3

TOAD SQL Editor

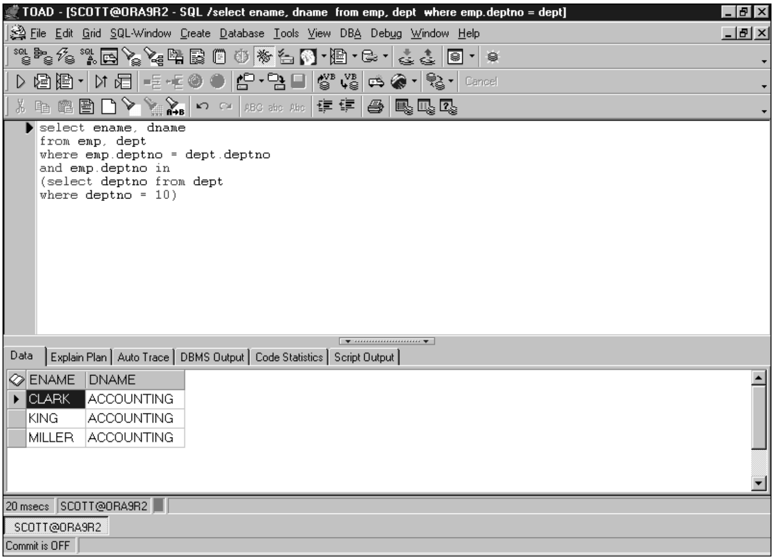
The SQL Editor is the original development area of TOAD. This window enables you to type, save, run, and tune SQL statements. In addition, you will learn how to use TOAD to create and execute SQL scripts, save the output, and examine the explain plan. This chapter will discuss and illustrate every option available in the SQL Editor.

Overview

TOAD provides a number of features that make SQL development easy:

- Keyboard shortcuts
- Table and column select lists
- SQL templates
- Options for creating and executing SQL scripts
- Options for reviewing, editing, and saving result-set data
- Compatibility with SQL*Plus

The Editor window is the basis of the entire TOAD tool, giving you the ability to create and edit SQL: both individual SQL statements (possibly to be inserted into applications) and scripts that contain multiple SQL statements. Figure 3.1 shows the basic SQL Editor window.


TOAD - [SCOTT@ORA9R2 - SQL /select ename, dname from emp, dept where emp.deptno = dept]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data Explain Plan Auto Trace DBMS Output Code Statistics Script Output
20 msecs SCOTT@ORA9R2
SCOTT@ORA9R2
Commit is OFF

```
select ename, dname
from emp, dept
where emp.deptno = dept.deptno
and emp.deptno in
(select deptno from dept
where deptno = 10)
```

ENAME	DNAME
CLARK	ACCOUNTING
KING	ACCOUNTING
MILLER	ACCOUNTING

FIGURE 3.1 TOAD SQL Editor window.

This illustration shows the default SQL Editor. Notice the various buttons at the top that perform about any function (including executing the current SQL, saving the current SQL, and so on). There are three rows of buttons, or three TOAD toolbars. Hover the mouse over a button and a balloon will appear with a description of its use.

A *shortcut* is a keystroke or keystrokes that perform a certain function. F1, for example, brings up the TOAD help facility. There is a button on the toolbar for about every shortcut. The savvy TOAD user makes extensive use of the shortcuts.

F1 brings up the TOAD help facility.

Figure 3.2 illustrates the toolbars.

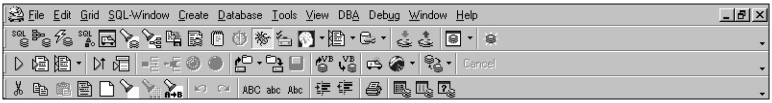

File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
ABC abc Abc

FIGURE 3.2 TOAD SQL Editor window toolbars.

The first toolbar provides easy access to the main TOAD browsers and editors as well as the save functions. Some additional TOAD features also appear on this toolbar. The first toolbar (left to right) contains the following icons:

Open a New SQL Window

Open a New Schema Browser Window

Open a New Procedure Edit Window

Open a New SQL Modeler Window

Explain Plan Window

Open a New DBMS Output Window

Find Object

Save All Options

Reports

Open a New Text Editor Window

Toggle PL/SQL Profiling

Toggle Compiling with Debug

Configure TOAD Options

Execute a Knowledge Xpert Module or Formatter Plus

Script Manager

Configure/Execute External Tools

Commit

Rollback

Show Windows by Connection

Open a New Oracle Connection

The second toolbar focuses on execution. This toolbar enables you to execute code and scripts, and allows code to be loaded into the environment by a number of methods. The second (middle) toolbar contains the following icons:

Execute Statement

Execute Current Statement

Execute as a Script

Recall a Previous Statement

Recall a Personal Statement

Insert a Row

Delete Current Row

Post Data Changes

Revert Data Changes

Load a File into the Editor

Save Editor to File

Save Edits to File

Create a Code Statement

Strip All Non-SQL Syntax

Run Explain Plan for Current Statement

Tune the Current Statement using SQLab Xpert tuner

Change Session for this Window

Cancel Cancel

The third and final toolbar contains shortcuts for the standard Windows actions like cut and paste, clear, and so on. This toolbar also enables you to get information on specific objects as well. The third toolbar contains the following icons:

Cut

Copy

Paste

Select All

Clear All

Find Text

Find Next

Replace Text

Undo Edit

Redo Last Undo

Convert to Uppercase

Convert to Lowercase

Convert to Init Cap

Indent Text

Unindent Text

Print Text

Show Table Select Window

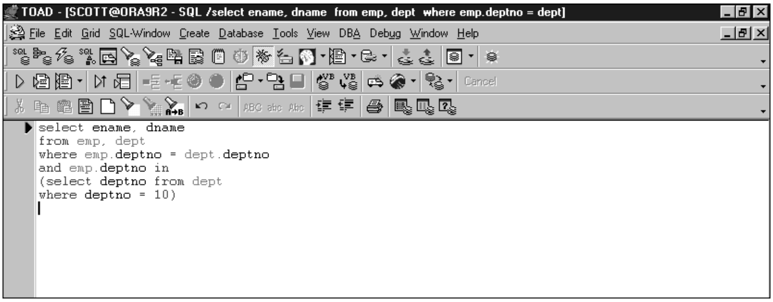
Show Column Select Window

Show SQL Template Window

The first shortcut is F2. This toggles the bottom output window, or a better description might be: toggles the SQL Editor window to full screen. Shift+F2 toggles the grid output (on the bottom) to full screen. Figure 3.3 shows the SQL Editor with the output toggled off, or the full-screen grid. This is helpful when working on longer SQL statements or SQL scripts. You can easily toggle on the output tabs when you want to see the output.

F2 toggles on/off the full-screen editor.

Shift+F2 toggles on/off the full-screen data grid.


TOAD - [SCOTT@ORA9R2 - SQL /select ename, dname from emp, dept where emp.deptno = dept]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel

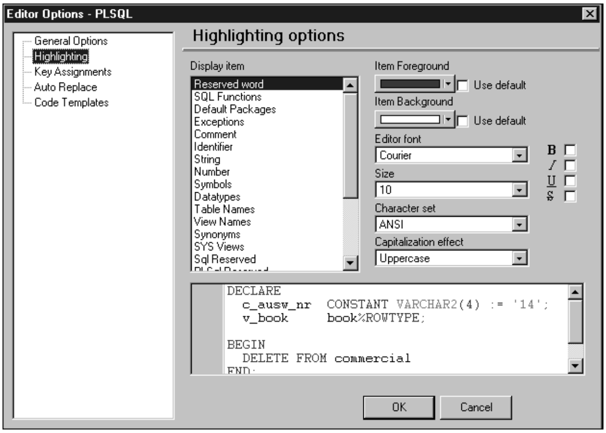
```
select ename, dname
from emp, dept
where emp.deptno = dept.deptno
and emp.deptno in
(select deptno from dept
where deptno = 10)
```

FIGURE 3.3 TOAD SQL Editor window, full-screen grid.

The lower section, or data grid, contains the result-set data from the query, the explain plan used to retrieve the data, code statistics, Auto Trace output, DBMS output, and Script output. Each of these will be covered in this chapter.

Notice that the SQL syntax appears (along with any other Oracle reserved words) in blue where the supplied columns, table names, and other variable syntax appear in black. Comments appear in green, and so on. These color patterns are controlled by the Editor Options. You can access these options by clicking Edit, Editor Options from the menu bar or by right-clicking and selecting Editing Options. Notice that TOAD lists the keyboard shortcuts whenever possible.

Figure 3.4 illustrates the Highlighting options in the SQL window. You can see that you have complete control over the editor environment (such as autoreplacement of words, general layout and text wrapping in the edit window, key assignments, and code templates).



Item Foreground □ Use default

Item Background □ Use default

Editor font: Courier

Size: 10

Character set: ANSI

Capitalization effect: Uppercase

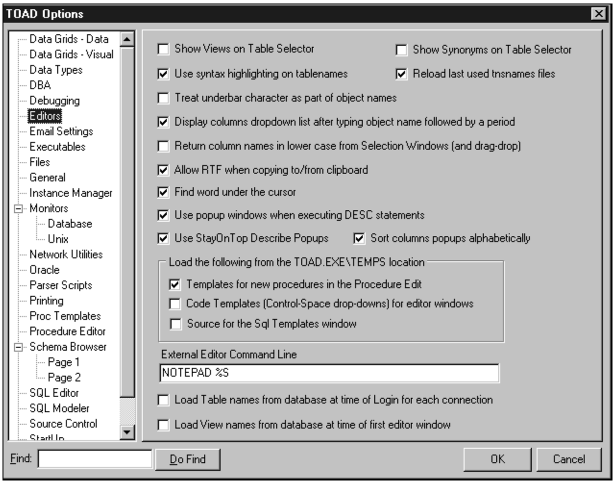
```
DECLARE
  c_ausw_nr  CONSTANT VARCHAR2(4) := '14';
  v_book     book%ROWTYPE;

BEGIN
  DELETE FROM commercial
END;
```

FIGURE 3.4 TOAD SQL Editor Options.

TOAD has three editors: the SQL Editor, the Procedure Editor (covered in Chapter 4), and a text editor of your choice. The editor environment applies its options to both TOAD editors. Additional editors can easily be added to TOAD. Choose View, Options from the menu bar, and then select Editors (or use the Configure TOAD Options button) to add your editor of choice. Figure 3.5 illustrates how to add the Notepad editor, for example. Be sure to use the variable `%s` to pass this editor the SQL that you are currently working on. If your current session has not been saved, you will be prompted to save it. Also, upon exiting your external editor, you will be prompted to reload your work from the saved file. Make sure the option Reload Files When Activating TOAD is checked on in the Procedure Editor section of the TOAD Options screen. You then use this external editor by choosing Edit, Load in External Editor from the menu bar or by using the shortcut Ctrl+F12. Figure 3.6 shows some work in the Notepad editor.

Ctrl+F12 accesses a previously defined external editor.

TOAD supports threads, which allows SQL statements to be canceled while they are running. If you want this behavior, make sure you check the box Process Statements in Threads in the SQL Editor part of the TOAD Options screen. The Cancel button (far right button on the middle SQL Editor toolbar) will become available during the execution of a SQL statement being run in this fashion. In this same area, you can also increase or decrease the SQL statements TOAD will automatically track. These SQL statements are stored in the file SQLS.DAT in your TOAD home directory. You have control over default behavior such as whether you are prompted to save the current SQL (Prompt to Save Contents), code format options, showing execution time, and so on.



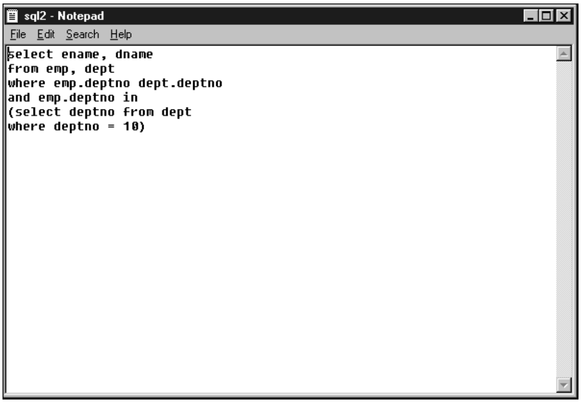
□ Show Views on Table Selector
□ Show Synonyms on Table Selector
☑ Use syntax highlighting on tablenames
☑ Reload last used tnsnames files
□ Treat underbar character as part of object names
☑ Display columns dropdown list after typing object name followed by a period
□ Return column names in lower case from Selection Windows (and drag-drop)
☑ Allow RTF when copying to/from clipboard
☑ Find word under the cursor
☑ Use popup windows when executing DESC statements
☑ Use StayOnTop Describe Popups
☑ Sort columns popups alphabetically

Load the following from the TOAD.EXE\TEMPS location
☑ Templates for new procedures in the Procedure Edit
□ Code Templates (Control-Space drop-downs) for editor windows
□ Source for the Sql Templates window

External Editor Command Line: NOTEPAD %S

□ Load Table names from database at time of Login for each connection
□ Load View names from database at time of first editor window

FIGURE 3.5 Defining external editors in TOAD.



```
select ename, dname
from emp, dept
where emp.deptno dept.deptno
and emp.deptno in
(select deptno from dept
where deptno = 10)
```

FIGURE 3.6 Using Notepad as editor in TOAD.

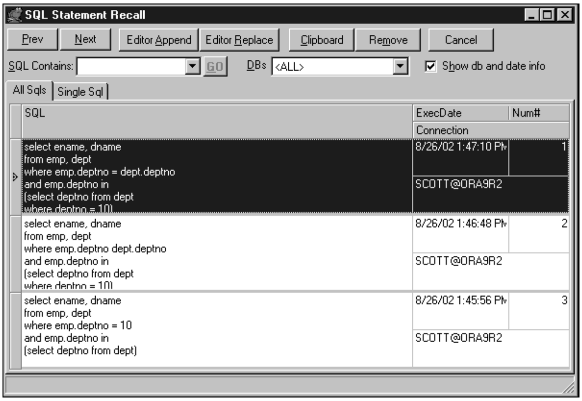
There are several ways to get SQL into the SQL Editor. You can simply type in a new SQL statement. You can use the SQL Statement Recall button (fourth button on the middle SQL Editor toolbar) and select a SQL statement from the stored TOAD SQL history (see Figure 3.7). Pressing Alt+Up arrow and Alt+Down arrow also walks you thru the SQL statement history. You can also choose File, Open from the menu bar (or Ctrl+O), and cut and paste SQL code into the Editor from other applications. The Load option is also useful for loading in SQL from files from the pop-up menu that appears when you right-click.

Alt+Up arrow gets the previous SQL statement from the TOAD history.

Alt+Down arrow gets the next SQL statement from the TOAD history.

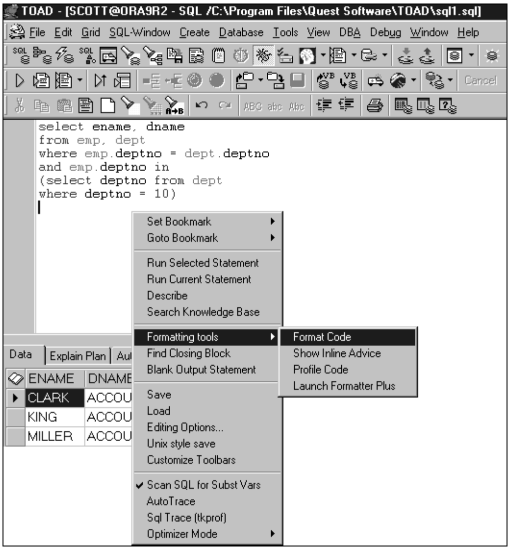
TOAD will also easily format your SQL into an easy-to-read format. Figure 3.8 shows how to access the formatter by right-clicking and selecting Formatting Tools, Format Code from the context menu. Figure 3.9 shows how TOAD formats the SQL.

This overview covered some of the basic concepts and features of the SQL Editor. The remainder of this chapter will cover specific topics in the SQL Editor.



SQL	ExecDate Connection	Num#
select ename, dname from emp, dept where emp.deptno = dept.deptno and emp.deptno in (select deptno from dept where deptno = 10)	8/26/02 1:47:10 PM SCOTT@ORA9R2	1
select ename, dname from emp, dept where emp.deptno dept.deptno and emp.deptno in (select deptno from dept where deptno = 10)	8/26/02 1:46:48 PM SCOTT@ORA9R2	2
select ename, dname from emp, dept where emp.deptno = 10 and emp.deptno in (select deptno from dept)	8/26/02 1:45:56 PM SCOTT@ORA9R2	3

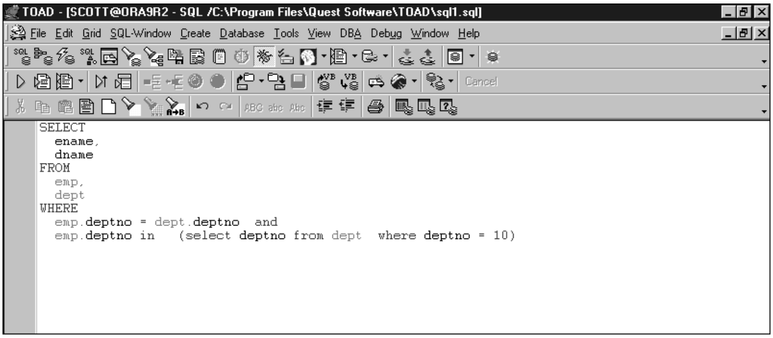
FIGURE 3.7 Selecting SQL from TOAD history.



```
select ename, dname
from emp, dept
where emp.deptno = dept.deptno
and emp.deptno in
(select deptno from dept
where deptno = 10)
```

ENAME	DNAME
CLARK	ACCOU
KING	ACCOU
MILLER	ACCOU

FIGURE 3.8 Accessing SQL Formatter.


TOAD - [SCOTT@ORA9R2 - SQL /C:\Program Files\Quest Software\TOAD\sql1.sql]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel

```
SELECT
  ename,
  dname
FROM
  emp,
  dept
WHERE
  emp.deptno = dept.deptno  and
  emp.deptno in   (select deptno from dept  where deptno = 10)
```

FIGURE 3.9 TOAD formatted SQL.

Predefined Shortcuts

Keyboard shortcuts are one of the features that make TOAD so powerful and easy to use. TOAD comes with a host of predefined shortcuts. These shortcuts save keystrokes and mouse actions and perform a variety of tasks such as issuing a describe on the current highlighted object, or find (find next, find previous, and so on). The shortcuts differ slightly between the SQL Editor and the PL/SQL Editor (covered in Chapter 4). This section focuses on the shortcuts for the SQL Editor only.

Table 3.1 shows all the shortcuts in keystroke order and Table 3.2 shows all the shortcuts in description order. The third column refers you to the figure in this chapter where the shortcut is illustrated.

TABLE 3.1 SQL Editor Shortcuts by Keystroke

Shortcut	Description	Illustration
F1	Windows Help File	
F2	Toggle Output Window	Figure 3.2
Shift+F2	Toggle Data Grid Window	
F3	Find Next Occurrence	
Shift+F3	Find Previous Occurrence	
F4	Describe Table, View, Procedure, Function	Figure 3.26
F5	Execute SQL as a script	Figure 3.34
F6	Toggle between SQL Editor and Results Panel	Figure 3.1
F7	Clear All Text	
F8	Recall Previous SQL Statement	
F9	Execute Statement	Figure 3.33
Ctrl+F9	Set Code Execution Parameters	Figure 3.33
Shift+F9	Execute Current SQL statement at Cursor	
F10 or right-click	Pop-up Menu	Figures 3.4, 3.10

TABLE 3.1 Continued

Shortcut	Description	Illustration
Ctrl+F12	External Editor, Pass Contents	Figures 3.5, 3.6
Ctrl+A	Select All Text	
Ctrl+C	Copy	
Ctrl+E	Execute Explain Plan on the Current Statement	Figure 3.41
Ctrl+F	Find Text	
Ctrl+G	Goto Line	
Ctrl+L	Convert Text to Lowercase	
Ctrl+M	Make Code Statement	
Ctrl+N	Recall Named SQL Statement	
Ctrl+O	Open a Text File	
Ctrl+P	Strip Code Statement	
Ctrl+R	Find and Replace	
Ctrl+S	Save File	
Shift+Ctrl+S	Save File As	Figure 3.18
Ctrl+T	Columns Drop-down	
Shift+Ctrl+R	Alias Replacement	
Shift+Ctrl+T	Columns Drop-Down no alias	Figure 3.19
Ctrl+Spacebar	Code Templates	Figure 3.30
Ctrl+U	Converts Text to Uppercase	
Ctrl+V	Paste	
Ctrl+X	Cut	
Ctrl+Z	Undo Last Change	
Ctrl+.	Display Pop-up List of Matching Table Names	Figure 3.16
Shift+Ctrl+Z	Redo Last Undo	
Alt+Up Arrow	Display Previous Statement	Figure 3.7
Alt+Down Arrow	Display Next Statement (After Alt+Up Arrow)	Figure 3.7
Ctrl+Home	In the data grid: goes to the top of the record set	
Ctrl+End	In the data grid: goes to the end of the record set	
Ctrl+Tab	Cycles through the Collection of MDI Child Windows	

The competent TOAD user makes use of Shift+F9 to execute SQL statements one at a time out of a script, and F8 to recall the previous SQL statement. The TOAD user also makes use of cut and paste to move code between TOAD windows.

Shift+F9 executes single SQL statements.

F8 recalls the previous SQL statement.

TABLE 3.2 SQL Editor Shortcuts by Description

Description	Shortcut	Illustration
Alias Replacement	Shift+Ctrl+R	
Clear All Text	F7	
Code Templates	Ctrl+Spacebar	Figure 3.30
Columns Drop-down	Ctrl+T	Figure 3.18
Columns Drop-down no alias	Shift+Ctrl+T	Figure 3.19
Convert Text to Lowercase	Ctrl+L	
Convert Text to Uppercase	Ctrl+U	
Copy	Ctrl+C	
Cut	Ctrl+X	
Cycles through the Collection of MDI Child Windows	Ctrl+Tab	
Describe Table, View, Procedure, Function, or Package	F4	
Display Next Statement (After Alt+Up Arrow)	Alt+Down Arrow	Figure 3.7
Display Previous Statement	Alt+Up Arrow	Figure 3.7
Display Pop-up List of Matching Table Names	Ctrl+.	Figure 3.16
Execute Current SQL Statement at Cursor	Shift+F9	
Execute SQL as a Script	F5	Figure 3.35
Execute Explain Plan on the Current SQL Statement	Ctrl+E	Figure 3.41
Execute SQL Statement	F9	Figure 3.34
External Editor, Pass Contents	Ctrl+F12	Figure 3.5
Find and Replace	Ctrl+R	
Find Next Occurrence	F3	
Find Previous Occurrence	Shift+F3	
Find Text	Ctrl+F	
Goto Line	Ctrl+G	
In the data grid: goes to the end of the record set	Ctrl+End	
In the data grid: goes to the top of the record set	Ctrl+Home	
Make Code Statement	Ctrl+M	
Open a Text File	Ctrl+O	
Paste	Ctrl+V	
Pop-up Menu	F10 or RT-Mouse	Figures 3.4, 3.10
Recall Named SQL	Ctrl+N	
Recall Previous SQL Statement	F8	
Redo Last Undo	Shift+Ctrl+Z	
Save File	Ctrl+S	
Save File As	Shift+Ctrl+S	
Select All Text	Ctrl+A	
Strip Code Statement	Ctrl+P	
Toggle between SQL Editor and Results Panel	F6	Figure 3.1
Toggle Full Screen Editor	F2	Figure 3.2
Undo Last Change	Ctrl+Z	
Verify Statement Without Execution (Parse)	Ctrl+F9	
Windows Help File	F1	

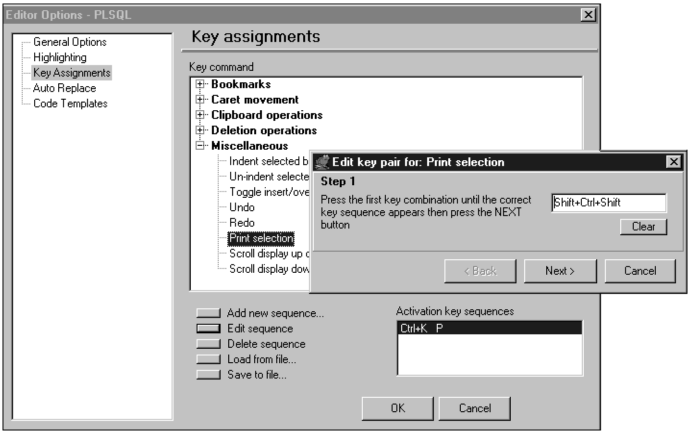
User-Defined Shortcuts

TOAD is completely configurable. You can easily add your own shortcuts to TOAD. It is easy to change the shortcut keystrokes for existing shortcuts, and it is just as easy to add your own shortcuts.

Access the Editor Options menu with a right-click or by pressing the F10 key and selecting Editing Options.

To change an existing keystroke assignment, select Key Assignments, locate the particular assignment to change, and click on the Edit Sequence button as illustrated in Figure 3.10.

To add additional keystroke assignments, select Key Assignments, locate the particular assignment to change, and click on the Add New Sequence button. Type the sequence desired to perform the key assignment task. In Figure 3.11, notice that Shift+Ctrl+D was added by pressing and holding down the Control key while pressing Shift and d.

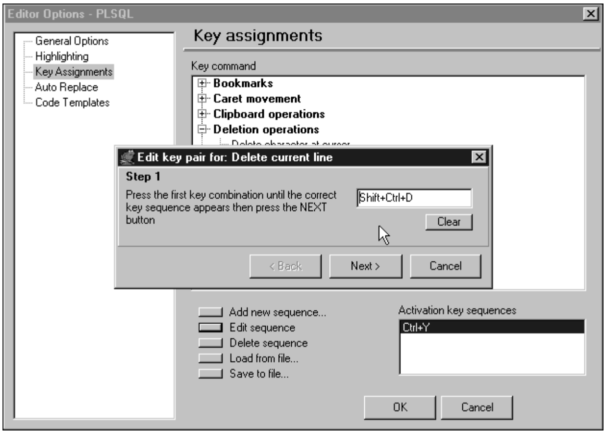


Press the first key combination until the correct key sequence appears then press the NEXT button

Shift+Ctrl+Shift

Ctrl+K P

FIGURE 3.10 Changing shortcut keystroke assignments in TOAD.



Press the first key combination until the correct key sequence appears then press the NEXT button

Shift+Ctrl+D

Ctrl+Y

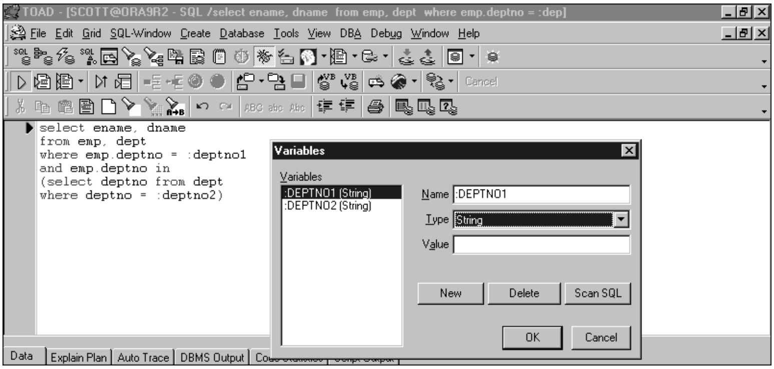
FIGURE 3.11 Adding shortcut keystroke assignments in TOAD.

Using Variables

TOAD supports all kinds of SQL, from all kinds of applications. If you were to bring in SQL, for example, from a SQL*Forms application, it will contain bind variables. Bind variables are used to supply SQL with data at execution time. This allows applications to use the same SQL statement to select and manipulate different data, depending on the data supplied to the bind variables.

> Using the same SQL statement makes efficient use of the Oracle RDBMS SQL pool as the SQL will not be reparsed when using bind variables. The text of the SQL remains the same, so Oracle will reuse the same execution plan, making for a better-performing database environment.

When TOAD encounters bind variables, it will prompt you for their value as in Figure 3.12. This illustration shows two bind variables. The Scan SQL button will check for any missing bind variables, which is particularly useful if you are adding and changing bind variables in this interface.



TOAD - [SCOTT@ORA9R2 - SQL /select ename, dname from emp, dept where emp.deptno = :dep]

```
select ename, dname
from emp, dept
where emp.deptno = :deptno1
and emp.deptno in
(select deptno from dept
where deptno = :deptno2)
```

Variables

:DEPTNO1 (String)
:DEPTNO2 (String)

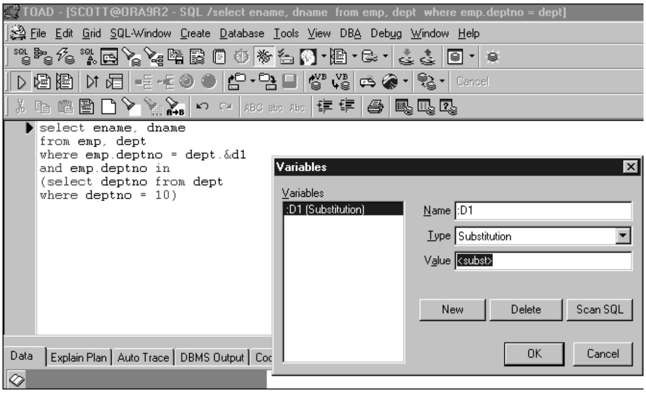
Name :DEPTNO1

Type String

Value

FIGURE 3.12 Resolving SQL bind variables.

Substitution variables work the same way as bind variables. Do remember that substitution variables are resolved into SQL text at parse time, but bind variables won't change the actual SQL text (this greatly enables SQL reuse in the Oracle SQL pool). TOAD will prompt for the data for each substitution variable as shown in Figure 3.13.


TOAD - [SCOTT@ORA9R2 - SQL /select ename, dname from emp, dept where emp.deptno = dept]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data
Explain Plan
Auto Trace
DBMS Output
Coc
Variables
New
Delete
Scan SQL
OK
Cancel

```
select ename, dname
from emp, dept
where emp.deptno = dept.&d1
and emp.deptno in
(select deptno from dept
where deptno = 10)
```

Variables: :D1 (Substitution)

Name :D1

Type Substitution

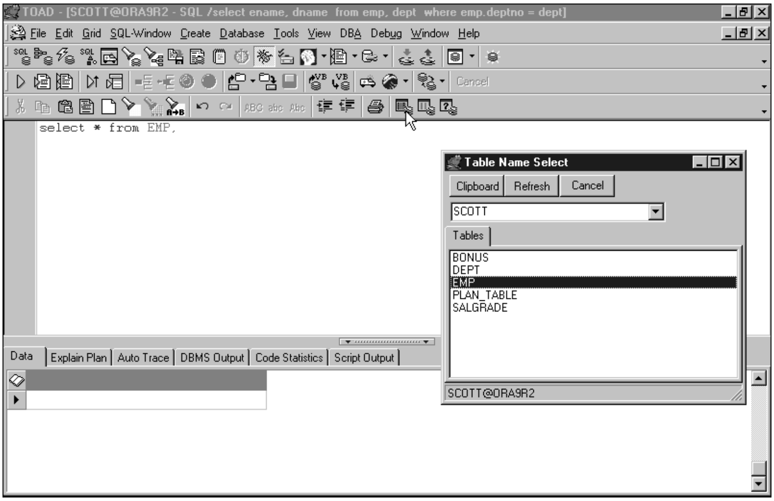
Value <subst>

FIGURE 3.13 Resolving SQL substitution variables.

Table and Column Name Select Lists

TOAD makes it easy to find and work with tables and columns. Figure 3.14 illustrates using the Show Table Select Window button (mouse cursor is pointing to this button). This brings up the Table Name Select window, with a list of available tables for this particular user. If the user has privileges to other schemas, the drop-down list in this window can be changed to the schema owner and those objects will then appear in the select list. Double-clicking on the selected object adds the selected table to the SQL Editor as illustrated in Figure 3.14.

Similarly, columns can be added to the SQL Editor by using the Show Column Select Window button. Figure 3.15 illustrates using the Column Name Select window to add the three columns EMPNO, ENAME, and JOB to the SQL being built in the SQL Editor.



TOAD - [SCOTT@ORA9R2 - SQL /select ename, dname from emp, dept where emp.deptno = dept]

```
select * from EMP,
```

Table Name Select

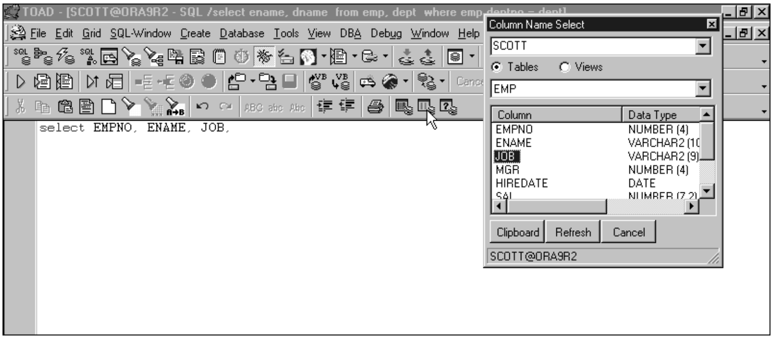
SCOTT

Tables

- BONUS
- DEPT
- EMP
- PLAN_TABLE
- SALGRADE

SCOTT@ORA9R2

FIGURE 3.14 Table Name Select list.



TOAD - [SCOTT@ORA9R2 - SQL /select ename, dname from emp, dept where emp.deptno = dept]

```
select EMPNO, ENAME, JOB,
```

Column Name Select

SCOTT

● Tables ○ Views

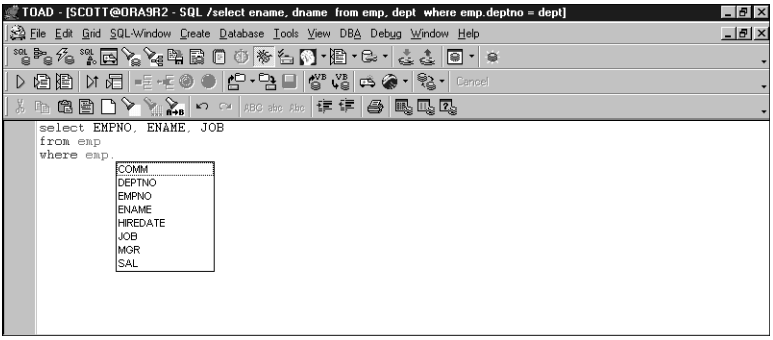
EMP

Column	Data Type
EMPNO	NUMBER (4)
ENAME	VARCHAR2 (10
JOB	VARCHAR2 (9)
MGR	NUMBER (4)
HIREDATE	DATE
SAL	NUMBER (7,2)

SCOTT@ORA9R2

FIGURE 3.15 Column Name Select list.

TOAD will automatically give a column select list if you type or select a valid schema table name followed by a '.'. Wait a second and the column select list will appear as illustrated in Figure 3.16!


TOAD - [SCOTT@ORA9R2 - SQL /select ename, dname from emp, dept where emp.deptno = dept]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
COMM
DEPTNO
EMPNO
ENAME
HIREDATE
JOB
MGR
SAL

```
select EMPNO, ENAME, JOB
from emp
where emp.
```

FIGURE 3.16 Automatic column select lists.

Using Aliases and Autoreplacement Substitutions

TOAD supports the use of its own alias names. Aliases are convenient to shorten keystrokes, using short names instead of rather long table names. TOAD also allows the automatic replacement of text, or in this case, these aliases. The purpose of the alias is to shorten keystrokes. The purpose of the automatic replacement is to automatically resolve the TOAD alias to the full table or column name.

TOAD aliases are used to shorten the number of keystrokes to develop SQL. TOAD aliases are not to be confused with Oracle RDBMS table aliases, which are used in qualifying columns in a multitable SQL statement.

TOAD aliases enable you to

- Access the Column Name Select drop-down list
- Type the alias as a shortcut rather than typing the full table name

Aliases are easy to set up with any text editor. Make sure TOAD is not running (on network installations, make sure all users are off TOAD) and edit the ALIASES.TXT file found in the TOAD installation directory under the `TEMPS` subdirectory.

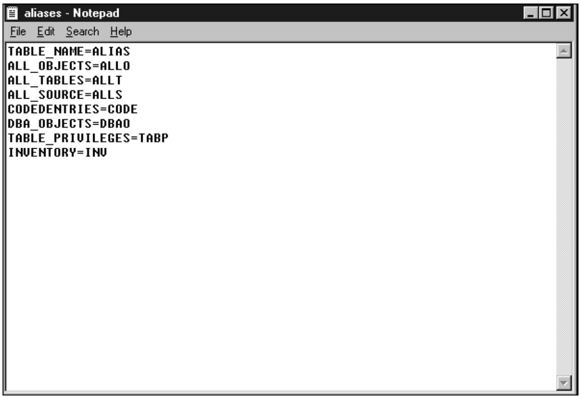
DO NOT edit this file with TOAD running. When TOAD exits, it rewrites this file and any changes you make will be lost!

Figure 3.17 illustrates the format of this file. The format is `<table name>=<alias name>`.

You will learn a method of quickly creating this file in the "Scripts That Write Scripts" section later in this chapter.

TOAD aliases are easy to use. Figure 3.18 shows a simple SQL statement using an alias to get to the Column Name Select drop-down. Notice that the user typed in 'inv.'. The '.' signaled the Column Name Select drop-down and the alias was resolved to the INVENTORY table.

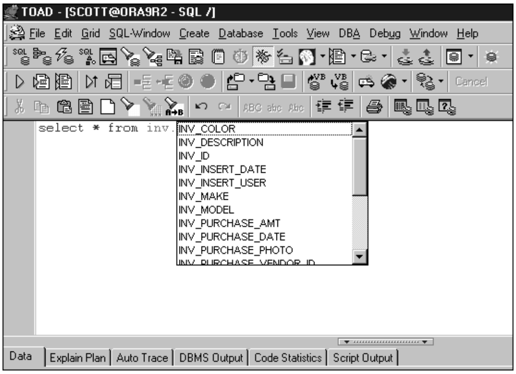
[ic:Keyboard]The '.' signals TOAD to see if this is an alias.



```
TABLE_NAME=ALIAS
ALL_OBJECTS=ALLO
ALL_TABLES=ALLT
ALL_SOURCE=ALLS
CODEDENTRIES=CODE
DBA_OBJECTS=DBAO
TABLE_PRIVILEGES=TABP
INVENTORY=INV
```

FIGURE 3.17 TOAD alias setup.

Notice in Figure 3.19 that there is both an INVENTORY table and an INV table. The alias INV was resolved in Figure 3.18 to the INVENTORY table, not the INV table. Notice the column names in Figure 3.19 compared to those in Figure 3.18. Granted, this is a poor naming convention, but bear with the example. To get the INV table displayed, use Shift+Ctrl+T (or Edit, Columns drop-down no alias) to ignore the alias and get the correct list of columns. See Figure 3.20.


TOAD - [SCOTT@ORA9R2 - SQL /]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Output

```
select * from inv.
```

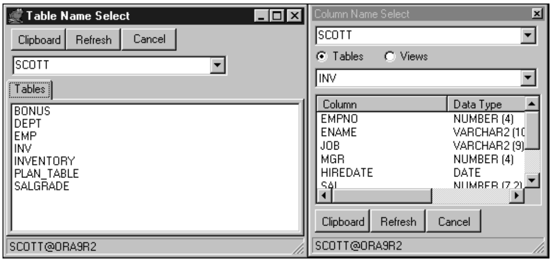
INV_COLOR
INV_DESCRIPTION
INV_ID
INV_INSERT_DATE
INV_INSERT_USER
INV_MAKE
INV_MODEL
INV_PURCHASE_AMT
INV_PURCHASE_DATE
INV_PURCHASE_PHOTO
INV_PURCHASE_VENDOR_ID

FIGURE 3.18 TOAD alias usage.

Shift+Ctrl+T ignores the alias request.

If an alias is identified in the SQL statement, and a Column Select is activated, the alias is automatically added to ALIASES.TXT.

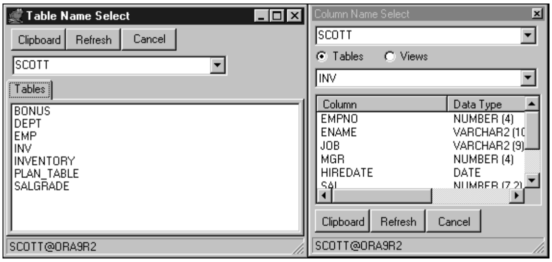
TOAD scans only the first `FROM` clause in any SQL statement, so any TOAD aliases in complex SQL statements that have subqueries, for example, will not be found and resolved.



BONUS
DEPT
EMP
INV
INVENTORY
PLAN_TABLE
SALGRADE

Column	Data Type
EMPNO	NUMBER (4)
ENAME	VARCHAR2 (1
JOB	VARCHAR2 (9)
MGR	NUMBER (4)
HIREDATE	DATE
SAL	NUMBER (7,2)

FIGURE 3.19 Available tables in the SCOTT schema.



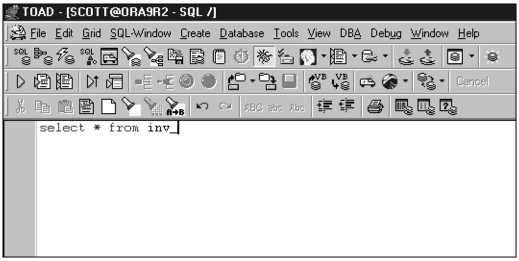
BONUS
DEPT
EMP
INV
INVENTORY
PLAN_TABLE
SALGRADE

Column	Data Type
EMPNO	NUMBER (4)
ENAME	VARCHAR2 (1
JOB	VARCHAR2 (9)
MGR	NUMBER (4)
HIREDATE	DATE
SAL	NUMBER (7,2)

FIGURE 3.20 Ignore the alias in action.

Autoreplace substitution replaces a short string with the full name. This differs from TOAD aliases because aliases make reference to a different name but do not change the text of the SQL statement. When autoreplace substitution is defined, it happens automatically when you press the spacebar. Autoreplace substitution is activated by typing the short sequence (illustrated in Figure 3.21) and pressing the spacebar (the autoreplace delimiter key defaults to the spacebar). This will then automatically substitute the predefined string in place of the short key sequence. See Figure 3.22.

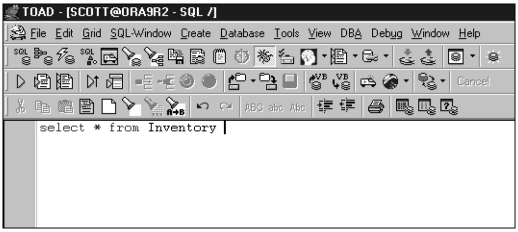
Replacement happens after pressing the spacebar.



```
select * from inv_
```

FIGURE 3.21 Autoreplace substitution key sequence.

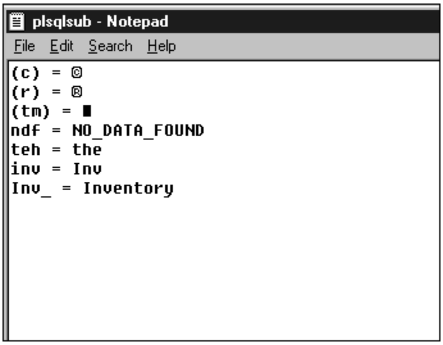
There are two ways to create automatic replacement substitution strings. You can edit the PLSQLSUB.TXT file in the `<`*`TOAD home directory`*`>\temps` directory. The format is the same as the alias: `<`*`short string`*`> = <`*`replacement string`*`>`. The other way is to enter the substitution string by using the Edit, Editor Options, Auto Replace tab. See Figure 3.24.


TOAD - [SCOTT@ORA9R2 - SQL 7]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel

```
select * from Inventory
```

FIGURE 3.22 Autoreplace substitution in action.

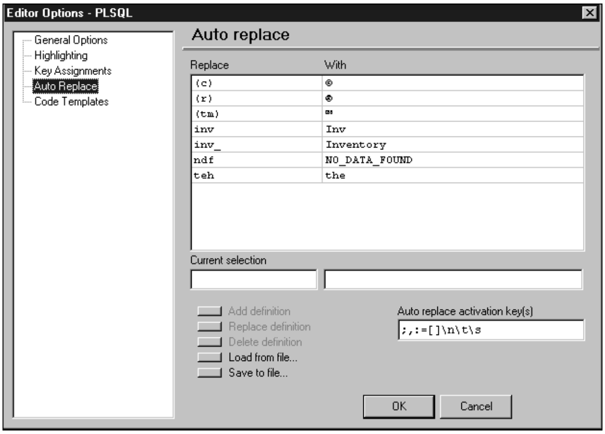
Figure 3.23 illustrates the contents of this file. Notice the common typo 'teh' will automatically be converted to 'the' because this sequence will always be followed by pressing the spacebar. Also notice the 'inv' and 'Inv_' strings that were used in Figure 3.21 and Figure 3.22.



```
(c) = ©
(r) = ®
(tm) = ▮
ndf = NO_DATA_FOUND
teh = the
inv = Inv
Inv_ = Inventory
```

FIGURE 3.23 Editing the PLSQLSUB.TXT file.

TOAD allows autosubstitution to be maintained by language type. You can edit and add to the list using the Editor Options, Auto Replace window. Supported languages for autoreplacement are HTML (`<`*`TOAD home`*`>\temps\HTMLSUB.TXT`), INI (`<`*`TOAD home`*`>\temps\INISUB.TXT`), JAVA (`<`*`TOAD home`*`>\temps\JAVASUB.TXT`), and TEXT (`<`*`TOAD home`*`>\temps\TEXTSUB.TXT`).



Replace	With
(c)	©
(r)	®
(tm)	™
inv	Inv
inv_	Inventory
ndf	NO_DATA_FOUND
teh	the

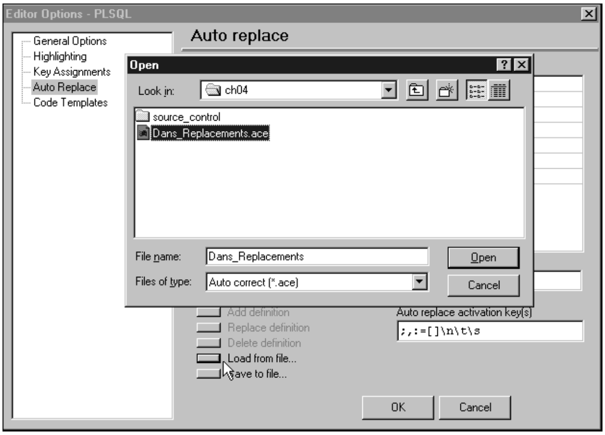
Current selection

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

FIGURE 3.24 Adding autoreplace substitution in the Editor Options screen.

Be sure TOAD is NOT running when you are editing any of these files in the TEMPS directory. TOAD rewrites these files when closing, and any changes made to the files with TOAD running will be lost.

TOAD allows you to export and import these autosubstitution definitions. This is a convenient way to move them when setting up a new computer to use TOAD. It also might be convenient to have various autosubstitution files by application. This allows you to have specific substitutions for various applications, same alias but resolved to different names perhaps. See Figure 3.25.



Look in: ch04

File name: Dans_Replacements

Files of type: Auto correct (*.ace)

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

FIGURE 3.25 Saving and loading autoreplace definitions.

When loading autoreplace definitions in from a saved file, all previous substitutions currently in TOAD are lost and replaced with the new definitions.

These .ACE files are in a binary format and are only to be used with the TOAD Load facility (from the Editing Options menu).

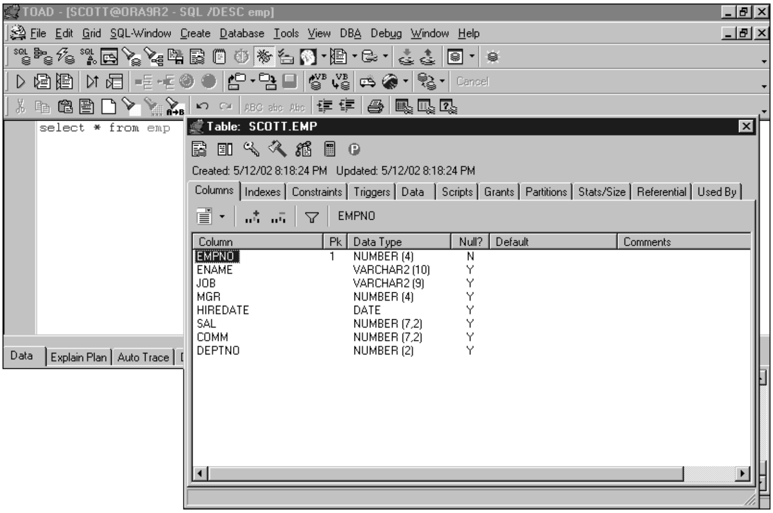
SQL Templates

TOAD can easily format a `SELECT` or `INSERT` statement for any data-oriented object (view or table). Simply place the cursor on the object you want and press F4. This will perform a description on the object, as illustrated in Figure 3.26.

Notice in Figure 3.26 that everything you need to know about the object appears in this window.

It is particularly nice to have TOAD build a script that created the object. This information comes from the data dictionary and accurately reflects the object that you are working with.

Most of this information is covered in other chapters. To get to the SQL Templates, use the Columns tab and right-click on any of the column names. This brings up another menu of mostly administrative functions (most of these features are covered in other chapters); see Figure 3.27. Notice where the cursor is (Generate Statement); you have the choice of building a SELECT or INSERT statement for this particular object. When making a selection, TOAD will tell you that the "script copied to the clipboard." Figure 3.28 shows the code in the TOAD SQL Editor that was pasted from the clipboard. Both the SELECT and the INSERT are illustrated.


TOAD - [SCOTT@ORA9R2 - SQL /DESC emp]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data Explain Plan Auto Trace
Table: SCOTT.EMP
Columns Indexes Constraints Triggers Data Scripts Grants Partitions Stats/Size Referential Used By
EMPNO

```
select * from emp
```

Created: 5/12/02 8:18:24 PM Updated: 5/12/02 8:18:24 PM

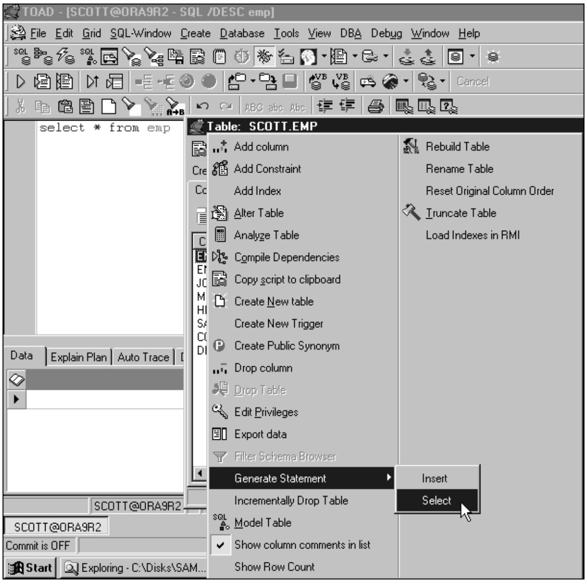
Column	Pk	Data Type	Null?	Default	Comments
EMPNO	1	NUMBER (4)	N		
ENAME		VARCHAR2 (10)	Y		
JOB		VARCHAR2 (9)	Y		
MGR		NUMBER (4)	Y		
HIREDATE		DATE	Y		
SAL		NUMBER (7,2)	Y		
COMM		NUMBER (7,2)	Y		
DEPTNO		NUMBER (2)	Y		

FIGURE 3.26 Object description.

F4 to describe the object.

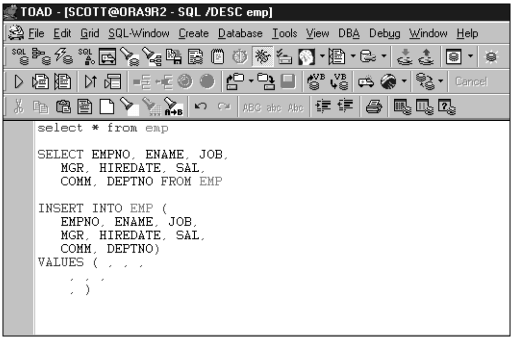
Right-click on a column in the Columns tab and select Generate Statement.

You can also use the Show SQL Template Window button (third toolbar, rightmost button) to paste in specific SQL templates while doing SQL coding. Figure 3.29 illustrates how to use these templates. Options include directly copying the template into the SQL window (as illustrated in Figure 3.29) or copying the template to the clipboard. Notice that there are all kinds of SQL templates. TOAD contains templates for row, group, and date functions. Also included are templates for PL/SQL coding, which will be covered in detail in the next chapter.



```
select * from emp
```

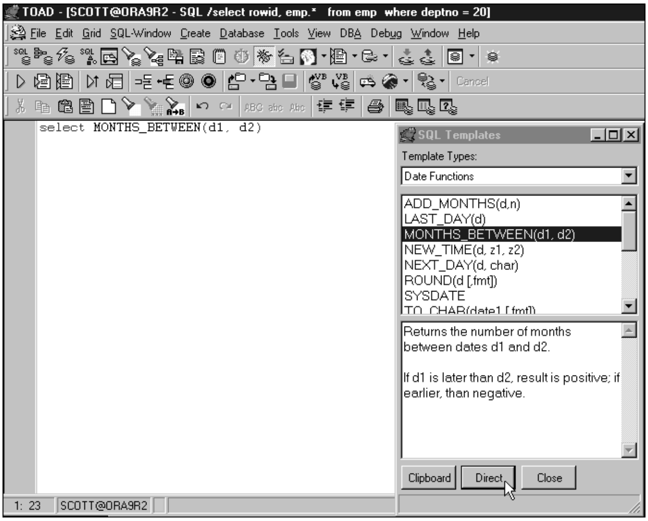
FIGURE 3.27 TOAD Generate Statement selection.



TOAD - [SCOTT@ORA9R2 - SQL /DESC emp]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel

```
select * from emp

SELECT EMPNO, ENAME, JOB,
   MGR, HIREDATE, SAL,
   COMM, DEPTNO FROM EMP

INSERT INTO EMP (
   EMPNO, ENAME, JOB,
   MGR, HIREDATE, SAL,
   COMM, DEPTNO)
VALUES ( , , ,
    , , ,
    , )
```

FIGURE 3.28 Generated SQL statements.


TOAD - [SCOTT@ORA9R2 - SQL /select rowid, emp.* from emp where deptno = 20]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
ABC abc Abc
SQL Templates
Clipboard
Direct
Close
1: 23
SCOTT@ORA9R2

```
select MONTHS_BETWEEN(d1, d2)
```

Template Types: Date Functions

ADD_MONTHS(d,n)
LAST_DAY(d)
MONTHS_BETWEEN(d1, d2)
NEW_TIME(d, z1, z2)
NEXT_DAY(d, char)
ROUND(d [,fmt])
SYSDATE
TO_CHAR(date1 [,fmt])

Returns the number of months between dates d1 and d2.

If d1 is later than d2, result is positive; if earlier, than negative.

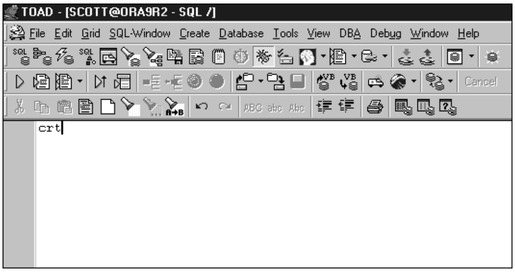
FIGURE 3.29 TOAD SQL templates in action.

Code Completion Templates

Code completion templates are boilerplate syntax where you type in the correct names, and so on, but the correct syntax with options is supplied.

Code completion templates work a lot like aliases except that they provide the entire syntax rather than just a word or object name. Figures 3.30 and 3.31 illustrate how this works. Instead of hitting the '.' to signal the replacement, you press Ctrl+Spacebar to signal the replacement.

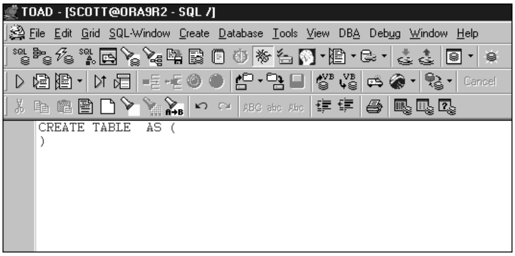
Ctrl+Spacebar signals code template.


TOAD - [SCOTT@ORA9R2 - SQL /]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
ABC abc Abc

```
crt
```

FIGURE 3.30 Code completion template key sequence.

Like the autoreplacement feature, additional code completion templates are easily added by using the Edit, Editor Options, Code Templates screen, as shown in Figure 3.32.

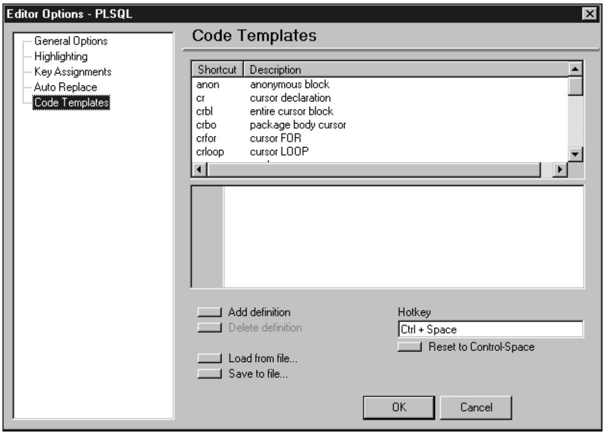

TOAD - [SCOTT@ORA9R2 - SQL 7]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help

```
CREATE TABLE  AS (
)
```

FIGURE 3.31 Code completion template in action.

You can also edit the PLSQL.DCI file in `<`*`TOAD Home Directory`*`>\temps` and add additional templates with a text editor. See Figure 3.33 for the layout example.

Supported languages for code completion are HTML (`<`*`TOAD home`*`>\temps\HTMLSUB.DCI`), INI (`<`*`TOAD home`*`>\temps\INISUB.DCI`), JAVA (`<`*`TOAD home`*`>\temps\JAVASUB.DCI`), and TEXT (`<`*`TOAD home`*`>\temps\TEXTSUB.DCI`).



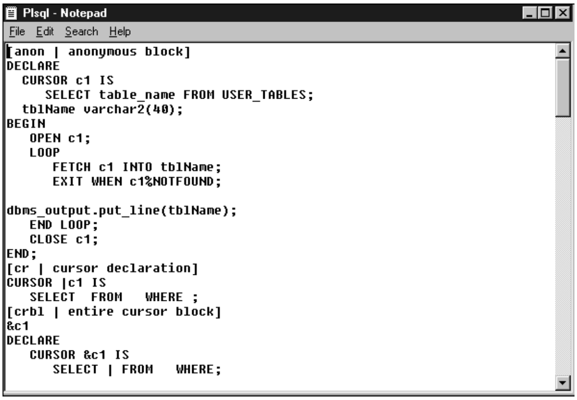
Code Templates

Shortcut	Description
anon	anonymous block
cr	cursor declaration
crbl	entire cursor block
crbo	package body cursor
crfor	cursor FOR
crloop	cursor LOOP

Hotkey: Ctrl + Space

FIGURE 3.32 Adding code completion templates.

Be sure TOAD is NOT running when you are editing any of these files in the `TEMPS` directory. TOAD rewrites these files when closing, and any changes made to the files with TOAD running will be lost.



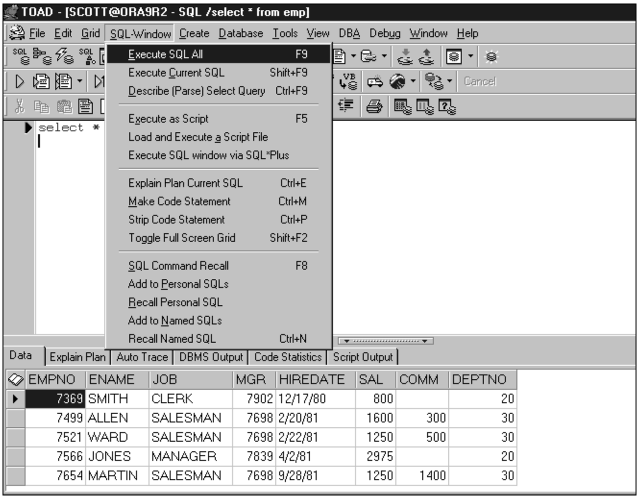
```
[anon | anonymous block]
DECLARE
  CURSOR c1 IS
     SELECT table_name FROM USER_TABLES;
  tblName varchar2(40);
BEGIN
   OPEN c1;
   LOOP
      FETCH c1 INTO tblName;
      EXIT WHEN c1%NOTFOUND;

dbms_output.put_line(tblName);
   END LOOP;
   CLOSE c1;
END;
[cr | cursor declaration]
CURSOR |c1 IS
   SELECT  FROM   WHERE ;
[crbl | entire cursor block]
&c1
DECLARE
   CURSOR &c1 IS
      SELECT | FROM   WHERE;
```

FIGURE 3.33 Code completion template file layout.

Executing SQL Statements

TOAD not only enables you to easily edit and create SQL and scripts containing SQL, but also to execute the SQL, review explain plans, and examine, edit, or change the result-set data.

The easiest way to execute SQL is with the Execute SQL button, the leftmost button on the middle toolbar. This executes the statement and returns the data when in full-view mode. Remember that F6 toggles between just the SQL Editor window and the results panel.


TOAD - [SCOTT@ORA9R2 - SQL /select * from emp]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Execute SQL All F9
Execute Current SQL Shift+F9
Describe (Parse) Select Query Ctrl+F9
Execute as Script F5
Load and Execute a Script File
Execute SQL window via SQL*Plus
Explain Plan Current SQL Ctrl+E
Make Code Statement Ctrl+M
Strip Code Statement Ctrl+P
Toggle Full Screen Grid Shift+F2
SQL Command Recall F8
Add to Personal SQLs
Recall Personal SQL
Add to Named SQLs
Recall Named SQL Ctrl+N
Cancel
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Output

```
select *
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	12/17/80	800		20
7499	ALLEN	SALESMAN	7698	2/20/81	1600	300	30
7521	WARD	SALESMAN	7698	2/22/81	1250	500	30
7566	JONES	MANAGER	7839	4/2/81	2975		20
7654	MARTIN	SALESMAN	7698	9/28/81	1250	1400	30

FIGURE 3.34 Executing SQL statements.

F9 executes all of the SQL statements in the SQL window, as does the Execute SQL All menu option, as in figure 3.34. Shift-F9 executes the SQL statement at the cursor, or the SQL statement that is highlighted.

Shift-F9 executes the current SQL statement.

If you are working with SQL in a script, highlight the SQL in the script and press F9 to execute the single SQL statement only.

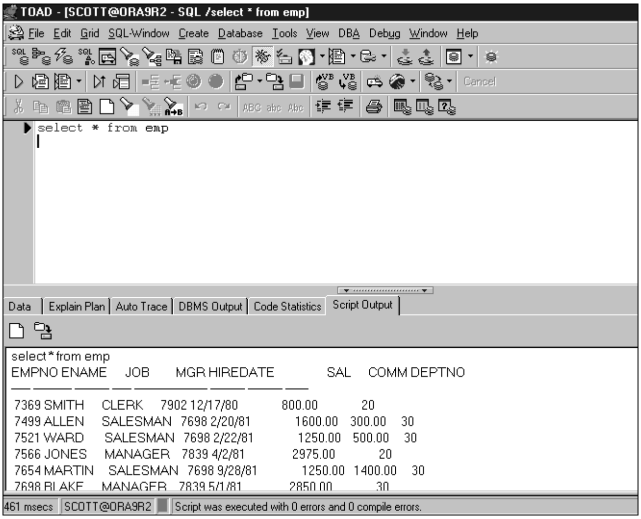
Executing SQL Scripts

TOAD allows you to execute SQL and SQL scripts as scripts. This will display the output of the script in the Script Output tab in the results panel; see Figure 3.35. This might be convenient when working with scripts so you don't have to exit the TOAD environment to run scripts in SQL*Plus, for example.

F5 executes the current SQL statement as a script.

The script output gives you the output that the script will produce, not just the data. Figure 3.36 shows a short SQL*Plus script and its output with column headings. SQL*Plus compatibility with TOAD is discussed later in this chapter.

If there are any syntax errors, TOAD will highlight the syntax problem in the SQL text and display the Oracle error across the bottom of the output area.

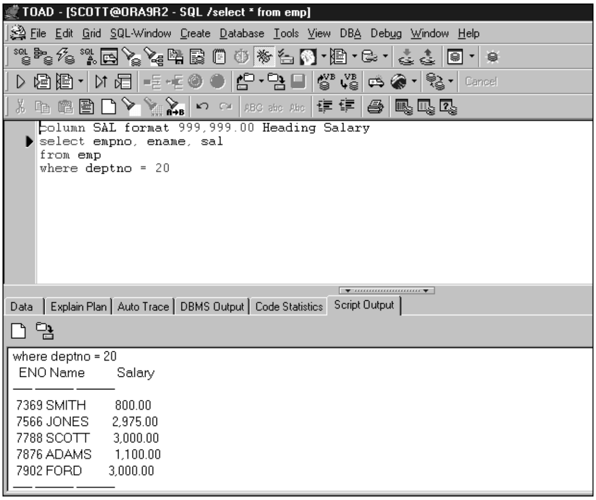

TOAD - [SCOTT@ORA9R2 - SQL /select * from emp]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
ABC abc Abc
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Output
461 msecs
SCOTT@ORA9R2

```
select * from emp
```

```
select * from emp
EMPNO ENAME    JOB      MGR HIREDATE          SAL    COMM DEPTNO
----- -------- -------- --- ----------- -------- -------- ------

 7369 SMITH    CLERK     7902 12/17/80       800.00          20
 7499 ALLEN    SALESMAN  7698 2/20/81         1600.00  300.00    30
 7521 WARD     SALESMAN  7698 2/22/81         1250.00  500.00    30
 7566 JONES    MANAGER   7839 4/2/81         2975.00          20
 7654 MARTIN   SALESMAN  7698 9/28/81         1250.00 1400.00    30
 7698 BLAKE    MANAGER   7839 5/1/81         2850.00          30
```

Script was executed with 0 errors and 0 compile errors.

FIGURE 3.35 Executing a SQL statement as a script.


TOAD - [SCOTT@ORA9R2 - SQL /select * from emp]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
ABC abc Abc
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Output

```
column SAL format 999,999.00 Heading Salary
select empno, ename, sal
from emp
where deptno = 20
```

```
where deptno = 20
 ENO Name       Salary
---- ------- -------

 7369 SMITH       800.00
 7566 JONES     2,975.00
 7788 SCOTT     3,000.00
 7876 ADAMS     1,100.00
 7902 FORD      3,000.00
---- ------- -------
```

FIGURE 3.36 Executing a SQL*Plus SQL script as a script.

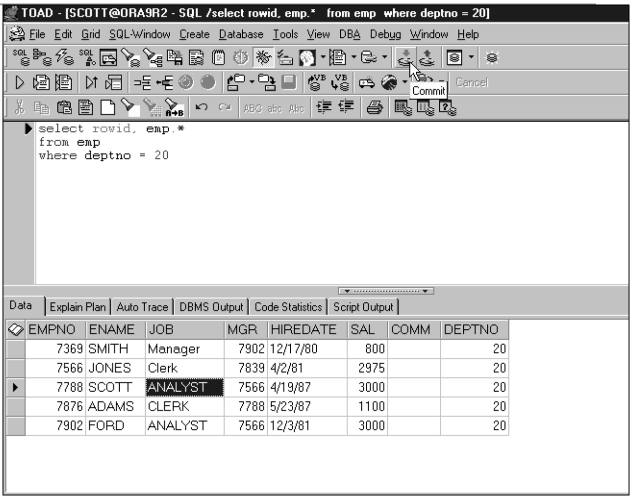
Editing Result-Set Data

TOAD allows you to change the data that appears in the Data tab of the results panel. The data columns being selected must include the ROWID column as well. Another way to select data from an object for update is by using `edit <object name>` syntax.

Figure 3.37 shows rows from the EMP table, demoting the manager Jones to a clerk and promoting the clerk Smith to manager.

To save the data changes, click the Commit button on the top toolbar (see mouse cursor in Figure 3.37).

Make sure you exit the column you have edited before committing the data.



TOAD - [SCOTT@ORA9R2 - SQL /select rowid, emp.*   from emp where deptno = 20]

```
select rowid, emp.*
from emp
where deptno = 20
```

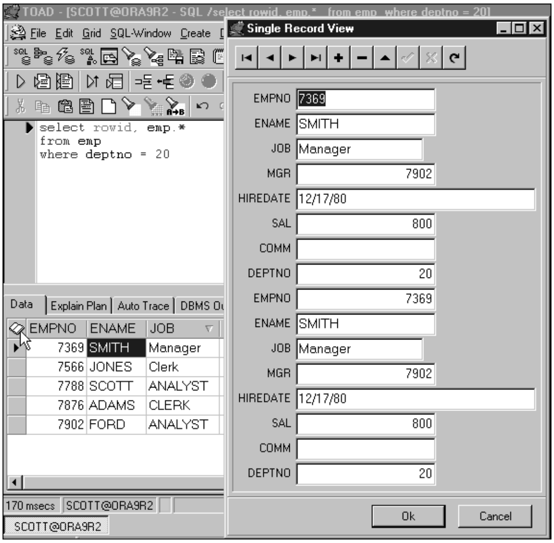
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	Manager	7902	12/17/80	800		20
7566	JONES	Clerk	7839	4/2/81	2975		20
7788	SCOTT	ANALYST	7566	4/19/87	3000		20
7876	ADAMS	CLERK	7788	5/23/87	1100		20
7902	FORD	ANALYST	7566	12/3/81	3000		20

FIGURE 3.37 Changing data values in the results panel.

You can also sort the data in the data grid by simply clicking on the columns and making your sort-order selections.

TOAD allows you to view all the data for a particular row. Click on the button that looks like a book, and the current row in the data grid will appear in a pop-up box. Figure 3.38 shows the data in the pop-up. Notice the mouse cursor is pointing to the icon that makes the pop-up appear. Also notice that this data is for the first row in the data grid because it is marked as the current row.

Character, blob/long raw, and date fields all have a memo editor that can help visualize ad change the data in the field. Right-mouse click and select Memo Editor from the popup menu. This capability will continue to be enhanced in future releases of TOAD.



```
select rowid, emp.*
from emp
where deptno = 20
```

EMPNO	ENAME	JOB
7369	SMITH	Manager
7566	JONES	Clerk
7788	SCOTT	ANALYST
7876	ADAMS	CLERK
7902	FORD	ANALYST

EMPNO 7369
ENAME SMITH
JOB Manager
MGR 7902
HIREDATE 12/17/80
SAL 800
COMM
DEPTNO 20
EMPNO 7369
ENAME SMITH
JOB Manager
MGR 7902
HIREDATE 12/17/80
SAL 800
COMM
DEPTNO 20

FIGURE 3.38 Viewing and changing single-row data values in the results panel.

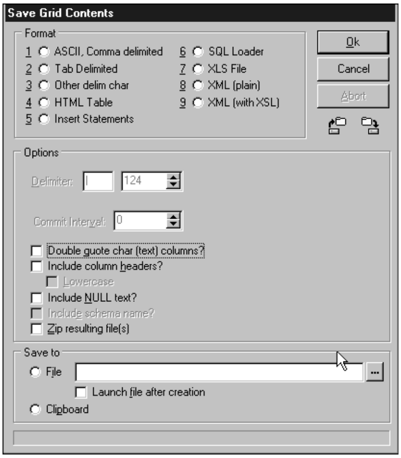
Saving Result-Set Data

The data in the results panel Data tab can easily be saved in a number of formats. Choose Grid, Save As and view the options available, as illustrated in Figure 3.39. You can create a delimited file, XML formatted file, and/or copy the results to the clipboard or to a named file.

Printing Result-Set Data

TOAD makes it easy to format the data grid into an attractive report. Select the Print Grid by either choosing Grid, Print Grid from the menu bar or by right-clicking on the data grid and selecting Print Grid. Both options bring up the Report Link Designer, as shown in Figure 3.40.

Right-clicking on the data grid gives many options including Print Grid.



Format
1 ○ ASCII, Comma delimited
2 ○ Tab Delimited
3 ○ Other delim char
4 ○ HTML Table
5 ○ Insert Statements
6 ○ SQL Loader
7 ○ XLS File
8 ○ XML (plain)
9 ○ XML (with XSL)

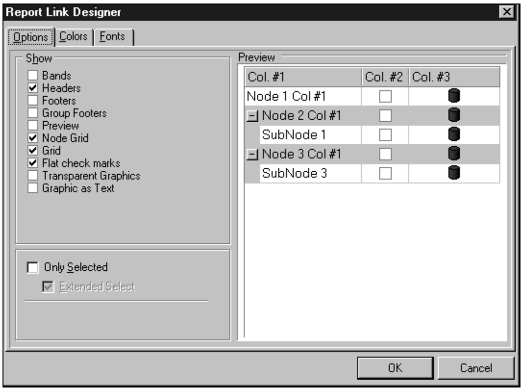
Options
Delimiter: | 124
Commit Interval: 0
□ Double quote char (text) columns?
□ Include column headers?
□ Lowercase
□ Include NULL text?
□ Include schema name?
□ Zip resulting file(s)

Save to
○ File
□ Launch file after creation
○ Clipboard

FIGURE 3.39 Saving data values in the results panel.

Change your data grid headings as desired for the report output. The Report Link Designer will allow you to add headers and footers, change the fonts, print only selected columns (right-click on the Data grid and check Preview Column or Remove Preview Column), and so on.

After selecting OK on this menu, you will be presented with the Print Options screen. From here, you can perform the standard print features, such as printing the whole report or only certain pages, and do page setup functions.



Report Link Designer

Options | Colors | Fonts

Show

- ☐ Bands
- ☑ Headers
- ☐ Footers
- ☐ Group Footers
- ☐ Preview
- ☑ Node Grid
- ☑ Grid
- ☑ Flat check marks
- ☐ Transparent Graphics
- ☐ Graphic as Text

- ☐ Only Selected
- ☑ Extended Select

Preview

Col. #1	Col. #2	Col. #3
Node 1 Col #1	☐	●
⊟ Node 2 Col #1	☐	●
SubNode 1	☐	●
⊟ Node 3 Col #1	☐	●
SubNode 3	☐	●

FIGURE 3.40 Toad Report Link Designer.

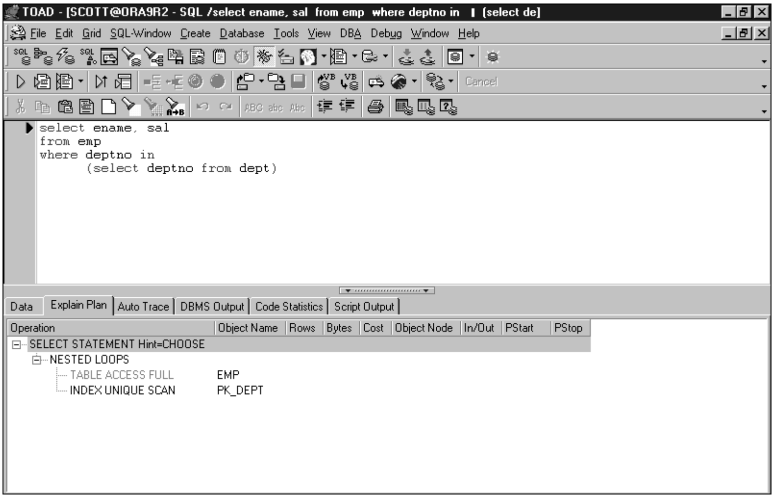
If you want column totals, choose Grid, Print Setup, select the column in the Columns tab, and then check the Total This Column check box. You can also change the column heading at this time.

Examining Explain Plans

TOAD allows you to easily see the explain plan for the currently executed SQL statement. This is visualized on the Explain Plan tab in the results panel. Figure 3.41 illustrates a rather simple explain plan.

Ctrl+E also runs and displays an explain plan.

It is beyond the scope of this book to provide a basic explanation of explain plans and the various features of the rule- and cost-based Oracle optimizers. TOAD does support changing the Optimizer Mode by right-clicking on the SQL statement and selecting the Optimizer Mode option. Cost-based hints can easily be added by using the SQL Templates option (discussed earlier in this chapter).



TOAD - [SCOTT@ORA9R2 - SQL /select ename, sal from emp where deptno in ▌ (select de]

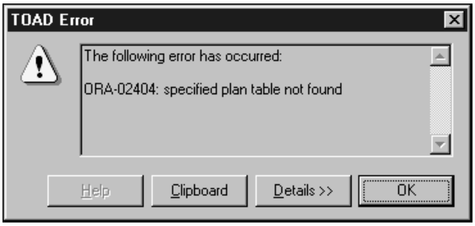
```
select ename, sal
from emp
where deptno in
      (select deptno from dept)
```

```
Operation                      Object Name  Rows  Bytes  Cost  Object Node  In/Out  PStart  PStop
SELECT STATEMENT Hint=CHOOSE
  NESTED LOOPS
    TABLE ACCESS FULL          EMP
    INDEX UNIQUE SCAN          PK_DEPT
```

FIGURE 3.41 TOAD explain plans.

Problem Solving the Explain Plan Tab

If you get the error `ORA-02404 Specified Plan Table Not Found` as shown in Figure 3.42, you have two options.



TOAD Error

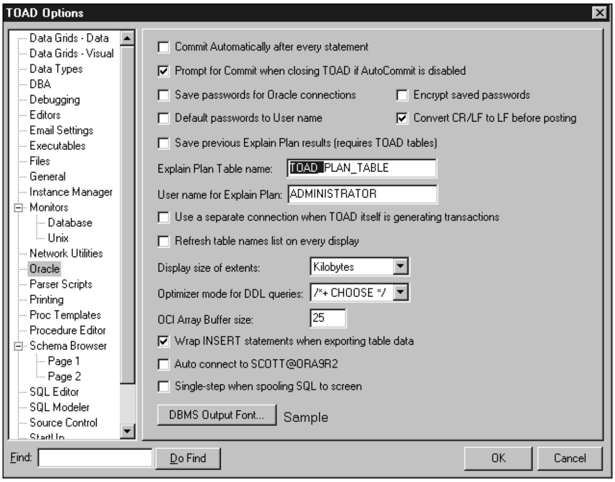
The following error has occurred:

ORA-02404: specified plan table not found

FIGURE 3.42 TOAD Explain Plan Not Found error.

The problem is that the TOADPREP.SQL script has not been run as described in Chapter 1, or the TOAD explain plan table has been renamed.

The solution is to run the script TOADPREP.SQL as defined in Chapter 1 and this, among other things, builds the TOAD_PLAN_TABLE. If you choose to use the Oracle RDBMS plan table (found in `<Oracle Home>\RDBMS\admin`) in file UTLXPLAN.SQL (this file is executed for each schema owner), click the TOAD Options button, and under the Oracle category, change the Explain Plan Table name to "PLAN_TABLE", removing the "TOAD_" from the beginning, as illustrated in Figure 3.43. This will allow TOAD to find the explain plan table for your schema.



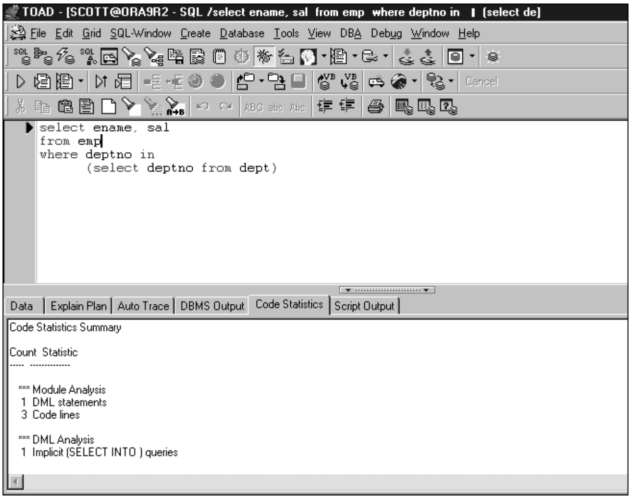
□ Commit Automatically after every statement
☑ Prompt for Commit when closing TOAD if AutoCommit is disabled
□ Save passwords for Oracle connections □ Encrypt saved passwords
□ Default passwords to User name ☑ Convert CR/LF to LF before posting
□ Save previous Explain Plan results (requires TOAD tables)
Explain Plan Table name: TOAD_PLAN_TABLE
User name for Explain Plan: ADMINISTRATOR
□ Use a separate connection when TOAD itself is generating transactions
□ Refresh table names list on every display
Display size of extents: Kilobytes
Optimizer mode for DDL queries: /*+ CHOOSE */
OCI Array Buffer size: 25
☑ Wrap INSERT statements when exporting table data
□ Auto connect to SCOTT@ORA9R2
□ Single-step when spooling SQL to screen

FIGURE 3.43 TOAD Explain Plan Definition window.

Examining Basic Performance Info

TOAD tracks basic information about the execution of the SQL statement or scripts. This information might be helpful in debugging certain issues with the SQL statement itself.

This information is on the results panel under the Code Statistics tab; see Figure 3.44.


TOAD - [SCOTT@ORA9R2 - SQL /select ename, sal from emp where deptno in | (select de]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Output

```
select ename, sal
from emp
where deptno in
      (select deptno from dept)
```

```
Code Statistics Summary

Count  Statistic
-----  ----------

 *** Module Analysis
  1  DML statements
  3  Code lines

 *** DML Analysis
  1  Implicit (SELECT INTO ) queries
```

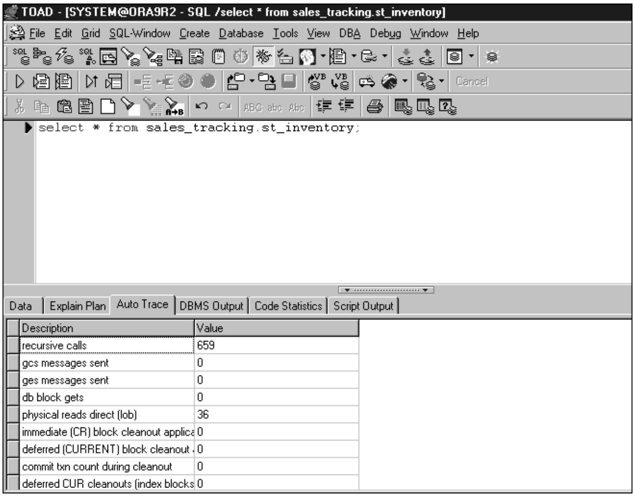
FIGURE 3.44 TOAD Code Statistics tab.

Using Auto Trace

TOAD will run the Oracle trace function for you, run TKPROF and display the important information in the Auto Trace tab of the results panel; see Figure 3.45.

If the Auto Trace feature is not turned on for your login session, TOAD will tell you that it is disabled and ask you if you want it enabled.

It is beyond the scope of this book to discuss all the various fields in this tab's output, but here are several of the most important fields:


TOAD - [SYSTEM@ORA9R2 - SQL /select * from sales_tracking.st_inventory]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
ABC abc Abc
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Output

```
select * from sales_tracking.st_inventory;
```

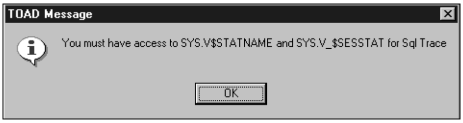
Description	Value
recursive calls	659
gcs messages sent	0
ges messages sent	0
db block gets	0
physical reads direct (lob)	36
immediate (CR) block cleanout applica	0
deferred (CURRENT) block cleanout	0
commit txn count during cleanout	0
deferred CUR cleanouts (index blocks	0

FIGURE 3.45 TOAD Auto Trace tab.

- Recursive calls: Oracle sometimes issues additional SQL statements on behalf of the running SQL statement. This is called recursive calls. Reasons include many extents on the object, dynamic space allocation (with an insert), and dictionary cache misses. The trace facility also generates recursive calls.
- Db block gets: This is the number of database block gets; it can be either physical or logical reads.
- Physical reads direct: This is the number of block fetch requests issued by Oracle.

It is beyond the scope of this book to go into great depth as to the meaning and interpretation of these various statistics.

The user running Auto Trace will need access to the V$STATNAME and V$SESSTAT dictionary views as illustrated in Figure 3.46.



You must have access to SYS.V$STATNAME and SYS.V_$SESSTAT for Sql Trace

FIGURE 3.46 TOAD Auto Trace required permissions.

SQL*Plus Compatibility

SQL*Plus is used for a variety of things in the Oracle environment. In the early days (version 5 and before), SQL*Plus was also used as an administrative tool. Today, SQL*Plus has taken back the Server Manager role. SQL*Plus has always been a good character-mode reporting tool, a tool used to create SQL-creating-SQL (next topic in this chapter), and so on. It is quite powerful with its formatting and file creation (spool commands) to perform a variety of tasks in the Oracle environment.

TOAD is an excellent tool for creating and maintaining these SQL*Plus scripts. The Script Output tab in the results panel is powerful in that you can see what the script has produced without having to leave TOAD and run the script in a separate window with SQL*Plus.

TOAD primarily supports the reporting features of SQL*Plus and not the administrative functions that have been given back to SQL*Plus.

Make sure to run any SQL*Plus script as a script!

TOAD supports these SQL*Plus commands:

`@` (“at” sign)

`@@` (double “at” sign)

`/` (slash)

`BREAK` (`BRE`) partial support

`CLEAR`

`COLUMN` (partial support, including: `ALIAS`, `NOPRINT`/`PRINT`, `NEW_VALUE`)

`COMPUTE` (partial support including: `SUM`, `MIN`, `MAX`)

`DEFINE`

`DESCRIBE`

`HEADING`

`JUSTIFY`

`SPOOL <Filename>`, `SPOOL OFF`, and `SPO`. Non-qualified files are created in the `<TOAD Home Directory>\TEMPS\` directory. Relative-location file creation is supported.

`SET AUTOTRACE`

`SET ECHO (ON/OFF)`

`SET ESCAPE (ON/OFF)`

`SET ESCAPE/ESC '\'` (`\` is any character)—defaults to `\`

`SET HEADING/HEA (ON/off)`

`SET FEEDBACK (ON/off)`

`SET LINESIZE`

`SET SERVEROUTPUT/SERVEROUT (ON/OFF)`

`SET TERM (ON/off)`

`DEFINE/DEF` and `UNDEFINE/UNDEF`

`PAUSE`

`EXIT/QUIT`

`PROMPT/PRO`

`CONNECT/CON` and D`ISCONNECT/DISC`.

`REM/REMARK`

`RUN`

TOAD simply ignores the following SQL*Plus commands:

`SET TAB`

`STORE SET`

`VERIFY`

`PAGESIZE`

and any other `SET` commands not supported by TOAD.

The following SQL*Plus commands are NOT supported in TOAD:

```
ACCEPT          APPEND            ARCHIVE
LOG
ATTRIBUTE       BTITLE/TTITLE
CHANGE          COPY              DEL
EDIT            EXECUTE           GET
HELP            HOST              INPUT
LIST            PASSWORD
```

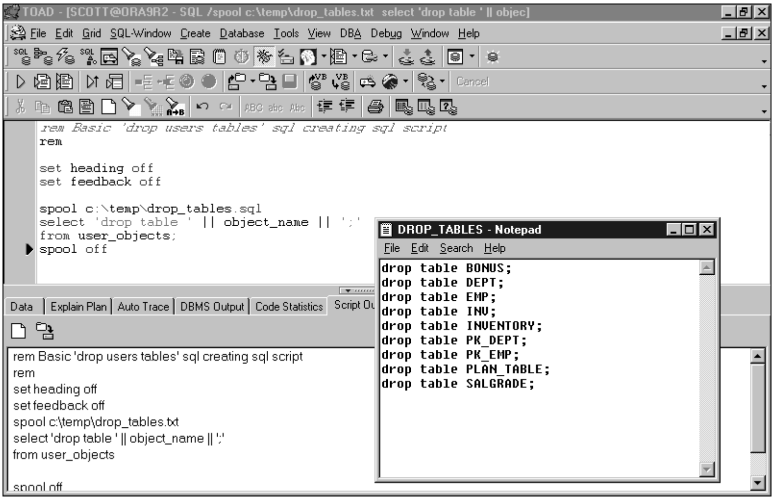
```
RECOVER                    REPHEADER/REPFOOTER
SAVE                       SHOW                       SHUTDOWN
START                      STARTUP
TIMING                     VARIABLE
WHENEVER OSERROR           WHENEVER SQLERROR
```

Scripts That Write Scripts

TOAD supports just about any kind of script. What makes Oracle so powerful is the ability to create scripts that actually build scripts!

Make sure to run any SQL*Plus script as a script!

The most basic SQL-creating-SQL is the DROP TABLE script. This script can be used to clean up a schema from a departing employee, for example. Figure 3.47 illustrates this simple script. Notice the `SPOOL` command. The output from this command appears in the Notepad window.


TOAD - [SCOTT@ORA9R2 - SQL /spool c:\temp\drop_tables.txt select 'drop table ' || objec]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Ou
DROP_TABLES - Notepad
File Edit Search Help

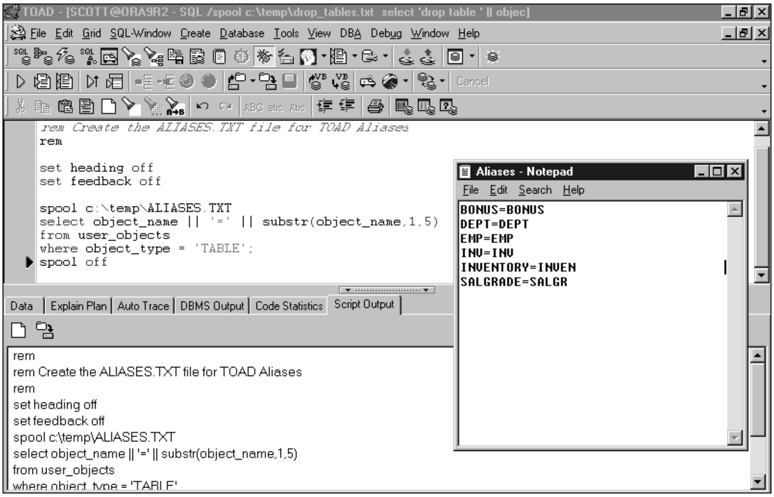
```
rem Basic 'drop users tables' sql creating sql script
rem

set heading off
set feedback off

spool c:\temp\drop_tables.sql
select 'drop table ' || object_name || ';'
from user_objects;
spool off
```

```
rem Basic 'drop users tables' sql creating sql script
rem
set heading off
set feedback off
spool c:\temp\drop_tables.txt
select 'drop table ' || object_name || ';'
from user_objects

spool off
```

```
drop table BONUS;
drop table DEPT;
drop table EMP;
drop table INV;
drop table INVENTORY;
drop table PK_DEPT;
drop table PK_EMP;
drop table PLAN_TABLE;
drop table SALGRADE;
```

FIGURE 3.47 DROP TABLE SQL-Creating-SQL script.

TOAD can also use SQL-Creating-SQL to create most of the TOAD configuration files such as the ALIASES.TXT file. Figure 3.48 shows this script. Notice that it spooled ALIASES.TXT to the `C:\TEMP` directory. This file would have to be moved to the *`<TOAD home directory>`*`\temps` directory. The `substr` can be adjusted.


TOAD - [SCOTT@ORA9R2 - SQL /spool c:\temp\drop_tables.txt select 'drop table ' || objec]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Window Help
Cancel
Data
Explain Plan
Auto Trace
DBMS Output
Code Statistics
Script Output
Aliases - Notepad
File Edit Search Help

```
rem Create the ALIASES.TXT file for TOAD Aliases
rem

set heading off
set feedback off

spool c:\temp\ALIASES.TXT
select object_name || '=' || substr(object_name,1,5)
from user_objects
where object_type = 'TABLE';
spool off
```

```
rem
rem Create the ALIASES.TXT file for TOAD Aliases
rem
set heading off
set feedback off
spool c:\temp\ALIASES.TXT
select object_name || '=' || substr(object_name,1,5)
from user_objects
where object_type = 'TABLE'
```

```
BONUS=BONUS
DEPT=DEPT
EMP=EMP
INV=INV
INVENTORY=INVEN
SALGRADE=SALGR
```

FIGURE 3.48 Creating the ALIASES.TXT file.

This technique of selecting data mixed with punctuation from USER_OBJECTS can be used to create most, if not all, of the TOAD configuration files.

Summary

This chapter covered using the TOAD SQL Editor, illustrating most of the features available. The real power in using TOAD is to get comfortable with the various keyboard shortcuts and to set up appropriate aliases and autoreplacements to aid the developer in the development cycle of SQL statements and SQL scripts.

The next chapter covers the TOAD Procedure Editor. Some of the features may appear similar to those features discussed in this chapter. In fact, some of the features are the same or similar. This book will make the distinction clear and use examples that are appropriate for the editor that the developer is using.