2 SQL*Plus

Introduction

SQL*Plus is the interactive (low-level) user interface to the ORACLE database management system. Typically, SQL*Plus is used to issue *ad-hoc* queries and to view the query result on the screen. Some of the features of SQL*Plus are:

- A built-in command line editor can be used to edit (incorrect) SQL queries. Instead of this line editor any editor installed on the computer can be invoked.
- There are numerous commands to format the output of a query.
- SQL*Plus provides an online-help.
- Query results can be stored in files which then can be printed.

Queries that are frequently issued can be saved to a file and invoked later. Queries can be parameterized such that it is possible to invoke a saved query with a parameter.

A Minimal User Guide

Before you start SQL*Plus make sure that the following UNIX shell variables are properly set (shell variables can be checked using the **env** command, e.g., **env** | **grep** ORACLE):

- `ORACLE_HOME`, e.g., `ORACLE_HOME=/usr/pkg/oracle/734`
- `ORACLE_SID`, e.g, `ORACLE_SID=prod`

In order to invoke SQL*Plus from a UNIX shell, the command **sqlplus** has to be issued. SQL*Plus then displays some information about the product, and prompts you for your user name and password for the ORACLE system.

```
gertz(catbert)54: sqlplus

SQL*Plus: Release 3.3.4.0.1 - Production on Sun Dec 20 19:16:52 1998

Copyright (c) Oracle Corporation 1979, 1996.  All rights reserved.

Enter user-name: scott
Enter password:

Connected to:
Oracle7 Server Release 7.3.4.0.1 - Production Release
With the distributed option
PL/SQL Release 2.3.4.0.0 - Production

SQL>
```


`SQL>` is the prompt you get when you are connected to the ORACLE database system. In SQL*Plus you can divide a statement into separate lines, each continuing line is indicated by a prompt such 2>, 3> etc. An SQL statement must always be terminated by a semicolon (;). In addition to the SQL statements discussed in the previous section, SQL*Plus provides some special SQL*Plus commands. These commands need not be terminated by a semicolon. Upper and lower case letters are only important for string comparisons. An SQL query can always be interrupted by using <Control>C. To exit SQL*Plus you can either type **exit** or **quit**.

Editor Commands

The most recently issued SQL statement is stored in the *SQL buffer*, independent of whether the statement has a correct syntax or not. You can edit the buffer using the following commands:

- `l[ist]` lists all lines in the SQL buffer and sets the current line (marked with an "∗") to the last line in the buffer.
- `l`<number> sets the actual line to <number>
- `c[hange]`/<old_string>/<new_string> replaces the first occurrence of <old_string> by <new_string> (for the actual line)
- `a[ppend]`<string> appends <string> to the current line
- `del` deletes the current line
- `r[un]` executes the current buffer contents
- `get`<file> reads the data from the file <file> into the buffer
- `save`<file> writes the current buffer into the file <file>
- `edit` invokes an editor and loads the current buffer into the editor. After exiting the editor the modified SQL statement is stored in the buffer and can be executed (command `r`).

The editor can be defined in the SQL*Plus shell by typing the command **define _editor** = <name>, where <name> can be any editor such as *emacs, vi, joe*, or *jove*.

SQL*Plus Help System and Other Useful Commands

- To get the online help in SQL*Plus just type **help** <command>, or just **help** to get information about how to use the **help** command. In ORACLE Version 7 one can get the complete list of possible commands by typing **help command**.
- To change the password, in ORACLE Version 7 the command
  alter user <user> **identified by** <new_password>;
  is used. In ORACLE Version 8 the command **passw** <user> prompts the user for the old/new password.
- The command **desc**[**ribe**] <table> lists all columns of the given table together with their data types and information about whether null values are allowed or not.
- You can invoke a UNIX command from the SQL*Plus shell by using **host** <UNIX_command>. For example, **host ls -la *.sql** lists all SQL files in the current directory.

- You can log your SQL*Plus session and thus queries and query results by using the command **spool** <file>. All information displayed on screen is then stored in <file> which automatically gets the extension `.lst`. The command **spool off** turns spooling off.
- The command **copy** can be used to copy a complete table. For example, the command
  copy from `scott/tiger` **create** `EMPL` **using select** $*$ **from** `EMP`;
  copies the table `EMP` of the user `scott` with password `tiger` into the relation `EMPL`. The relation `EMP` is automatically created and its structure is derived based on the attributes listed in the **select** clause.
- SQL commands saved in a file <name>.sql can be loaded into SQL*Plus and executed using the command **@**<name>.
- Comments are introduced by the clause **rem**[**ark**] (only allowed between SQL statements), or - - (allowed within SQL statements).

Formatting the Output

SQL*Plus provides numerous commands to format query results and to build simple reports. For this, format variables are set and these settings are only valid during the SQL*Plus session. They get lost after terminating SQL*Plus. It is, however, possible to save settings in a file named `login.sql` in your home directory. Each time you invoke SQL*Plus this file is automatically loaded.

The command **column** <column name> <option 1> <option 2> ... is used to format columns of your query result. The most frequently used options are:

- **format** A<n> For alphanumeric data, this option sets the length of <column name> to <n>. For columns having the data type **number**, the **format** command can be used to specify the format before and after the decimal point. For example, **format** 99,999.99 specifies that if a value has more than three digits in front of the decimal point, digits are separated by a colon, and only two digits are displayed after the decimal point.
- The option **heading** <text> relabels <column name> and gives it a new heading.
- **null** <text> is used to specify the output of null values (typically, null values are not displayed).
- **column** <column name> **clear** deletes the format definitions for <column name>.

The command **set linesize** <number> can be used to set the maximum length of a single line that can be displayed on screen. **set pagesize** <number> sets the total number of lines SQL*Plus displays before printing the column names and headings, respectively, of the selected rows.

Several other formatting features can be enabled by setting SQL*Plus variables. The command **show all** displays all variables and their current values. To set a variable, type **set** <variable> <value>. For example, **set timing on** causes SQL*Plus to display timing statistics for each SQL command that is executed. **set pause on** [<text>] makes SQL*Plus wait for you to press `Return` after the number of lines defined by **set pagesize** has been displayed. <text> is the message SQL*Plus will display at the bottom of the screen as it waits for you to hit `Return`.