Oracle (PL/SQL)

Lesson 08: SQL * Plus Reports

Lesson Objectives

To understand the following topics:

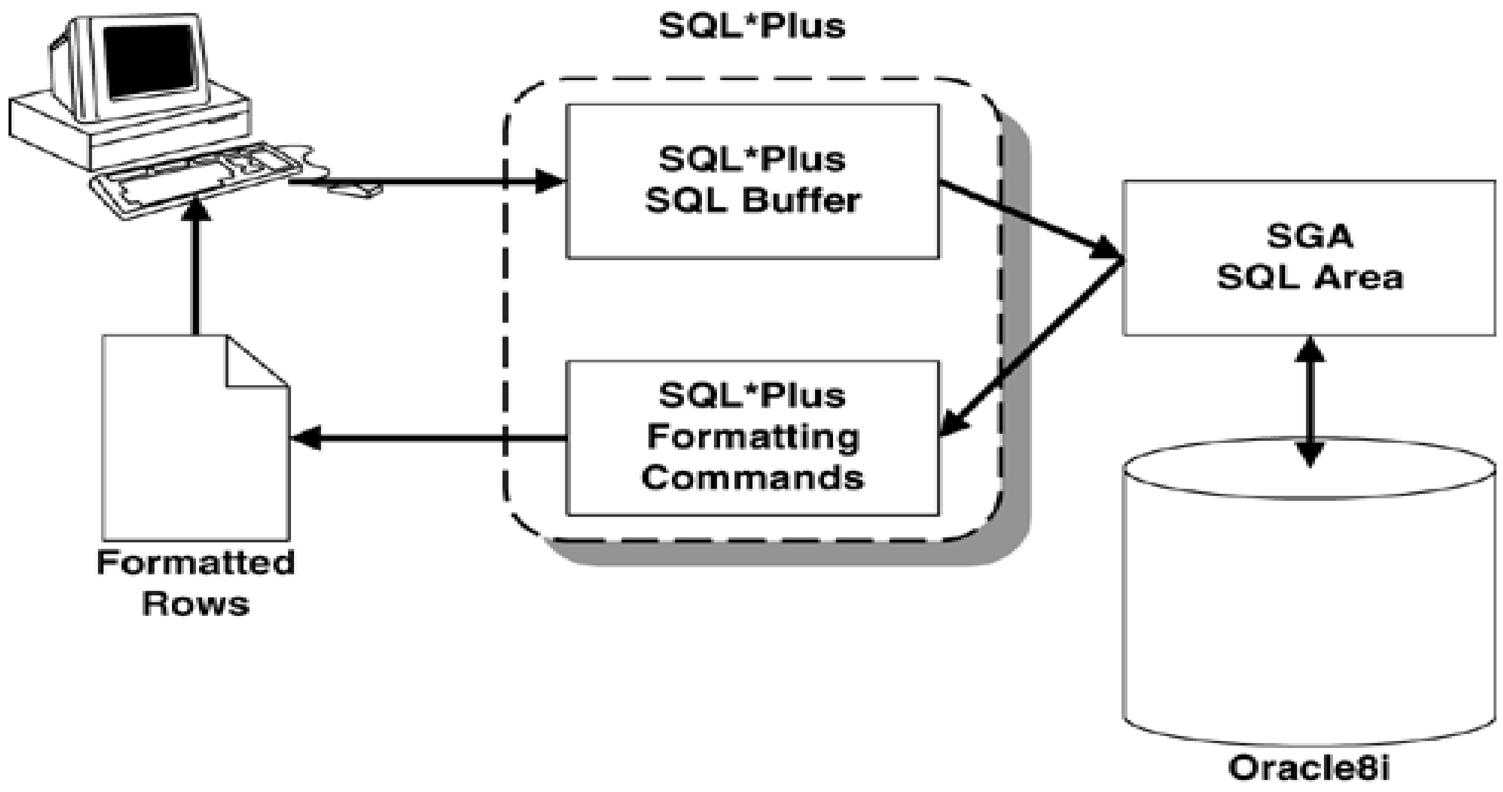
- SQL*Plus reporting Commands
- Generation of SQL Reports with different formats

Overview

SQL *Plus is an interactive tool for the Oracle RDBMS environment.

- You can use SQL *Plus:
  - to process SQL statements one at a time,
  - to process SQL statements interactively with end users,
  - to use PL/SQL for the procedural processing of SQL statements,
  - to list and print query results,
  - to format query results into reports,
  - to describe the contents of a given table, and
  - to copy data between databases

Reporting in SQL *Plus



SQL *Plus Commands

There are six types of SQL*Plus commands:

- Commands that initiate the SQL*Plus environment
- SQL*Plus execute commands
- SQL*Plus editing commands
- SQL*Plus formatting commands
- Miscellaneous commands
- Access commands for various databases

Commands to initiate SQL *Plus environment

SQL*Plus is an interactive, ad hoc environment that also can be preprogrammed with the use of SQL*Plus commands, SQL statements, and PL/SQL blocks submitted via a file

After successfully logging on to SQL*Plus, the user, regardless of the environment he or she is using, receives a SQL *Plus prompt: SQL>

You can change this prompt message to any text string by changing the SQL *Plus system variable SQLPROMPT

Usage of Execute Commands

You can use the execute commands to:

- Initiate the processing of SQL statements and PL/SQL blocks,
- Measure the processing time of SQL or PL/SQL statements,
- Execute non-Oracle programs,
- Execute SQL*Forms programs, or
- Obtain additional help

Execute Commands

The following table lists the execute commands:

Command	Description
/	Executes the SQL statement or PL/SQL block currently in the SQL buffer (This is probably the most-used of the SQL *Plus commands).
HELP topic	Provides online assistance with SQL, PL/SQL, or SQL *Plus commands.
HOST	Provides online assistance with SQL, PL/SQL, or SQL *Plus commands.
RUN	Displays and executes the contents of the SQL buffer.
TIMING	Displays the system CPU time with the SQL prompt.

Usage of Editing Commands

The SQL buffer is a work area assigned to the SQL*Plus environment.

- This buffer contains only SQL or PL/SQL syntax.

You can use the editing commands to load, save, and manipulate the contents of this buffer.

Usage of Formatting Commands

You use the SQL *Plus formatting commands to manipulate the result set from a SQL query

The formatting commands follow in the subsequent slides

Formatting Commands

BREAK ON column_name and options:

- This command controls the organization of rows returned by the query
- BREAK can manipulate the appearance of the output by specifying under what conditions a BREAK should occur and what actions should be taken at the BREAK
- The appearance of the output can be controlled by skipping a line or skipping to the top of the next page and providing totals when used with COMPUTE

Formatting Commands

BTITLE print_options and / or text or variable options:

- BTITLE places text at the bottom of each page.
  - You can use various print options to position text at various locations
  - BTITLE simply centers the text if no print options are specified
- Print options include BOLD, CENTER, COL, FORMAT, LEFT, RIGHT, SKIP, and TAB
- BTITLE spelled out by itself, displays the current text setting
- Other options that you can specify are ON and OFF
  - BTITLE is ON by default

Formatting Commands

COLUMN column_name and options:

- COLUMN alters the default display attributes for a given column (column_name) of a SQL query
- You can use a variety of options. However, the more common ones are FORMAT, HEADING, JUSTIFY, NEWLINE, NEW_VALUE, and NOPRINT

08.2: SQL *Plus commands

Usage of Miscellaneous Commands

Miscellaneous Commands provide a variety of commands that enable you to interact with the user, comment on the code, and enhance coding options.

Miscellaneous Commands

ACCEPT:

- ACCEPT receives input from the terminal and places the contents in variable. This variable can already have been defined with the DEFINE command.
- If the PROMPT option is specified, then the text is displayed after skipping a line.
- You can specify the variable attributes of number or char at this stage. The variable is a char if not otherwise defined.

Miscellaneous Commands

DEFINE variable:

- DEFINE creates a user-defined variable and assigns it to be of char (character) format
- You can assign this variable to be a default value at this stage

08.2: SQL *Plus commands

More on Formatting Commands

Formatting Columns:

- Through the SQL *Plus COLUMN command, you can change the column headings and reformat the column data in your query results.
- Changing Column Headings:
  - When displaying column headings, you can either use the default heading or you can change it by using the COLUMN command
  - The following sections describe how default headings are derived and how to alter them using the COLUMN command. Refer the COLUMN command for more details

08.2: SQL *Plus

Overview

Default Headings:

- SQL *Plus uses column or expression names as default column headings when displaying query results.
- Column names are often short and cryptic. However, expressions can be hard to understand.
- Changing Default Headings:
  - You can define a more useful column heading with the HEADING clause of the COLUMN command, in the following format:
  - COLUMN column_name HEADING column_heading

Using Commands - Examples

Example 1: Changing a Column Heading

- To produce a report from EMP_DETAILS_VIEW with new headings specified for LAST_NAME, SALARY, and COMMISSION_PCT, key in the following commands:

```
COLUMN LAST_NAME HEADING 'LAST NAME'
COLUMN SALARY HEADING 'MONTHLY SALARY'
COLUMN COMMISSION_PCT HEADING COMMISSION
SELECT LAST_NAME, SALARY, COMMISSION_PCT
FROM EMP_DETAILS_VIEW
 WHERE JOB_ID='SA_MAN';
```

Using Commands - Examples

- To change a column heading to two or more words, enclose the new heading in single or double quotation marks when you enter the COLUMN command.

- To display a column heading on more than one line, use a vertical bar (|) where you want to begin a new line.

Note: You can use a character other than a vertical bar by changing the setting of the HEADSEP variable of the SET command.

Using Commands - Examples

Example 2: Splitting a Column Heading

- To give the columns SALARY and LAST_NAME the headings as MONTHLY SALARY and LAST NAME respectively, and to split the new headings onto two lines, key in the following commands:

```
COLUMN SALARY HEADING 'MONTHLY|SALARY'
COLUMN LAST_NAME HEADING 'LAST|NAME'
```

- Now rerun the query with the slash (/) command

Using Commands - Examples

➢ Example 3: Setting the Underline Character

- To change the character used to underline headings to an equal sign and rerun the query, key in the following commands:

```
LAST NAME      MONTHLY SALARY            COMMISSION
=======================================
Russell        14000              .4
Partners       13500              .3
Errazuriz      12000              .3
```

Formatting Number Columns

- When displaying NUMBER columns, you can either accept the SQL *Plus default display width or you can change it by using the COLUMN command.

```
LAST NAME       MONTHLY SALARY      COMMISSION
==========================================
Russell         $14,000             .4
Partners        $13,500             .3
Errazuriz       $12,000             .3
```

```
COLUMN column_name CLEAR or exit from SQL*Plus.
COLUMN SALARY FORMAT $99,990
```

Formatting Datatypes

- When displaying datatypes, you can either accept the SQL*Plus default display width or you can change it using the COLUMN command.
- The format model will stay in effect until you enter a new one, reset the column’s format with the following command:

```
COLUMN column_name CLEAR
```

or exit from SQL*Plus.

Formatting Character Column - Example

- To set the width of the column LAST_NAME to four characters and rerun the current query, key in the following command:

```
COLUMN LAST_NAME FORMAT A4
/
```

```
LAST NAME        MONTHLY SALARY        COMMISSION
==================================================
Russ             $14,00                .3
ell
Part             $13,500               .4
ners
Erra        $12,000          .3
zure
```

Listing & Resetting Column Display Attributes

- To list the current display attributes for a given column, use the COLUMN command followed by the column name only, as shown:

```
COLUMN column_name
```

- To list the current display attributes for all columns, key in the COLUMN command with no column names or clauses after it:

```
COLUMN
```

Listing & Resetting Column Display Attributes

- To reset the display attributes for a column to their default values, use the CLEAR clause of the COLUMN command as shown:

```
COLUMN column_name CLEAR
```

Suppressing & Displaying Column Display Attributes

- You can suppress and restore the display attributes you have given a specific column. To suppress a column's display attributes, key in a COLUMN command in the following form:

```
COLUMN column_name OFF
```

- **OFF** tells SQL *Plus to use the default display attributes for the column. However, it does not remove the attributes you have defined through the COLUMN command

Suppressing & Displaying Column Display Attributes

- To restore the attributes you defined through COLUMN, use the ON clause:

```
COLUMN column_name ON
```

Suppressing Duplicate Values in Break Columns

- The BREAK command suppresses duplicate values by default in the column or expression you name.
- Thus, to suppress the duplicate values in a column specified in an ORDER BY clause, use the BREAK command in its simplest form as follows:

```
BREAK ON break_column
```

Suppressing Duplicate Values in Break Columns

- In this example, to suppress the display of duplicate department numbers in the query results shown, key in the following command:

```
BREAK ON DEPARTMENT_ID;
```

- for the following query (which is the current query stored in the buffer):

```
SELECT DEPARTMENT_ID, LAST_NAME, SALARY FROM
EMP_DETAILS_VIEW WHERE SALARY > 12000 ORDER BY
DEPARTMENT_ID;
```

Inserting space when break column value changes

- BREAK ON DEPARTMENT_ID SKIP 1

```
DEPARTMENT_ID    LAST_NAME      SALARY
------------------------------------------------
20               Hartstein  13000

80               Russel     14000
                 Partner    35000

90               King       12000
                  De Haan   50000
                  Kochhar   40000
```

Summary

- In this lesson, you have learnt:
  - Using different SQL *Plus Reporting commands
  - Generating SQL Reports in different formats

Review Question

- Question 1: ___ command lists the last line in the SQL buffer
- Question 2: ___ command places text at the bottom of each page
- Question 3: SET ___ suppresses the number of query rows returned

Review Question

- Question 4: PAUSE prints the contents of text after skipping a line and then waits for you to press Enter key
  - True / False

- Question 5: SET PAGESIZE controls the width of the output report line
  - True / False