

Creating a Report

References:

SQL Report Generation: http://docs.oracle.com/cd/A87860_01/doc/server.817/a82950/ch4.htm

In SQLPlus, we can design and generate a report by formatting column headings, variable values etc.

Reports are designed in SQLPLUS using a select statement.

1. First, design a select statement which retrieves the desired set of records. This select statement may reference one or more tables, selecting all or some of the attributes from each.

1. Example: To acquire a list of products and where to call for further information

```
SELECT P_PROD_ID, P_NAME, SP_COMPANY, SP_PHONE#
```

```
FROM MROBBERT.OR_PROD, MROBBERT.OR_SUPP
```

```
WHERE P_SUPPLIER_ID = SP_SUPPLIER_ID;
```

1. Carefully check that only those columns and rows desired in the body of the report are selected. Additional information not available from the tables might be required, this will be inserted later as variables.

1. If the records are to appear in alphabetic order, add an ORDER BY clause.
 1. Be sure the attribute you are ordering by appears first in the SELECT clause. If there is a primary attribute on which the order is set, plus secondary order attributes, the order must remain consistent with that in the first SELECT line.
 2. Example: To print the list of products alphabetically by company

```
SELECT SP_COMPANY, SP_PHONE#, P_PROD_ID, P_NAME,
```

```
FROM MROBBERT.OR_PROD, MROBBERT.OR_SUPP
```

```
WHERE P_SUPPLIER_ID = SP_SUPPLIER_ID
```

```
ORDER BY SP_COMPANY;
```

1. The completed SELECT statement will be used as the core of the code needed for report generation.
2. The statements that will generate the report are organized and saved as a Report Generating START FILE, similar to previously constructed Table Generating START FILES.

3. Report Generating START FILE

1. All desired format statements, titles, etc. are listed first, without delimiting punctuation.
2. Only one semicolon is inserted, this after the select statement which appears last.
3. The file name extension of this *report generator file* is SQL.

Following are report enhancements that can be added prior to the SELECT statement, making the report conform to company standards and increasing readability.

Clearing prior settings:

All settings need to be cleared before beginning to set up a report.

- CLEAR SCREEN
- CLEAR BREAKS
- CLEAR COLUMNS

The above statements will clear the settings only, there is no interaction with base table description or content data.

Variables:

System Variables: These are the SET commands that establish the SQL*PLUS environment for the single report session. The syntax is:

SET *system_variable value*

- SET PAGESIZE *n* Where *n* is the number of lines printed per page. The default setting is 14, the size of the monitor screen. This can be enlarge to a printed page, 54, or kept small and multiple *pages* can be printed on a single sheet.
- SET LINESIZE *n* Where *n* is the number of characters per line. 80 characters is the common setting for a printed page. 120 is used for wide green band computer paper.
- SET ECHO Default is OFF. When the file is started, ON lists the command as executed, OFF suppresses the listing.
- SET FEEDBACK Default ON, prints the number of rows returned from a query. Can be set OFF.

- SET HEADING controls printing the headings in the reports. ON, the default, prints column heading in report, OFF eliminates column headings.
- SET NULL by default NULL displays as "", an empty space. This can be changed to print a text string.
- SET PAUSE controls the scrolling of the screen. The default is OFF or NONE. If set ON, one page will display and [RETURN] must be hit in order to retrieve next page. You may specify text that displays at pause by typing phrase in quotes after ON.

SET PAUSE 'Hit return to continue'

SET PAUSE ON

- SET VERIFY with the default, ON, all variables with old and new values are displayed. set verify to OFF to suppress the printing of old values.

SET WRAP controls the word wrap. OFF truncates long values, ON allows data item to wrap to next line. The WRAPPED and TRUNCATED COLUMN clauses will override this setting.

User Variables: A variable can be defined to a value for use in the command file.

DEFINE employee = JONES (no quotes used)

Substitution Variables: Input can be requested from user and entered into commands using PROMPT and ACCEPT. Prompt writes a message to the screen defining clearly what is to be entered, ACCEPT establishes the storage location for the value, name and type.

PROMPT 'Enter a customer number. It must be a 5 digit number.'

ACCEPT CUSTOMER_NUMBER NUMBER PROMPT 'Cust. number: '

The procured value preceded by & can then be used in a SELECT statement.

SELECT CUS_NAME, CUS_ADDR, CUS_CITY

FROM CUSTOMER

WHERE CUST_NUMBER = &CUSTOMER_NUMBER

where &CUSTOMER_NUMBER is the value captured from the screen.

Skip *n*: At the end of each statement the number of lines to be skipped can be specified. The default is no lines skipped.

Breaks and Compute need to be coordinated with the SELECT, GROUP BY and ORDER BY clauses.

Breaks: Breaks can be set to group the returned data, skip lines between attributes or separate the report in to pages. Breaks can be defined on an attribute, expression, alias or the report. For each break, ON xxx is used to define where the break should occur. For example:

```
BREAK ON REPORT ON DEPT
```

would cause a break to occur at the end of the report and for each department change. This allows for computations to be done by department and totals computed for the entire report.

NODUP: Suppresses the repeating of identical column values for a group.

```
BREAK ON CUSTOMER# NODUP SKIP PAGE ON ORDER# SKIP 1
```

In this example a break is set advancing the print to the top of the next page when a new customer is encountered. Customer numbers are not repeated if they are the same as in the prior row. If a customer has more than one order, a line is skipped and the next orders are continued on the page.

Note: The SELECT statement must contain ORDER BY CUSTOMER#, ORDER#, for the report to be correct.

Compute: performs computation on columns or expressions selected from a table. The **break** command must accompany the compute command. With number values, AVG, MAX, MIN, STD, SUM and VAR will work. COUNT will work on any data type.

```
COMPUTE SUM AVG MAX OF P_UNIT_PRICE ON REPORT
```

gives the sum, average and maximum of the unit price for the whole report.

```
COMPUTE SUM OF P_UNITS_ONHAND ON P_PROD_ID REPORT
```

gives the sum of all the units on hand, grand total, plus the subtotals by product id.

Compute works only with the **Break** command. ON coordinates the two commands. **Compute** prints the computed values and restarts the computation when the ON expression's value

changes, or when a specified ROW, PAGE or REPORT break occurs. The syntax of **Compute** differs from **Break** in the ON,

```
COMPUTE SUM OF P_UNITS_ONHAND ON P_PROD_ID REPORT
```

```
BREAK P_PROD_ID ON REPORT
```

Columns: Specify display attributes for each column. The syntax is:

```
COLUMN command_name
```

- **ALIAS** Assigns specified alias to use to refer to column in **BREAK** or **COMPUTE**.

```
COLUMN wage+bonus-taxes-insurance ALIAS net_pay
```

where wage+bonus-taxes-insurance is a column in the **SELECT** statement

- **FORMAT** Specifies the display format. Use text constant such as A10 for character width of 10 or \$9,999 to indicate money. The column heading length is the default width for unspecified columns. If a specification is made less than the column title, the column title is truncated.

```
COLUMN net_pay FORMAT $9,999,999.99
```

- **HEADING** Column names can formatted for output by typing desired name in quotes and using a | to generate line feed in column heading. Define a column heading:

```
COLUMN CUS_NAME HEADING 'Customer | Name'
```

defines a two line heading, | indicating line feed.

- **JUSTIFY** Aligns the heading either Left, Center or Right. The default is numbers justify right and other columns justify left.
- **NEWLINE** Starts a newline before displaying the columns values.
- **NEW_VALUE** *variable* Specifies a variable to hold a column value. If used to display values in the TTITLE, the column must be included in a **BREAK** command with a skip page action.

```
COLUMN job NOPRINT NEW_VALUE jobvar
```

```
BREAK ON job
```

TTITLE CENTER 'Job Report'

LEFT 'Job ' jobvar SKIP 3

the SELECT statement must include the attribute job, however it will not be displayed as a separate column.

- NOPRINT Turns the column printing off for that line. It allows a variable assignment without printing the value. See above example.
- WRAPPED or WORD_WRAPPED or TRUNCATED are used to specify how to handle character data too wide for the column.

COLUMN remark FORMAT A20 WORD_WRAP

the selected column remark is given a width of 20 characters. The contents of the field is printed 20 characters to the line with the break ending after a full word on each line. This will mean some records actual print on more than one line.

DATES:

The date of report generation can be included in the title of each report. This can be achieved within the SELECT statement by adding

COLUMN today NOPRINT NEW_VALUE datevar

...

TTITLE RIGHT datevar

...

above the SELECT, and then

TO_CHAR(SYSDATE,'MM/DD/YY') today,

as one of the attributes in the SELECT clause.

OR

A separate command can be written, prior to the report SELECT, which uses the table DUAL

BREAK ON TODAY

COLUMN TODAY NEW_VALUE MYDATE

SELECT TO_CHAR(SYSDATE,'MM/DD/YY') today FROM DUAL;

CLEAR BREAKS

Title:

TTITLE:

- A multi-line top title can be assigned. This title will repeat at the top of each page.
- Each character string can be aligned, LEFT, RIGHT or CENTER. A TAB *n* inserted into text will skip forward *n* columns.
- A variable can be inserted into the title. This must be a user variable declared prior to use (see above) or a system-maintained variable.
- Format *char* specifies the format for the character string. This is the same as the COL formats explained below.
- SKIP will advance to the next line, while SKIP *n* will advance *n* lines.
- System maintained values can be included:
- SQL.PNO (current page number)
- SQL.USER (current username)
- Example:

```
SQL> TTITLE CENTER 'ANNUAL REPORT' RIGHT 'Page:' FORMAT 999 >SQL.PNO SKIP  
LEFT 'Data represents weekly wages.'
```

ANNUAL REPORT Page: 1

Data Represents weekly wages

- TTITLE OFF suppresses display of the title.
- TTITLE with no clauses reinstates the title previously defined.

BTITLE:

- Sets the bottom title using statements similar to those for