Oracle's SQL*plus client

SQL*plus

SQL*plus is an interactive, command-line client for the Oracle DBMS. Upon login, you get a prompt at which you can enter any SQL command. The command will be immediately passed to the DBMS, and the results will be displayed back. In addition, SQL*plus has a number of internal commands that control its behavior. Some of these commands provide directions for output formatting, and work with SQL and SQL/PL scripts in SQL*plus's memory. Others enable the users to set up and run SQL and SQL/PL code from existing files in a batch mode.

Setup Instructions

These setup instructions refer to running SQL*plus from the CSL linux machines, such as the ones found in the database lab, 14-302.

All CSL machines have the following environment variable defined:

```
ORACLE_HOME=/app/oracle/product/10.2.0/client_1
```

This variable specifies the path to the directory where Oracle is installed on the system. The binaries for all Oracle software are located in

```
/app/oracle/product/10.2.0/client_1/bin
```

or

```
$ORACLE_HOME/bin
```

You can add `$ORACLE_HOME/bin` to your `PATH` environment variable (in `csh/tsch`, the command is `` setenv PATH `echo $PATH`:$ORACLE_HOME/bin ``).

In the instructions below, it is assumed that `$ORACLE_HOME/bin` is in your `PATH`.

sqlplus login

There are two Oracle DBMS servers running on the CS system. One of these servers, ora10g is used for instruction in our class. Each of you will be issued a loginid for this server. Your loginids will have the format STx or STxx where "x"/"xx" is a number between 1 and 50. Initial passwords will be issued to you. Upon your first login, you will be asked to change them.

Note: It should be obvious, but it bears repeating: your Oracle accounts (and passwords) are independent of yor CSC lab accounts.

To start SQL*plus, from the command line, type

```
> sqlplus
```

In the login prompt that appears enter your user id (e.g., "ST38") followed by the name of the Oracle server on which you are logging: ora10g. The dialog will proceed as follows:

```
> sqlplus

SQL*Plus: Release 10.2.0.1.0 - Production on Sun Sep 30 16:43:16 2007

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Enter user-name: ST38@ora10g
Enter password:
```

Enter your Oracle DBMS password at the prompt.

Once your login is authentificated, the SQL*plus prompt will appear. You many now begin your session.

SQL*plus commands

SQL statements

SQL*plus allows you to enter any valid SQL statement. Large SQL statements can be split into multiple lines (press *"Enter"* key and continue). To complete an SQL statement do one of the following:

- type ";" and hit "Enter". SQL*plus will send the SQL statement to the DBMS server immediately.
- type "/" on a line by itself. SQL*plus will send the SQL statement to the DBMS server immediately.
- hit "Enter" twice (i.e., end the statement with a blank line). SQL*plus will add the statement to its buffer, but will not execute it.

In the SQL*plus session shown below, the first two commands are executed immediately, creating the table and inserting one row into it, but the third command is not executed, as evidenced by the SELECT query[1].

```
SQL> create table classes
  2  ( Name CHAR(8) PRIMARY KEY,
  3  Department CHAR(20)
```

[1] we will learn more about SELECT queries shortly, but suffice it to say for now, that `SELECT * from <TableName>;` returns the list of all tuples currently stored in the table.

```
  4  );

Table created.

SQL> INSERT INTO classes
  2  VALUES('CPE 365','Computer Science')
  3  /

1 row created.

SQL> INSERT INTO classes
  2  VALUES ('CPE 366','Computer Science')
  3
SQL> SELECT * FROM classes;

NAME     DEPARTMENT
-------- --------------------
CPE 365  Computer Science

SQL>
```

SQL*plus general commands

SQL*plus is not case-sensitive. For example, DESCRIBE, describe and Describe is the same command.

Getting help.

`SQL>HELP`

Typing `HELP` results in a help screen suggesting the usage of the sqlplus built-in help system. To find the list of special sqlplus commands, try

```
SQL> help index

Enter Help [topic] for help.

 @             COPY         PAUSE                    SHUTDOWN
 @@            DEFINE       PRINT                    SPOOL
 /             DEL          PROMPT                   SQLPLUS
 ACCEPT        DESCRIBE     QUIT                     START
 APPEND        DISCONNECT   RECOVER                  STARTUP
 ARCHIVE LOG   EDIT         REMARK                   STORE
 ATTRIBUTE     EXECUTE      REPFOOTER                TIMING
 BREAK         EXIT         REPHEADER                TTITLE
 BTITLE        GET          RESERVED WORDS (SQL)     UNDEFINE
 CHANGE        HELP         RESERVED WORDS (PL/SQL)  VARIABLE
 CLEAR         HOST         RUN                      WHENEVER OSERROR
 COLUMN        INPUT        SAVE                     WHENEVER SQLERROR
 COMPUTE       LIST         SET
 CONNECT       PASSWORD     SHOW
```

To get help about specific command, try

```
HELP <command>
```

Changing password. `PASSWORD` or `PASSW` changes the Oracle DBMS password.

Retrieving database schema information. DESCRIBE command allows the users to see the relational schema of any table stored in the database, to which current user has access. Type

```
SQL>describe <TableName>
```

For example:

```
SQL> describe classes
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL CHAR(8)
 DEPARTMENT                                         CHAR(20)
```

Running SQL, SQL/PL and SQL*plus commands from a system file. Any sequence of SQL, SQL/PL and SQL*plus commands can be recorded into a file (typically with an .sql extension). SQL*plus allows the user to either immediately execute such commands, or load the contents of the file into the buffer.

To execute commands stored in a file, type

```
SQL>start <filename.ext>
```

where `filename.ext` is the file which contains the commands. For example, consider the following file `t.sql`:

```
describe classes

select * from classes;
```

Executing it in SQL*plus yields the following session:

```
SQL> start t.sql
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL CHAR(8)
 DEPARTMENT                                         CHAR(20)


NAME     DEPARTMENT
-------- --------------------
CPE 365  Computer Science
```

Other important SQL*plus commands are:

- host: execute an operating system command.
- remark or rem: comment.
- spool: redirection of output. `spool <filename>` redirects (or, actually, duplicates) SQL*plus output into the given file. `spool off` ends output redirection(duplication).
- execute: executes a single SQL/PL statement (we will be needing it later in the course).

Buffer manipulation

SQL*plus remembers the last executed SQL command in it buffer. It provides the user with the opportunity to view, execute, modify, and clear the buffer. The appropriate commands are (see also Figure 2.1, page 24 of the textbook):

Command	Shorthand	Explanation
run	r	execute current contents of the buffer.
list	l	display contents of the buffer.
clear buffer	cl buf	clear the contents of the buffer
get <filename>		input the contents of the file into the buffer
save <filename>	sav <filename>	save the contents of the buffer into a file.
input	i	add line(s) to the buffer
del		delete a line from the buffer
append *text*	a *text*	add text to the end of current line
change */old/new*	c */old/new*	replace text in the buffer
change */old*	c */old*	delete text from the buffer

Formatting of output in SQL*plus

SQL*plus allows the users to modify the default format in which data retrieved from the database is displayed. This is done by (a) setting a number of parameters controlling the formatting of the output and (b) specifying for each column its specific format.

Setting SQL*plus environment variables. SQL*plus has a set command which allows the user to specify a wide range of parameters controlling the behavior of SQL*plus. The format of the command is:

```
set <parameter> <value>
```

Full documentation about the set command and its parameters can be found here:

`http://download-west.oracle.com/docs/cd/B12037_01/server.101/b12170/ch13.htm#sthref2477`

(note, it refers to Oracle 8i, but most options persist from that version).

Some nice to know parameters are:

Parameter name	Values	Explanatation
echo	on | off	list all commands when executing from a file
null	*text*	text that represents the NULL value
colsep	_|*text*	column separator character(s)
underline	-|OFF|ON|*char*	character used to underline column names
headsep	ON|OFF|*char*	heading separator charater
recsepchar	*char*	record separator character
feedback	*n*|ON|OFF	display `k rows selected` if $k \geq n$
heading	ON|OFF	display/suppress display of column headers
linesize	n	length of output lines
embedded	OFF|ON	reports start only at new pages (OFF) or anywhere (ON)
newpage	NONE|*n*	number of blank lines printed at the top of each new page
pagesize	n	number of line in a page
nuformat	*format*	format for displaying numbers
numwidth	n	width of number columns
timing	ON|OFF	control of timing display

Setting breaks. SQL*Plus provides BREAK command to control changes in the output of queries. In particular, it helps suppress duplicate values in the same column, and print a blank line if there is a change in the column value.

BREAK with no parameters lists current BREAK specificiation. The full command has the following syntax:

```
BRE[AK] [ON <element> [<action>]*]
```

Here, `<element>` is one of the following: `<columnName>`|`<expression>`|`ROW`|`REPORT`.

`<action>` is one of `[SKI[P] n|[SKI[P]] PAGE] [NODUP[LICATES]|DUP[LICATES]]`.

`ON <column>` specifies the column whose value change will trigger the break. The break can cause one of the following actions:

- `SKIP` n : insert n blank lines
- `SKIP PAGE`: insert a pagebreak
- `NODUP[LICATES]`: will print blanks instead of duplicate column values
- `DUP[LICATES]`: will print all values, including duplicate

The session below illustrates the difference in the output achieved by the BREAK command:

```
SQL> clear breaks
breaks cleared
SQL> select * from classes;

NAME       DEPT
---------- --------------------
CPE 366    Computer Science
CPE 101    Computer Science
STAT 217   Statistics
STAT 252   Statistics
CE 111     Civil Engineering
CE 201     Civil Engineering
CE 337     Civil Engineering

SQL> break on Dept skip 1 nodup
SQL> run
  1* select * from classes

NAME       DEPT
---------- --------------------
CPE 366    Computer Science
CPE 101

STAT 217   Statistics
STAT 252

CE 111     Civil Engineering
CE 201
CE 337
```

Clearing current settings. clear command will remove some of the current settings:

- clear bre[aks] removes break definition.
- clear buff[er] clears text from the buffer.
- clear col[umns] clears all formatting instructions set by column command.
- clear scr[een] clears screen.
- clear SQL clears text from the (SQL) buffer.
- clear timi[ng] clears all timers.

Formatting columns. column command allows the user to set up the desired way in which a column is to be presented, when results of queries are reported. The formet of the command is described in detail at

```
http://download-west.oracle.com/docs/cd/B12037_01/server.101/b12170/ch13.htm#i2697128
```

We look at only a few options that can be set by this command:

```
column <Name> heading <heading> format <format>
```

<heading> is the text which would replace the default heading of the column.

<format> is a textual description of the column format. The following some of possible formats:

Type	Format	Explanation
Strings (CHAR(), etc.)	An	sets column witdth to n (alphanumeric characters)
DATE	An	
Number	9999	four-digit number
	9,999	comma to separate thousands
	99.99	decimal point
	$999	dollar sign in front of number
	0999 or 9990	leading or trailing zeros
	B999	blanks for the integer part of a fixed-point number when it is zero (.34 instead of 0.34)
	9.99EEEE	uses scientific notation
	999MI	tailing minus sign for negative numbers
	999PR	negative values in angle brackets
	RN or rn	upper/lowercase Roman numerals
	999S or S999	trailing or leading minus/plus sign
	XXXX or xxxx	hexadecimal value

Working with Oracle's system tables

To see what tables are available in your databse:

```
SELECT TABLE_NAME FROM USER_TABLES;
```

(you can also try SELECT * FROM USER_TABLES, but the output will be ugly).

To see what tables are available in all accessible databases:

```
SELECT TABLE_NAME, OWNER FROM ALL_TABLES;
```

This outputs the name of the table and its owner. Typically, the list will include all tables you have created plus any system tables used by Oracle DBMS to “run the show”.