April 2006 Volume II Issue 2

ORAtips Journal

ORAtips

On Development

TOAD: Tips and Techniques

By Dan Hotka

*Editor's Note: Every Oracle Developer and Database Administrator knows the power of TOAD. It is both freeware and a licensed product, and is an excellent alternative to SQL*Plus for developing Oracle scripts and reviewing database schema information. But unlike the cliché, "you cannot teach an old dog new tricks", first time ORAtips contributor and TOAD instructor Dan Hotka has put together a few of his favorite things about TOAD for both the novice and experienced TOAD user. The result is a tip-packed article of new tricks to save keystrokes, reduce time spent searching for syntax errors, and make more efficient use of the symbolic debugger.*

Introduction

TOAD is undoubtedly the defacto standard development tool for the Oracle database. TOAD came from the shareware days and is sold by Quest Software. A free trial version is available at **www.TOADsoft.com and www.Quest.com.**

What is TOAD? TOAD is a Windows-based graphical tool that allows for Oracle SQL and script execution, and viewing object information, relationships, scripts, and data. Data can also be easily viewed, saved in various formats (including MS Excel, XML, and insert scripts), manipulated, and more. TOAD is very powerful in the Oracle PL/SQL arena, its true strength.

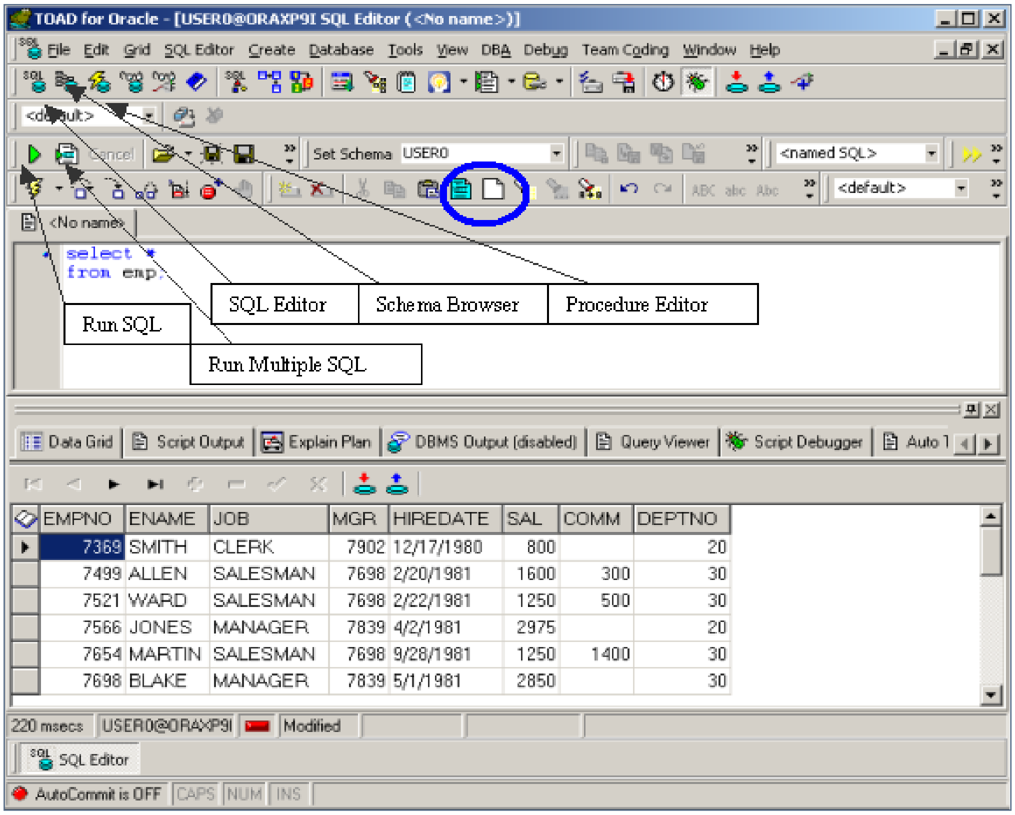
TOAD is a Windows-based graphical tool that allows for Oracle SQL and script execution.

This article will illustrate some of the more popular uses of TOAD. It will detail some of the keystroke saving features such as auto replace and code templates. There are several ways to obtain information and at least three ways to do most things. For example, menu selection, hot-key sequence, right-mouse click, and assigned function key all have similar purposes. This article will focus on the three main TOAD utilities: the SQL Editor, the Schema Browser, and the Procedure Editor.

SQL Editor

Figure 1 illustrates the main TOAD interface, the SQL Editor. The SQL Editor allows you to easily enter and execute Oracle SQL or unnamed PL/SQL blocks. The results appear in the lower half of the screen.

F9 or the green arrow-shaped button allows you to run a single SQL statement or the SQL statement that the mouse cursor is currently on. This button also executes SQL that is high-

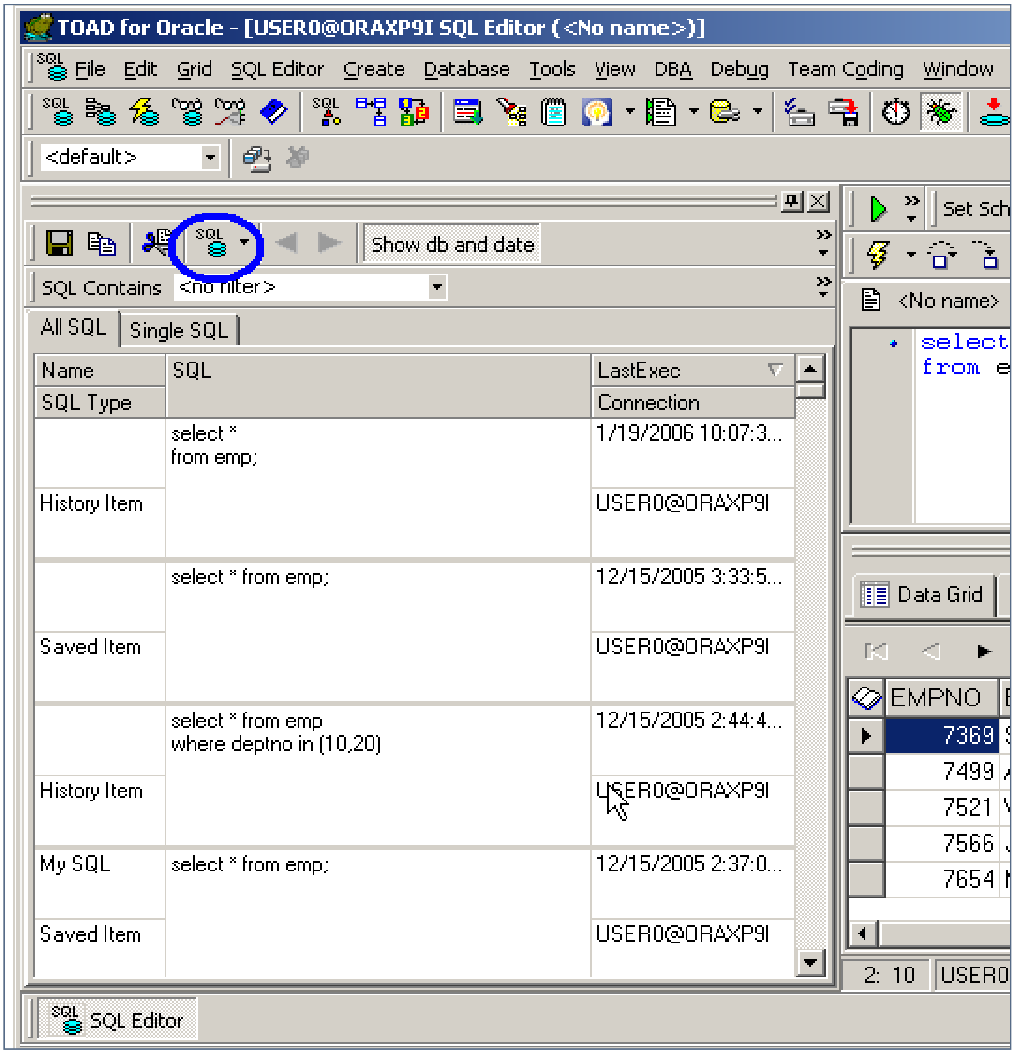


```
select *
from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30

Figure 1: TOAD SQL Editor

ORAtips
On Development


TOAD for Oracle - [USER0@ORAXP9I SQL Editor (<No name>)]
File Edit Grid SQL Editor Create Database Tools View DBA Debug Team Coding Window
<default>
Show db and date
SQL Contains <no filter>
All SQL
Single SQL
Set Sch
<No name>
Data Grid
SQL Editor
2: 10 USER0

Name	SQL	LastExec
SQL Type		Connection
	select * from emp;	1/19/2006 10:07:3...
History Item		USER0@ORAXP9I
	select * from emp;	12/15/2005 3:33:5...
Saved Item		USER0@ORAXP9I
	select * from emp where deptno in (10,20)	12/15/2005 2:44:4...
History Item		USER0@ORAXP9I
My SQL	select * from emp;	12/15/2005 2:37:0...
Saved Item		USER0@ORAXP9I

```
select
from e
```

EMPNO
7369
7499
7521
7566
7654

Figure 2: TOAD SQL Recall

lighted from inside a PL/SQL routine. The output from this button appears in the DATA GRID tab in the lower area. There will also be an Explain Plan generated for this SQL, which you can view by clicking the EXPLAIN PLAN tab.

The button next to the green arrow-shaped button runs all of the SQL that is in the SQL area. The output from this execution appears in the SCRIPT OUTPUT tab.

Multiple connections to the same or different databases can be maintained. And, multiple TOAD windows can be open; the only limit is the amount of memory on your workstation! It is common practice to have connections to the test, development, and even production environments open at the same time. Under the Window menu item, it is easy to tile or cascade all open windows.

The clear button, encased in the blue circle in Figure 1, clears all the SQL in the window.

Pressing F8 brings up the TOAD SQL History or SQL Recall, as shown in Figure 2. This is one of the most popular features of TOAD because it makes it easy to bring up SQL worked on previously or, when tuning SQL 5 iterations ago. There are options to control how many SQL statements get saved. TOAD defaults to the last 100 SQL statements executed. Simply highlight the SQL and press the SQL button (encased in the blue circle in Figure 2) to load the SQL into the SQL Editor. This is also a dockable window; notice the push-pin button. This provides an auto-hide type function but still allows easy access to the SQL history.

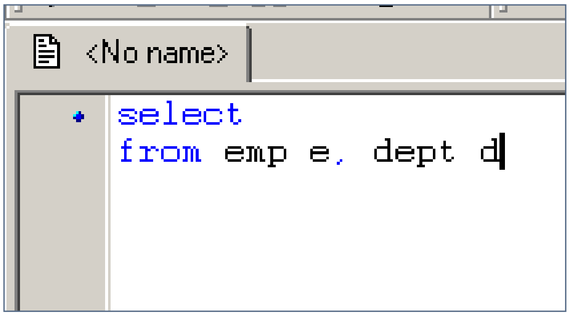
ALT-Up Arrow and ALT-Down Arrow also access SQL. Recall from the SQL Editor window, and it will walk you up or down through recently submitted SQL.

There are a lot of TOAD features that can eliminate extra keystrokes when coding. One of the more popular features is the embedded alias processing. Using table aliases is a good practice to get into, and in TOAD, it greatly aids in the quick development of SQL statements.

As an example, starting with the Oracle SQL illustrated in Figure 3, add a SELECT and FROM clause and give each table in the FROM clause an alias.

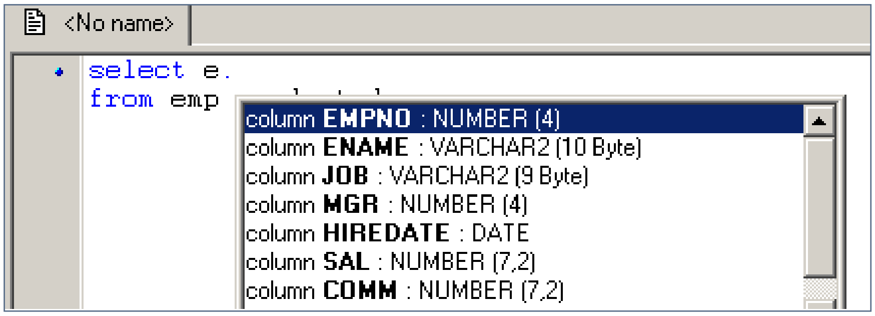
Now, navigate to other parts of the SQL statement and enter the alias followed by a“.” and wait one second.

A column popup, as shown in Figure 4, will appear. Double click on any of the columns, and they will be pasted into your SQL session.



```
select
from emp e, dept d
```

Figure 3: TOAD Enabling Alias Processing



```
select e.
from emp
```

```
column EMPNO : NUMBER (4)
column ENAME : VARCHAR2 (10 Byte)
column JOB : VARCHAR2 (9 Byte)
column MGR : NUMBER (4)
column HIREDATE : DATE
column SAL : NUMBER (7,2)
column COMM : NUMBER (7,2)
```

Figure 4: TOAD Alias Pop-Up

You can highlight several columns at once and hit return, and they will all be pasted into your SQL session as illustrated in Figure 5, complete with aliases! This saves a lot of typing and research, and the impending syntax errors because you did not spell the column name correctly.

This feature works anywhere in TOAD where you can work with SQL including SQL Editor and Procedure Editor.

The DESCRIBE window is another very popular feature. Highlight or just place the cursor on the object name in the FROM clause and press F4. This brings up a window shown in Figure 6, which shows the columns, data, scripts, indexes, and relationships. This window can be resized and moved around.

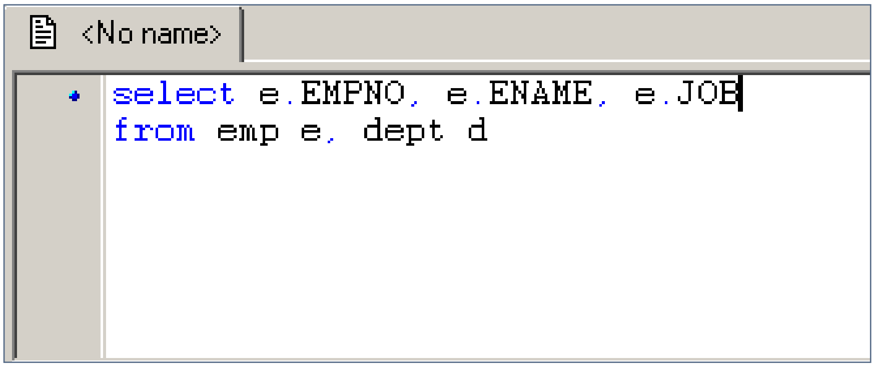
Column names can also be dragged and dropped from this window. Data can easily be viewed along with all the other information you see on these tabs. The INDEXES tab is useful when doing SQL tuning work. Being an old DBA, I am used to working with scripts. I like to see the script that can be used to recreate this object in the SCRIPTS tab. Oracle data can be viewed, filtered and sorted, and saved in a number of formats, as will be discussed in the Schema Browser section of this article.

The Schema Browser shows all database objects the user owns or has access to.

Note the buttons along the top of Figure 6. You can easily start a wizard to edit an object, view key relationships, run statistics, or even truncate all of the data! But be careful out there!

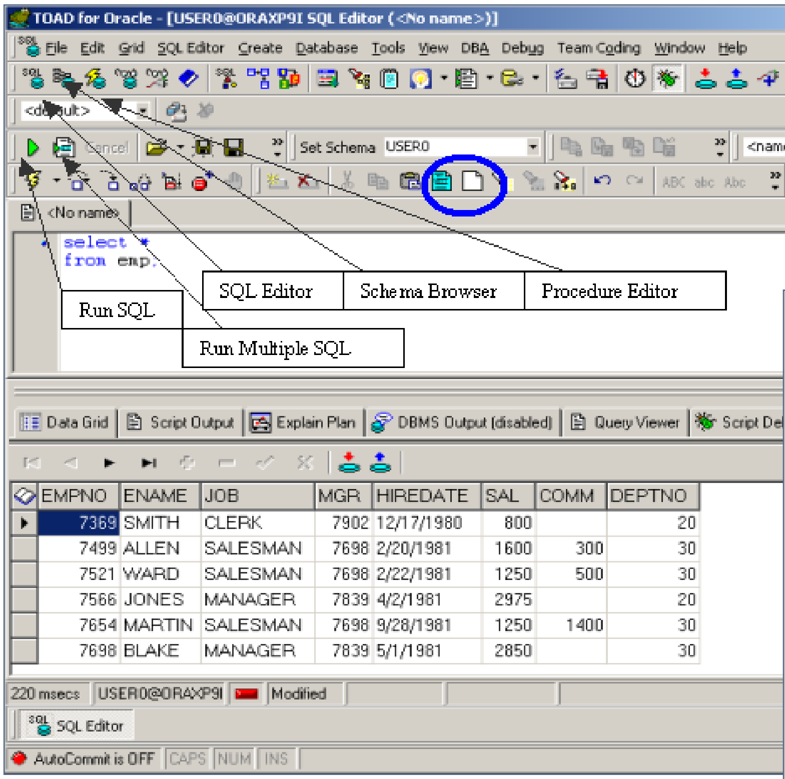
Auto replace is typically used for typo corrections. Auto replace is activated in TOAD every time you hit the space bar. You can easily add your own items for replacement. Typically items for long table names, putline for PL/SQL coding, etc., are added.

Using the menu item Editor Options (Edit > Editor Options), as shown in Figure 7, you can see what is in AUTO REPLACE as shown in Figure 8.



```
select e.EMPNO, e.ENAME, e.JOB
from emp e, dept d
```

Figure 5: Results of Alias Processing



```
select *
from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7566	JONES	MANAGER	7839	4/2/1981	2975		20
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30

Figure 6: F4 Describe Window

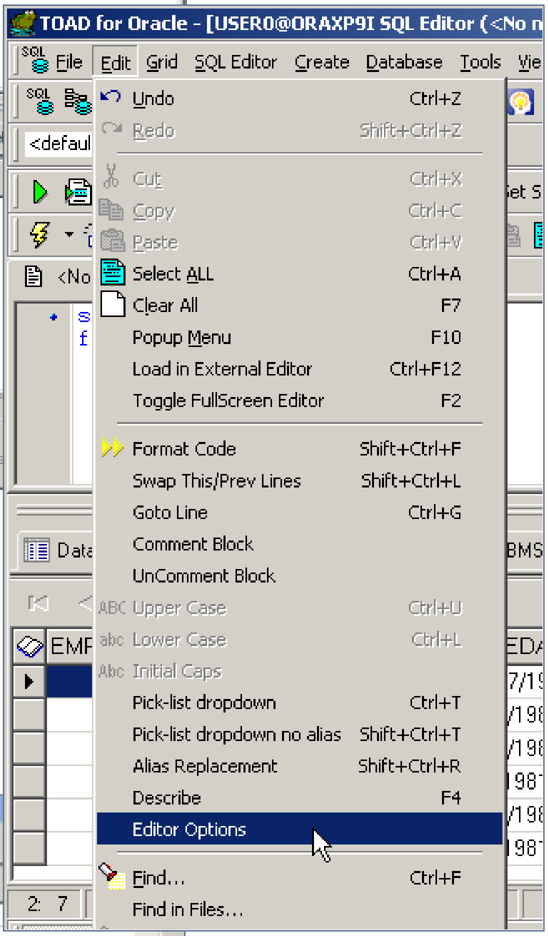
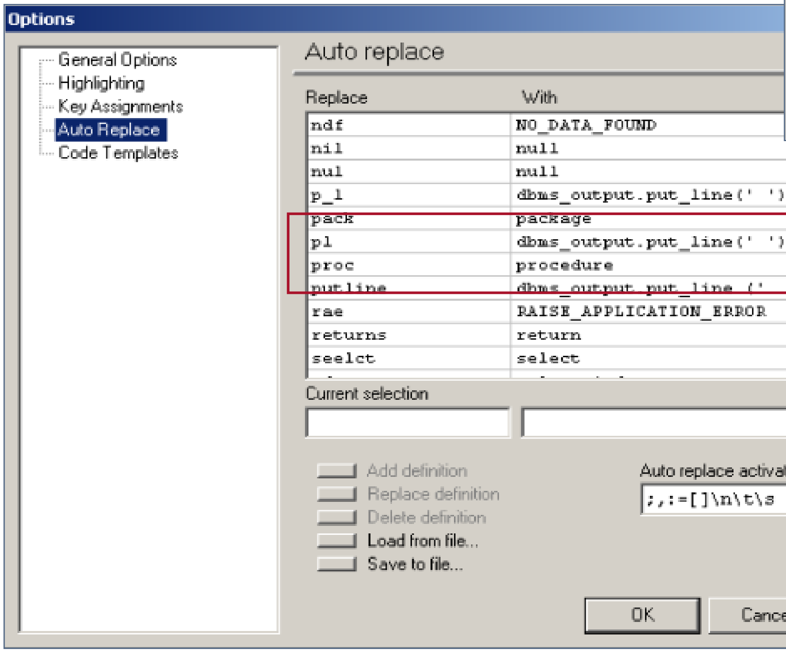

TOAD for Oracle - [USER0@ORAXP9I SQL Editor (<No n
File Edit Grid SQL Editor Create Database Tools Vie
Undo Ctrl+Z
Redo Shift+Ctrl+Z
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Select ALL Ctrl+A
Clear All F7
Popup Menu F10
Load in External Editor Ctrl+F12
Toggle FullScreen Editor F2
Format Code Shift+Ctrl+F
Swap This/Prev Lines Shift+Ctrl+L
Goto Line Ctrl+G
Comment Block
UnComment Block
Upper Case Ctrl+U
Lower Case Ctrl+L
Initial Caps
Pick-list dropdown Ctrl+T
Pick-list dropdown no alias Shift+Ctrl+T
Alias Replacement Shift+Ctrl+R
Describe F4
Editor Options
Find... Ctrl+F
Find in Files...

Figure 7: Accessing Auto Replace Options

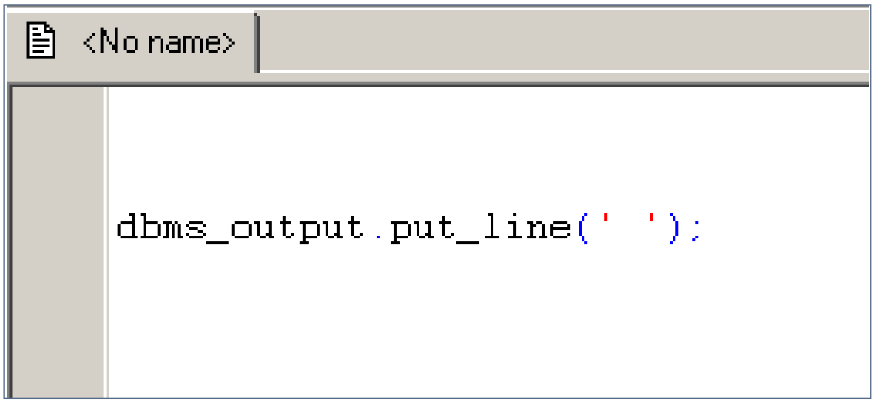


Replace	With
ndf	NO_DATA_FOUND
nil	null
nul	null
p_l	dbms_output.put_line(' ');
pack	package
pl	dbms_output.put_line(' ');
proc	procedure
putline	dbms_output.put_line (' ');
rae	RAISE_APPLICATION_ERROR
returns	return
seelct	select

Auto replace activation key(s): `;,:=[]\n\t\s`

Figure 8: Auto Replace Options

In this example, there are a variety of common typos that are addressed. You can also see the "pl" that I added for my dbms_output.put_line template. Every time I enter "pl" followed by the space bar, this code will be substituted in its place, as shown in Figure 9.



<No name>

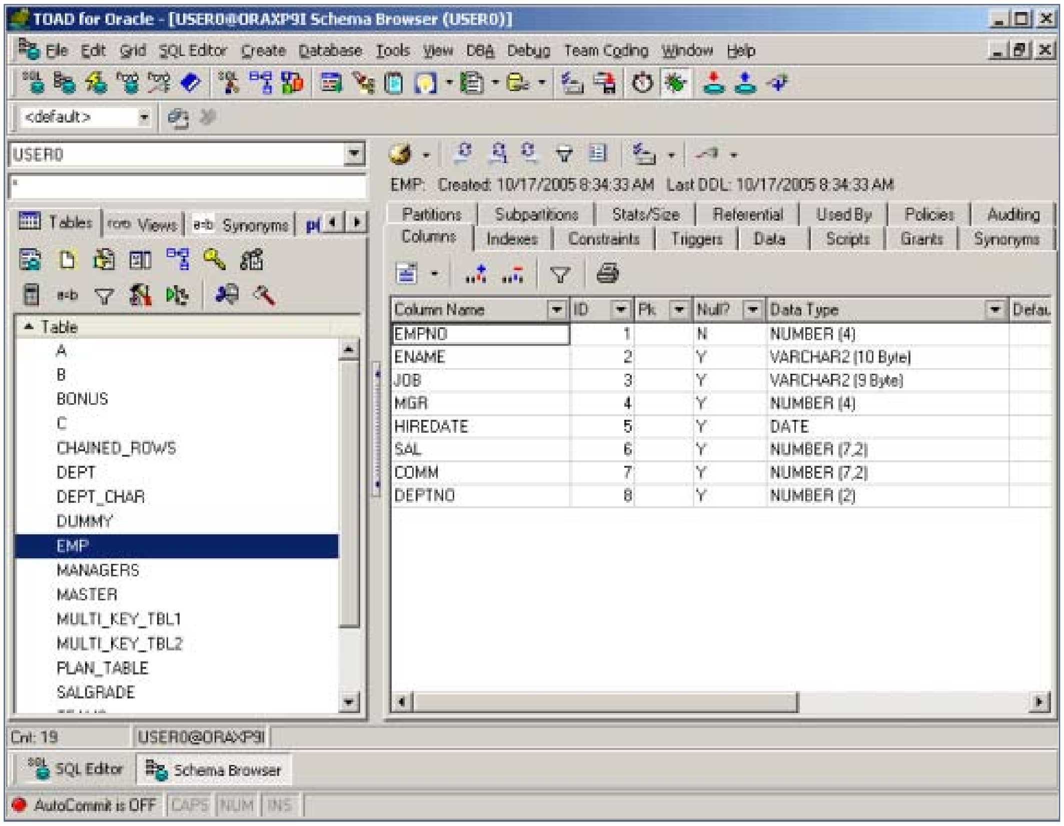
```
dbms_output.put_line(' ');
```

Figure 9: Using Auto Replace

That was a brief overview of some of the top tips and techniques for using TOAD's SQL Editor. But it was barely the tip of the iceberg! Topics not covered in-clude generating template SQL code, saving data, code formatting, opening existing SQL, and saving SQL, which will be addressed in future articles.

Schema Browser

Pressing the Schema Browser button, as shown in Figure 10, accesses the Schema Browser. This interface shows all database objects the user owns or has access to.



TOAD for Oracle - [USER0@ORAXP9I Schema Browser (USER0)]

EMP: Created: 10/17/2005 8:34:33 AM Last DDL: 10/17/2005 8:34:33 AM

Column Name	ID	Pk	Null?	Data Type	Defau
EMPNO	1		N	NUMBER (4)	
ENAME	2		Y	VARCHAR2 (10 Byte)	
JOB	3		Y	VARCHAR2 (9 Byte)	
MGR	4		Y	NUMBER (4)	
HIREDATE	5		Y	DATE	
SAL	6		Y	NUMBER (7,2)	
COMM	7		Y	NUMBER (7,2)	
DEPTNO	8		Y	NUMBER (2)	

Table
A
B
BONUS
C
CHAINED_ROWS
DEPT
DEPT_CHAR
DUMMY
EMP
MANAGERS
MASTER
MULTI_KEY_TBL1
MULTI_KEY_TBL2
PLAN_TABLE
SALGRADE

Cnt: 19 USER0@ORAXP9I

Figure 10: Schema Browser Interface

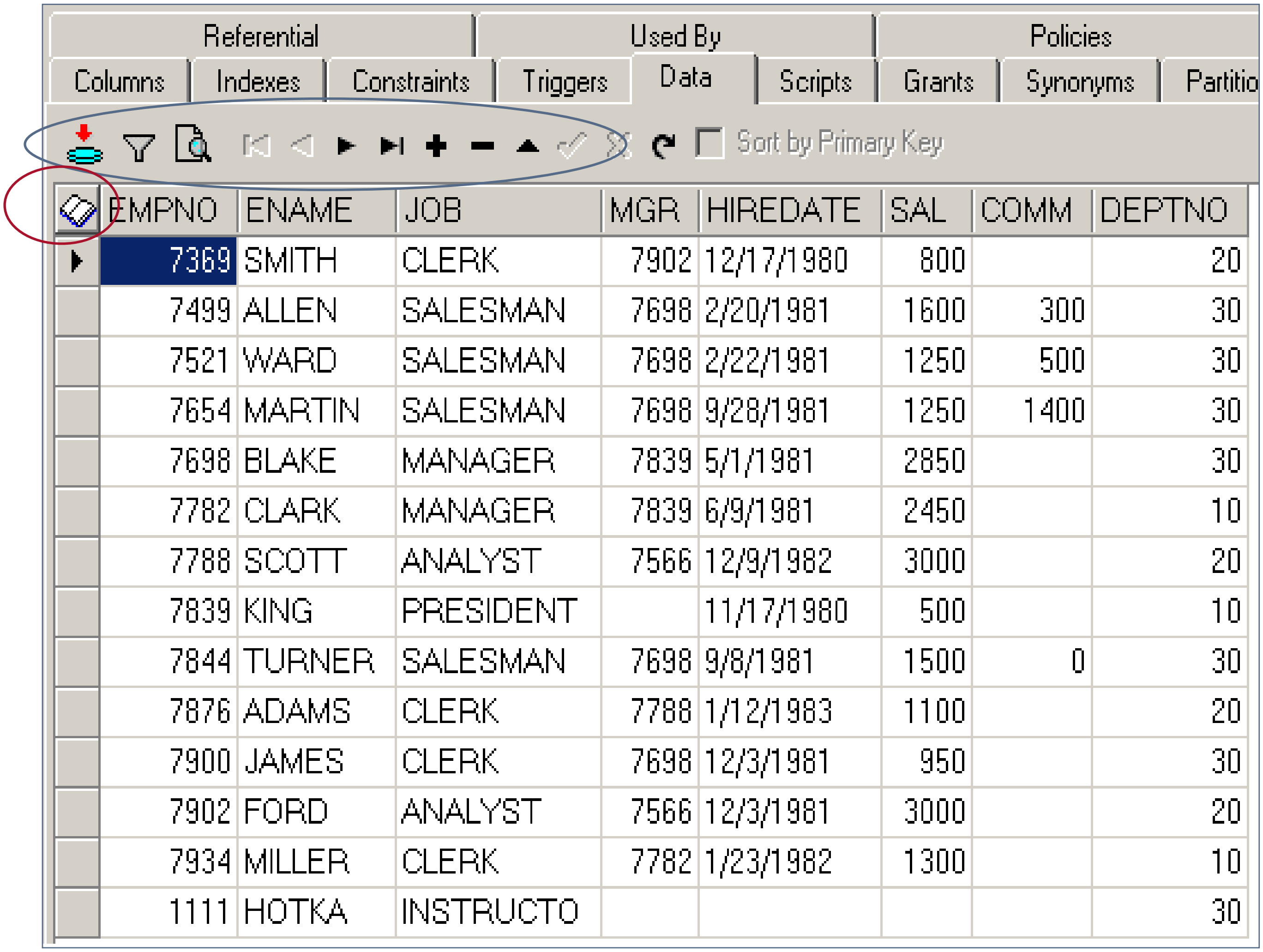
On the left side, the logged in user appears. Using the dropdown menu can change this. TOAD can access any information on the database that the user has permissions to see.

The box just under the user box is a filter. You can have only certain objects appear in the tabbed navigator window (left side). This is handy when the application you are working with has hundreds of objects.

Notice again on the left side that the TABLES tab is displayed. There is a tab for every database object type. Tabs can be tiled with an option and right-mouse clicking on the tabs can hide tabs and uncheck the ones you do not want displayed.

The right side of the Schema Browser is remarkably similar to the F4 DESCRIBE screen in the SQL Editor. In fact, all of the same features discussed here also apply to the F4 DESCRIBE screen.

You can have a Schema Browser window open and drag and drop columns from this window back into the SQL Editor or the Procedure Editor.



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	12/17/1980	800		20
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7521	WARD	SALESMAN	7698	2/22/1981	1250	500	30
7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7782	CLARK	MANAGER	7839	6/9/1981	2450		10
7788	SCOTT	ANALYST	7566	12/9/1982	3000		20
7839	KING	PRESIDENT		11/17/1980	500		10
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
7876	ADAMS	CLERK	7788	1/12/1983	1100		20
7900	JAMES	CLERK	7698	12/3/1981	950		30
7902	FORD	ANALYST	7566	12/3/1981	3000		20
7934	MILLER	CLERK	7782	1/23/1982	1300		10
1111	HOTKA	INSTRUCTO					30

Figure 11: Data Tab

The DATA tab, as shown in Figure 11, has several options. You can add, delete, or edit the data using the buttons along the top, (encased in a blue circle on Figure 11). If you click on a row, then click the little book button (encased in a red circle on Figure 11), this will bring the whole row up for easy editing.

TOAD has two different reporting options.

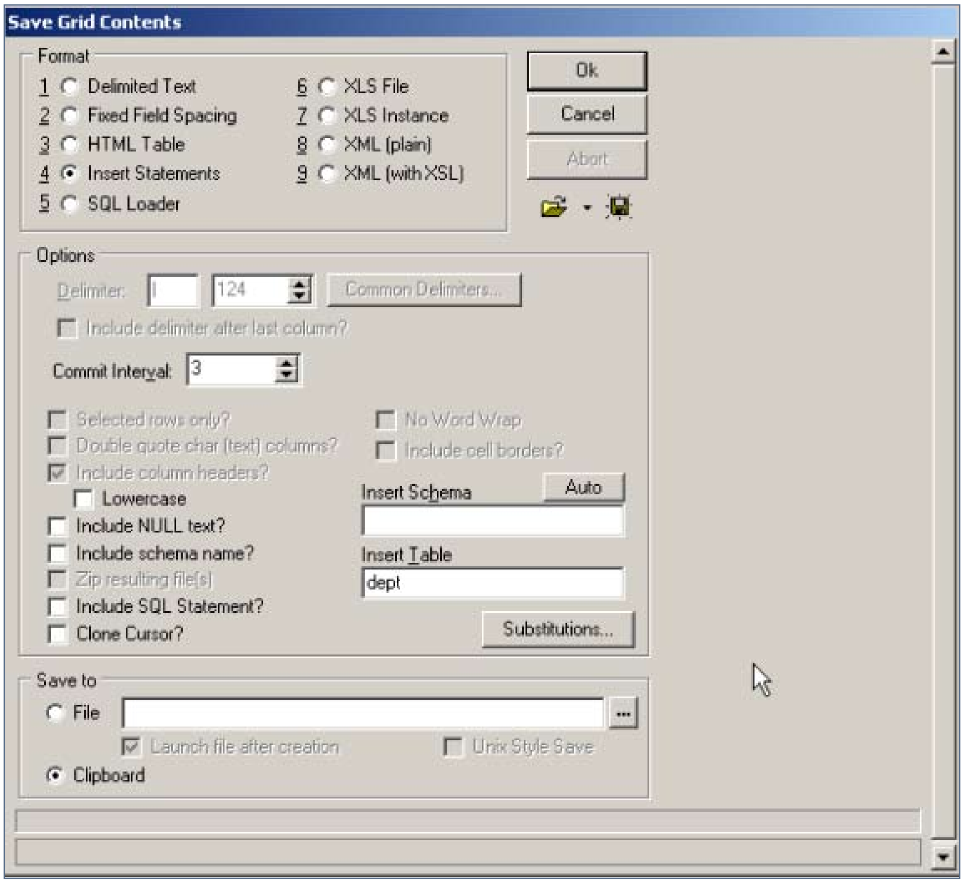
1. You can print the displayed data.

2. You can save the displayed data in a number of formats. I say displayed data because you can use filters and sorts to get the data the way you want and the print and Save As options will only save the data displayed. In the SQL Editor, again, use the SQL itself to filter the data (WHERE clauses) and sort the data (ORDER BY clause).

These options are all available by right-mouse clicking on the data itself. Select Save As from the popup menu and you will see the screen in Figure 12.

Data can be saved in a variety of formats, as shown in Figure 12.

Code templates are similar to the auto replace except that they are signaled with the CTRL-space bar key combination.



Save Grid Contents

Format

1 ○ Delimited Text
2 ○ Fixed Field Spacing
3 ○ HTML Table
4 ● Insert Statements
5 ○ SQL Loader
6 ○ XLS File
7 ○ XLS Instance
8 ○ XML (plain)
9 ○ XML (with XSL)

Options

Delimiter: | 124

□ Include delimiter after last column?

Commit Interval: 3

□ Selected rows only?
□ Double quote char (text) columns?
☑ Include column headers?
□ Lowercase
□ Include NULL text?
□ Include schema name?
□ Zip resulting file(s)
□ Include SQL Statement?
□ Clone Cursor?
□ No Word Wrap
□ Include cell borders?

Insert Schema

Insert Table: dept

Save to

○ File
☑ Launch file after creation
□ Unix Style Save
● Clipboard

Figure 12: Save As Options Screen

There are several different options depending on what you selected in the upper radio group.

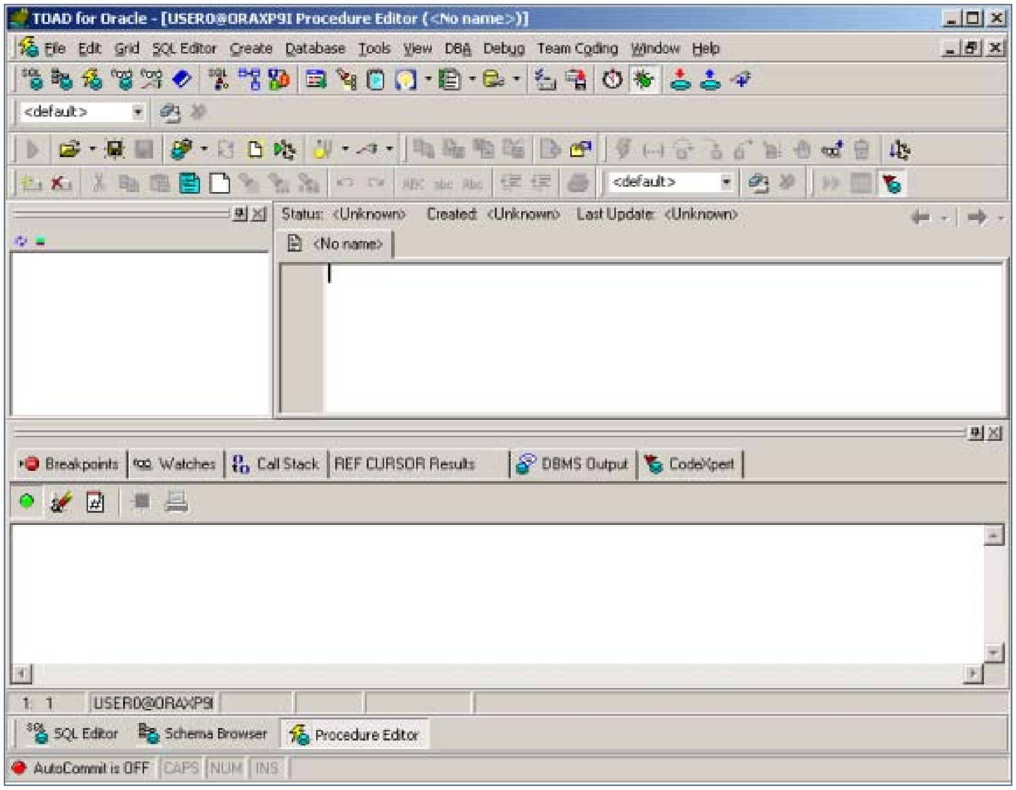
The INSERT statement selection allows you to put “COMMIT;” into the created text after a selected number of rows. Output options here include the clipboard; use it to paste back into the SQL Editor window, or a file. This feature along with DDL generation helps quickly and easily generate a test environment complete with test data.

Procedure Editor

The Procedure Editor is accessed via the Procedure Editor button. This brings up a procedure, function, package development environment, as shown in Figure 13, complete with symbolic debugger. Existing procedures, functions, triggers, and packages can be opened from a file or loaded into the Procedure Editor from the Schema Browser (PROCS tab, and either double clicking on the code or right-mouse clicking and clicking on load into procedure editor).

All the coding shortcuts described in the SQL Editor also work here.

Code templates are similar to the auto replace except that they are signaled with the CTRL-space bar key combination. TOAD comes with a variety of existing templates



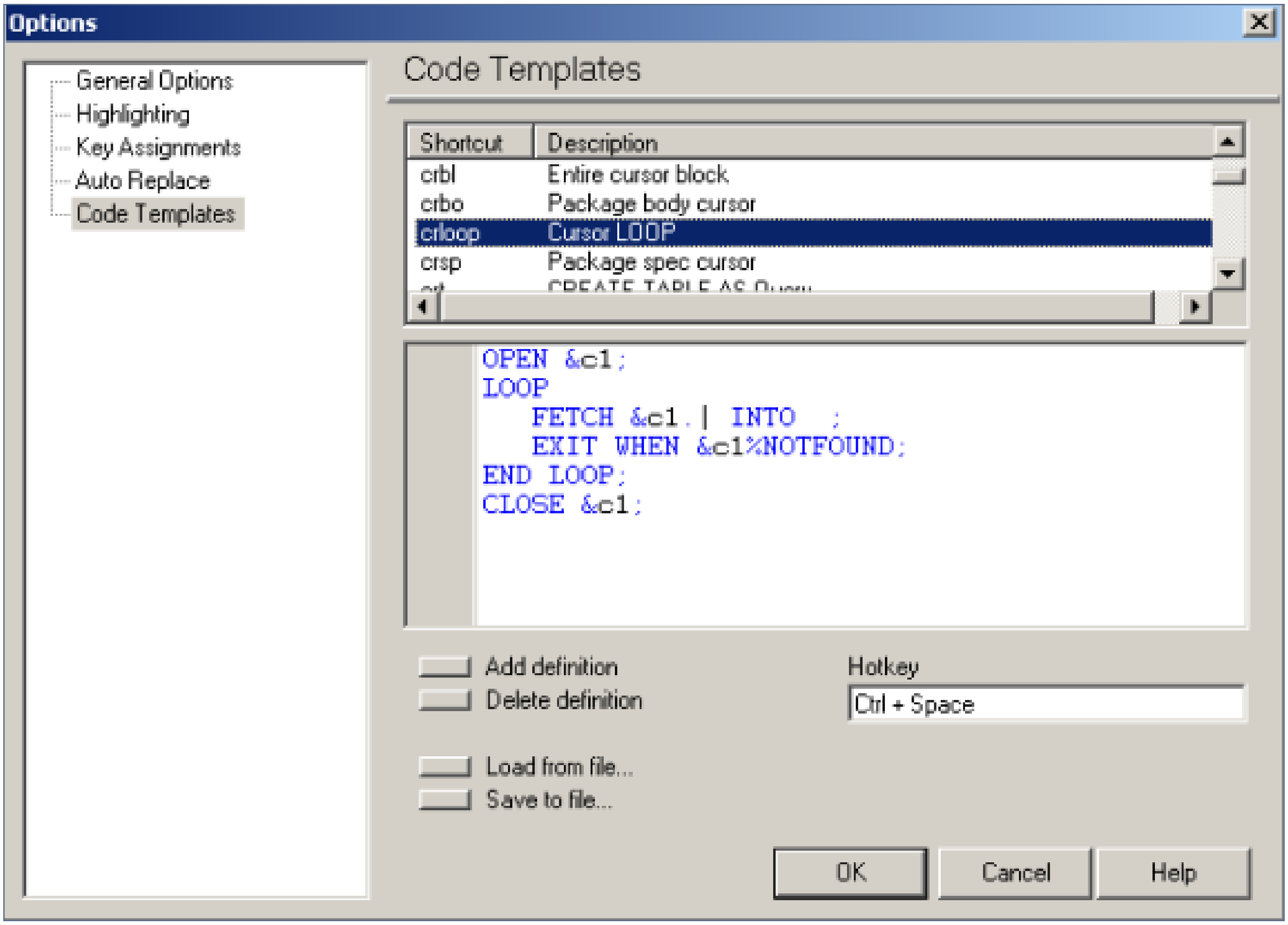
TOAD for Oracle - [USER0@ORAXP9I Procedure Editor (<No name>)]

Status: <Unknown> Created: <Unknown> Last Update: <Unknown>

1: 1 USER0@ORAXP9I

AutoCommit is OFF CAPS NUM INS

Figure 13: Procedure Editor



Shortcut	Description
crbl	Entire cursor block
crbo	Package body cursor
crloop	Cursor LOOP
crsp	Package spec cursor
crt	CREATE TABLE AS Query

```
OPEN &c1;
LOOP
   FETCH &c1.| INTO  ;
   EXIT WHEN &c1%NOTFOUND;
END LOOP;
CLOSE &c1;
```

Hotkey: Ctrl + Space

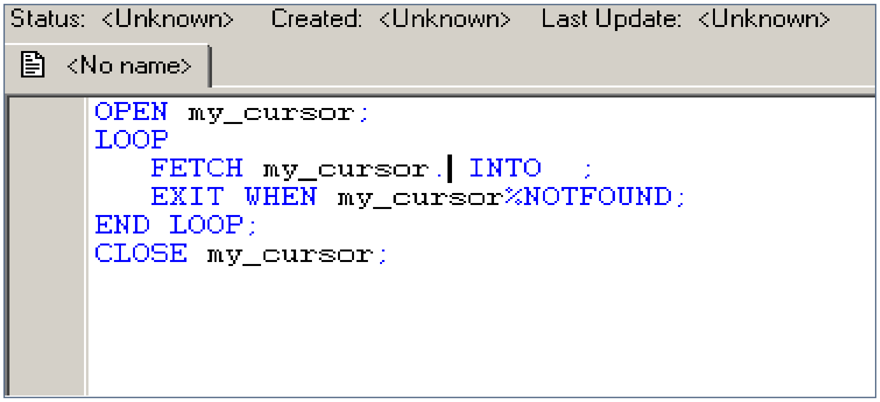
Figure 14: Code Template Options

and like auto replace, you can easily add your own.

Adding or changing the code templates is accessed from the menu item Code Templates (Edit > Editor Options > Code Templates). Figure 14 shows the code template options screen.

Figure 15 displays a sample cursor loop code template. This is accessed by entering "crloop" and then pressing the CTRL-space key combination. Note the vertical bar in the code: the cursor will be placed at that point. Also notice the substitution variables. TOAD will prompt you for these. In this example, you would enter in the cursor name.

The symbolic debugger supports break points, (stops the code), conditional break points (can stop the code after so many passes of a loop or when a variable meets a condition), watch variables, watch cursors, modify variable contents, and step in and out of called procedures. You can run to a break point, or you can step through the code one line at a time, or you can run the code until the debugger reaches the line with the cursor on it.


Status: <Unknown>   Created: <Unknown>   Last Update: <Unknown>

```
OPEN my_cursor;
LOOP
   FETCH my_cursor.| INTO  ;
   EXIT WHEN my_cursor%NOTFOUND;
END LOOP;
CLOSE my_cursor;
```

Figure 15: Cursor Loop Template

Another popular feature is that you can hover the cursor over any variable and TOAD will display its contents. If you hover over an implicit cursor, TOAD will display the current row's information!

These buttons from the Debug menu bar, as shown in Figure 16, control the debug environment. Each of these items also has an associated button on the tool bar and as you can see, most of these items also have an associated function key.

Summary

We just scratched the surface of TOAD's capabilities in this article. I have covered what I consider to be TOAD's major features and features that will get you productive using TOAD.

Watch for additional TOAD articles on using the debugger and finding application problems.

Dan Hotka – Dan is a Training Specialist who has over 27 years in the computer industry and over 22 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 Database Oracle10g Linux Administration by Oracle Press. He is also the author of Oracle9i Development by Example and Oracle8i from Scratch by Que and has co-authored six other popular books. He is frequently published in Oracle trade journals and regularly speaks at Oracle conferences and user groups around the world. Dan may be contacted at **Dan.Hotka@ERPtips.com.**

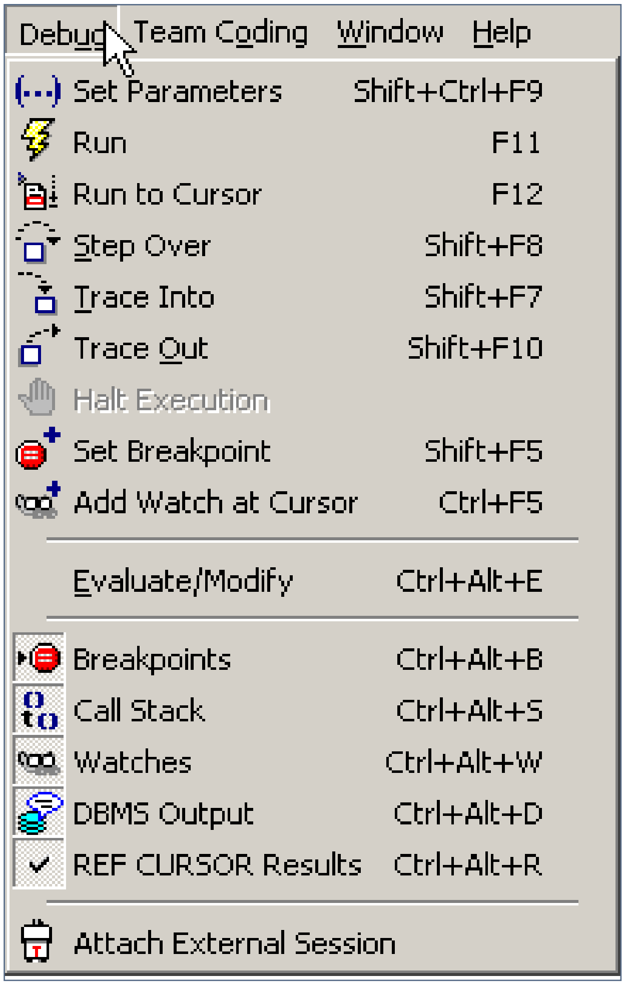


Figure 16: Debugger Menu Bar

ORAtips Journal

The information on our website and in our publications is the copyrighted work of Klee Associates, Inc. and is owned by Klee Associates, Inc. NO WARRANTY: This documentation is delivered as is, and Klee Associates, Inc. makes no warranty as to its accuracy or use. Any use of this documentation is at the risk of the user. Although we make every good faith effort to ensure accuracy, this document may include technical or other inaccuracies or typographical errors. Klee Associates, Inc. reserves the right to make changes without prior notice. NO AFFILIATION: Klee Associates, Inc. and this publication are not affiliated with or endorsed by Oracle Corporation. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Klee Associates, Inc. is a member of the Oracle Partner Network

This article was originally published by Klee Associates, Inc., publishers of JDEtips and SAPtips. For training, consulting, and articles on JD Edwards or SAP, please visit our websites: **www.JDEtips.com** and **www.SAPtips.com**.