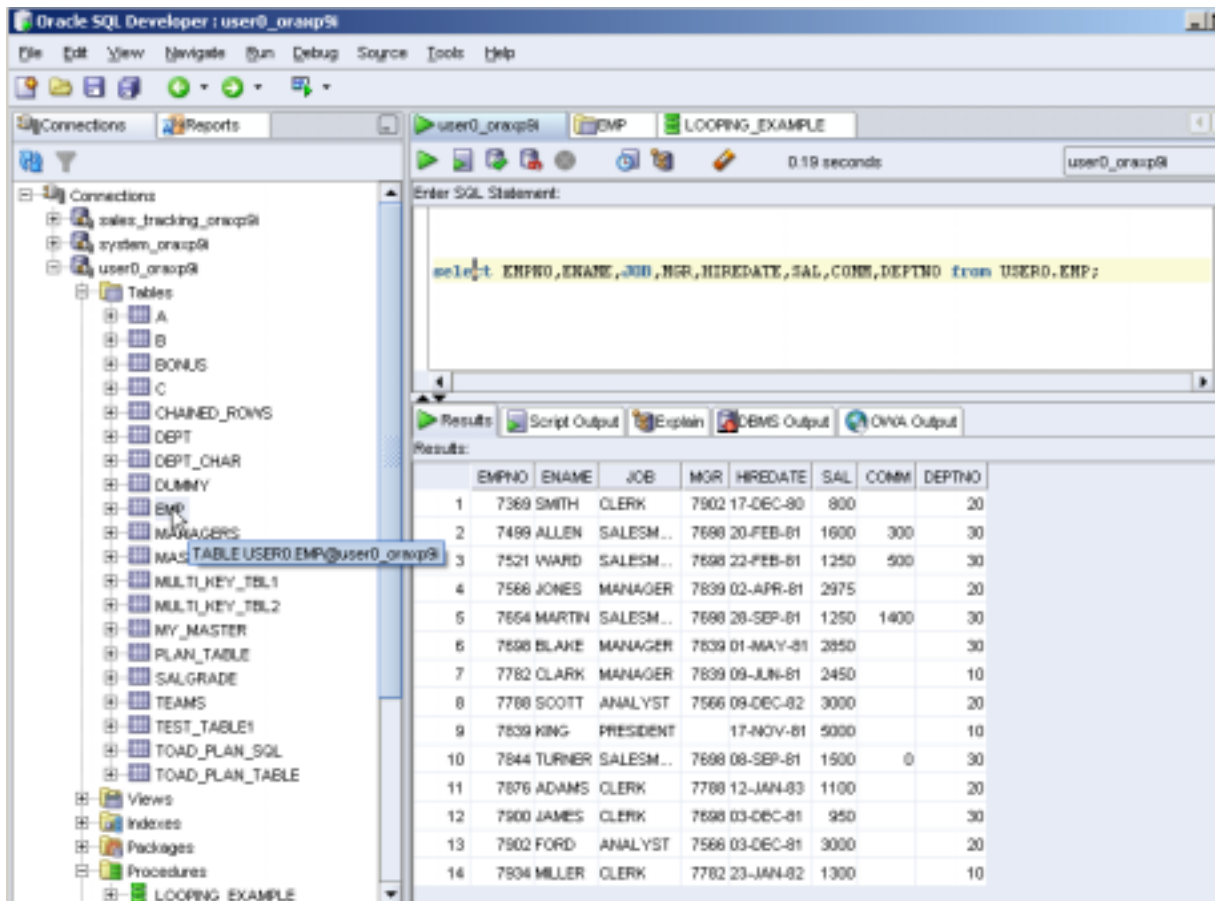
This tool is darn complete for a free tool!

SQL Developer is downloaded in 2 forms: with (50mb) or without (19mb) its required JDK runtime environment. Installation is as simple as unzipping the compressed download file.

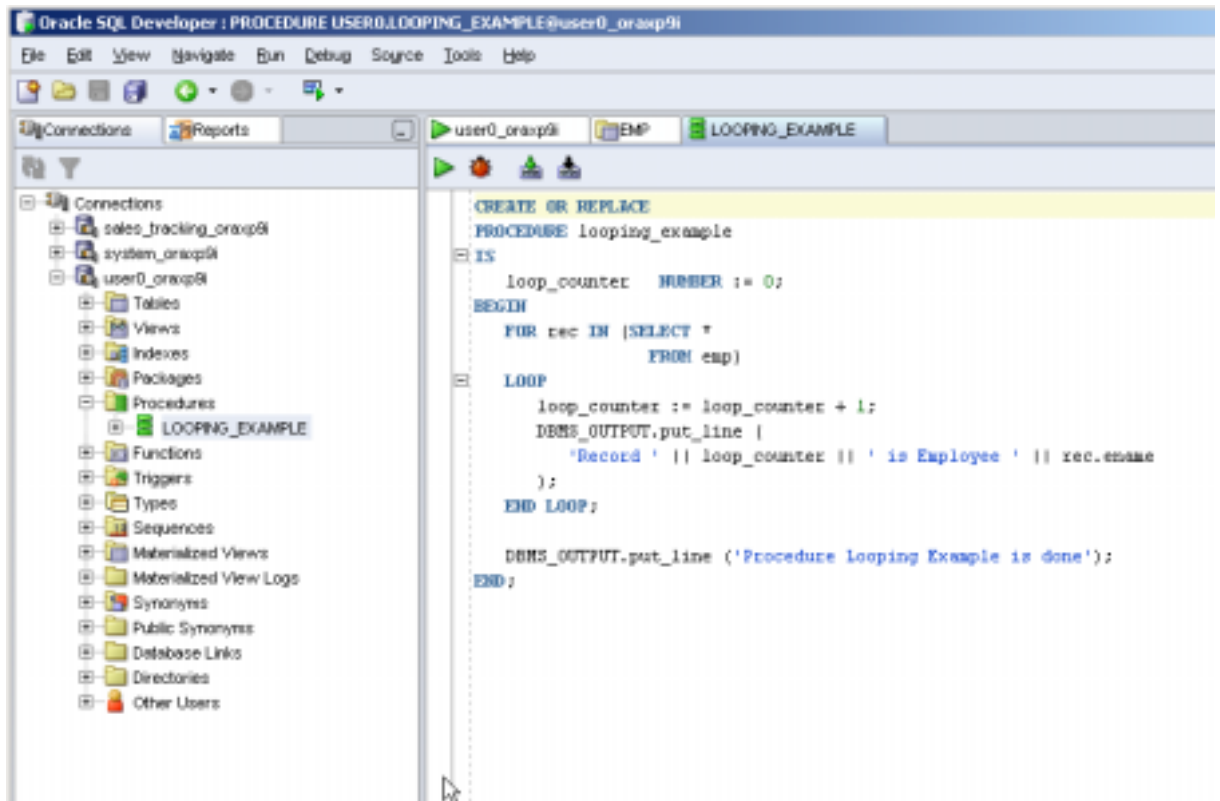
Download SQL Developer from [ww.oracle.com/technology/software/products/sql/index.html](http://ww.oracle.com/technology/software/products/sql/index.html) .



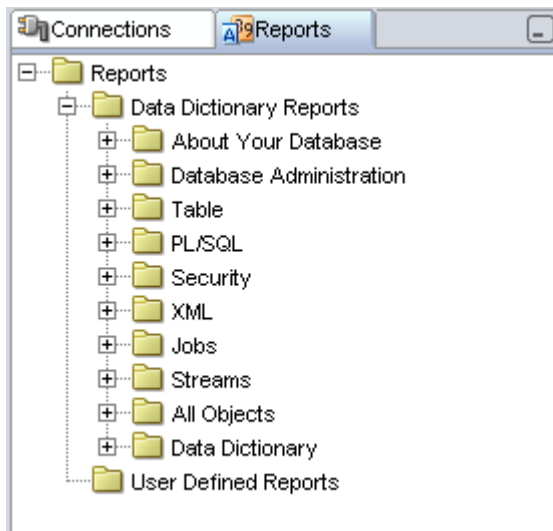
SQL Developer has a navigator touch and feel. Once your connection is established (with/without password protection). Click on a table name and you can see the columns, indexes, data (including maintaining data, making scripts of the data, exporting to Excel, and more). Each object type has similar functionality. With the correct permissions, you can create and maintain an entire application environment (users, tables, objects, sequences, packages, procedures, functions, links, synonyms, etc).



SQL is easily created by dragging/dropping the table name into the SQL window. This will create a complete SQL statement! You can drag and drop column names from the navigator as well. SQL Developer has Code Snippets that are full of SQL, PL/SQL, functions, and hint templates. SQL Developer has a nice explain plan as well.



SQL Developer has a robust PL/SQL package/procedure/function development environment. This environment utilizes the same Code Snippets, compiles, and debugs any PL/SQL code.



SQL Developer has a variety of reports that shows a variety of information from Tablespace available space, to extent utilization, and more. You can even make reports of your own favorite SQL scripts.

See my article in the March 2006 of Oracle Professional for a nice walkthrough of this new tool.



## Other Free Tools NOT covered in this article:

Other free tools/resources not covered in this presentation are UTLBSTAT, UTLESTAT, Statspack, and the PL/SQL profiler.

Let me know if you know of other free tools from Oracle Corp!

## Summary

This paper and its associated presentation highlights free tuning and development tools from Oracle Corp. I started out showing the latest features available from an explain plan point of view. I illustrated the existing and new SQL Trace Analyzer tools. I introduced Oracle's latest free application development tool: SQL Developer.

Dan Hotka is a Training Specialist who has over 28 years in the computer industry and over 23 years of experience with Oracle products. He is an internationally recognized Oracle expert with Oracle experience dating back to the Oracle V4.0 days. Dan's latest book is the SQL Developer Handbook by Oracle Press. He is also the author of Oracle9i Development By Example and Oracle8i from Scratch by Que and has co-authored 7 other popular books including the Database Oracle10g Linux Administration by Oracle Press. He is frequently published in Oracle trade journals, and regularly speaks at Oracle conferences and user groups around the world. Visit his website at [www.DanHotka.com](http://www.DanHotka.com). Dan can be reached at [dhotka@earthlink.net](mailto:dhotka@earthlink.net).

Dan Hotka - Author/Instructor/Expert  
[www.DanHotka.com](http://www.DanHotka.com)  
[DHotka@Earthlink.net](mailto:DHotka@Earthlink.net)  
515 279-3361