Table of Contents

SQL*Plus is the traditional client used to work with an Oracle database that provides access to your data and lets you create and run SQL statements. You need to be familiar with using SQL*Plus even if you often use more graphical tools.

1. The login screen from Linux

Change to the directory you are using to store your scripts.

The login command is sqlplus; you can include your user name on the command line.

```
The Oracle base remains unchanged with value /oracle/app
ORACLE_HOME is /oracle/app/product/12.1.0
ORACLE_SID is cs12
$ sqlplus rendres

SQL*Plus: Release 12.1.0.1.0 Production on Sat Jun 21 17:42:47 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Wed Apr 30 2014 15:56:55 -07:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing opt
ions

SQL>
```

2. Exiting SQL*Plus

The session ends when you log out of SQL*Plus. To quit SQL*Plus, use the command **exit**

3. Entering an SQL statement

One way to enter an SQL query is by typing the SQL statement followed by a semicolon. The `SQL>` in the line below is the system prompt; you do not type that.

```
SQL> select sysdate from dual;

SYSDATE
---------
10-JAN-16

SQL>
```

Notice that the output column headers are displayed in uppercase letters. The column headers may be truncated to fit in the column width defined in the table.

To enter an SQL statement from the SQL*Plus environment, simply type it at the SQL> prompt. The SQL statement can extend over several lines and does not use a continuation character. You can terminate the SQL statement in three ways:

- End the last line with a semicolon. The SQL statement will be executed. The semicolon can be on a line all by itself. For the examples, I have generally ended the SQL statement with a semicolon.
- Enter a slash on a line all by itself. The SQL statement will be executed.
- Enter a blank line. The SQL statement will not be executed, but will stay in the SQL buffer. This is helpful if you notice that you made a mistake and want to re-enter the command.

To execute an SQL statement that is in a buffer, you can

- use a forward slash on a new line
- the run command (this also displays the buffer)

4. Command line interpretation

This section will help you understand some error situations - don't worry about it too much right now.

There are three types of commands you can enter at the SQL*Plus prompt:

- SQL*Plus commands
- SQL commands/statements
- PL/SQL blocks

SQL*Plus determines which type of command you are entering by examining the start of the command line. It sends SQL statements to the SQL engine and PL/SQL blocks to the PL/SQL engine. If the command line is not SQL or PL/SQL, then it is executed by SQL*Plus.

For example:

```
set echo on;        This is an SQL*Plus command
select sysdate from dual;        This is an sql command
```

We will see some PL/SQL later.

Some command lines include both SQL*Plus components and SQL components—in that case, SQL*Plus does its work before sending the SQL or PL/SQL to the database engines.

The results of an SQL statement is returned to SQL*Plus for display. Environment settings in SQL*Plus will influence the final display of the result.

SQL*Plus maintains a buffer called the SQL buffer. It contains the most recent SQL statement that you entered (or the most recent PL/SQL block). Executing the SQL statement does not empty the SQL buffer. SQL*Plus commands do not get stored in the SQL buffer.

You can clear the SQL buffer with the SQL*Plus command CLEAR BUFFER

You can clear the screen buffer with the SQL*Plus command CLEAR SCREEN

This will make more sense as the semester goes on- for now just understand that in the SQL*Plus client you can enter some commands the client executes and some that the database server executes.

5. Some useful SQL*Plus variables

Your SQL & PL/SQL statements work within the SQL*Plus environment and the output that you see is influenced by both the SQL statement and the SQL*Plus environment. You can modify how your SQL*Plus environment works by setting system variables. The values of the system variables remain until you terminate your session; therefore they continue to affect subsequent statements that you run in that session

You can give the SQL*Plus command SHOW ALL to see the values for all of your system variables. You can give the SQL*Plus command SHOW varbName to see the current setting for a particular variable.

We are going to start with only a few SQL*Plus variables that affect the spool files you will produce for assignments.

The usage of these commands is: SET FEEDBACK 15

The Set command, followed by the variable name and any option

System Variable	Values	Description
ECHO	ON OFF	Echo causes the commands in your script files to be displayed on the screen (and put in the spool file). We will use this for assignments.
FEEDBACK	n ON OFF	Feedback produces a line after the SQL statement runs that states how many rows were displayed by the query. The default value is 6; if your query produces fewer than 6 output rows, then the feedback line is not displayed. For assignments, set feedback to 1
PAGESIZE	n	Pagesize sets the number of lines per page. This determines how many lines will be displayed on the screen before the column headers are repeated. This is a physical line count, not a row count. So it includes column headers and blank lines as well as lines that display data. The default value is 24. For assignments, set pagesize to 999
TRIMSPOOL	ON OFF	Trimspool handles some of the problems with excess spaces at the end of the output lines. For assignments this should be set ON.
LINESIZE	n	The Linesize setting controls the number of characters SQL*Plus displays on a physical line. This can help prevent linewraps with long display lines.
TAB	ON OFF	SQL*Plus can keep spaces in the output or replace them with tab characters. For assignments this should be set OFF.

For our script assignments I want the following SQL*Plus statements included at the top of each script. For the page size, I do mean the literal 999. These commands are in the template you use for assignments.

```
set echo on
set feedback 1
set pagesize 999
set trimspool on
set linesize 200
set tab off
clear columns
```

6. SQL*Plus commands

SQL*Plus was designed with a series for commands that could be used for simple report formatting. Most report formatting should be done in another tool and is not covered in this class. Therefore we need very few SQL*Plus commands.

We have seen a few SQL*Plus command so far: exit, set, clear.

- SQL*Plus commands are not case specific.
- Most SQL*Plus command are short, but if you need to extend an SQL*Plus command over two lines use the hyphen (-) as the continuation character.
- Many SQL*Plus commands keywords have an abbreviated version.

6.1. Desc

Suppose you have a table you created a few weeks ago and you want to see the names of the columns and their data types. You can use the SQL*Plus command Describe (desc). We will create this table in this unit- see document First Table.

```
SQL> desc zoo_2016;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------

 Z_ID                                      NOT NULL NUMBER(38)
 Z_NAME                                             VARCHAR2(25)
 Z_TYPE                                    NOT NULL VARCHAR2(25)
 Z_COST                                    NOT NULL NUMBER(7,2)
 Z_DOB                                     NOT NULL DATE
 Z_ACQUIRED                                NOT NULL DATE

SQL> _
```

This gives you a quick view of the table structure- showing the names of the columns, if they are nullable or not (more on this later in the semester) and their data type. If you compare this to the Create table statement, you will see a few differences:

- Some data type names are different- I use the type Integer and internally Oracle Database changed that to Number(38)
- The constraint I set up for the z_cost value is not shown here.

6.2. Column

The column command in SQL*Plus has a lot of options. For assignment scripts, you may use only three of these options. In the zoo_2016 table we have a column for the animal name that was set up to store 25 characters. And you have already seen that our rows are rather wide. I decide that I want the animal names to be in a column of width 10. I can set this up as an SQL*Plus column command as shown here

```
column z_name format a10
column z_name format a10 word_wrapped
column z_name format a10 trunc
```

I provide the column name and a format- here the format is A10 which says this is a width 10 alphanumeric(string) column. Note that this does not use the table name.

But what happens if I have an animal named 'Sally Robinson'- that is longer than 10 characters. I can use a second option word_wrapped to say that I want the data wrapped with line breaks between words. And there is a third option Trunc

The following shows a Select command with three different column options. For class you do not need to worry about these options but they can be helpful to make the output more readable.

-- this is the default with 25 spaces for the name

```
SQL> select z_id, z_name, z_type
  2  from zoo_2016
  3  where z_id in (52,85);

      Z_ID Z_NAME                    Z_TYPE
---------- ------------------------- -------------------------
        85 Sally Robinson            Giraffe
        52 Dewey                     Giraffe

2 rows selected.

SQL> column z_name format a10
SQL> /

      Z_ID Z_NAME     Z_TYPE
---------- ---------- -------------------------
        85 Sally Robi Giraffe
           nson

        52 Dewey      Giraffe

2 rows selected.

SQL> column z_name format a10 word_wrapped
SQL> /

      Z_ID Z_NAME     Z_TYPE
---------- ---------- -------------------------
        85 Sally      Giraffe
           Robinson

        52 Dewey      Giraffe

2 rows selected.

SQL> column z_name format a10 trunc
SQL> /

      Z_ID Z_NAME     Z_TYPE
---------- ---------- -------------------------
        85 Sally Robi Giraffe
        52 Dewey      Giraffe

2 rows selected.
```

6.3. Clear columns

This SQL*Plus command will clear the columns settings you have in your session. You are required to use this in your scripts (it is in the template) and that means that if you use any column commands, they need to be in the script after this command.