Generating SQL Plus Reports

Report Writing…a science??

- Even though report writing is a small effort it is still a project. Why?
  - Must be thought out
    - Easy to understand
    - Simple to read
    - Provides possible insight

The Steps in Report Writing

- Take these steps to keep you on target..
  - Formulate your query
  - Format the columns
  - Add page headers/footers
  - Format the page
  - Print it
  - If more advanced:
    - Add page and line breaks
    - Add totals and subtotals

Design a Simple Report

- What projects is each employee assigned?
- How many hours have been charged to each project?
- What is the cost of those hours?

Step 1: The Query….type into NOTEPAD

```
--Execute the query to generate the report.
SELECT E.EMPLOYEE_NAME,
       P.PROJECT_NAME,
       SUM(PH.HOURS_LOGGED) ,
       SUM(PH.DOLLARS_CHARGED)
  FROM EMPLOYEE E,
       PROJECT P,
       PROJECT_HOURS PH
 WHERE E.EMPLOYEE_ID = PH.EMPLOYEE_ID
   AND P.PROJECT_ID = PH.PROJECT_ID
 GROUP BY E.EMPLOYEE_ID, E.EMPLOYEE_NAME,
          P.PROJECT_ID, P.PROJECT_NAME;
```

Step 2: Column Formatting

```
--Format the columns
COLUMN employee_name HEADING 'Employee Name' FORMAT A20 WORD_WRAPPED
COLUMN project_name HEADING 'Project Name' FORMAT A20 WORD_WRAPPED
COLUMN hours_logged HEADING 'Hours' FORMAT 9,999
COLUMN dollars_charged HEADING 'Dollars|Charged' FORMAT $999,999.99

--Execute the query to generate the report.
SELECT E.EMPLOYEE_NAME,
       P.PROJECT_NAME,
       SUM(PH.HOURS_LOGGED) hours_logged,
       SUM(PH.DOLLARS_CHARGED) dollars_charged
  FROM EMPLOYEE E,
       PROJECT P,
       PROJECT_HOURS PH
 WHERE E.EMPLOYEE_ID = PH.EMPLOYEE_ID
   AND P.PROJECT_ID = PH.PROJECT_ID
 GROUP BY E.EMPLOYEE_ID, E.EMPLOYEE_NAME,
          P.PROJECT_ID, P.PROJECT_NAME;
```

Use of a Record Separator

- When a line wraps it adds an additional line space. This is a Record Separator placed by SQL Plus
- To turn it off…..

  SQL> SET RECSEP OFF

  SQL> SET RECSEP ON

Step 3: Add Page Headers/Footers

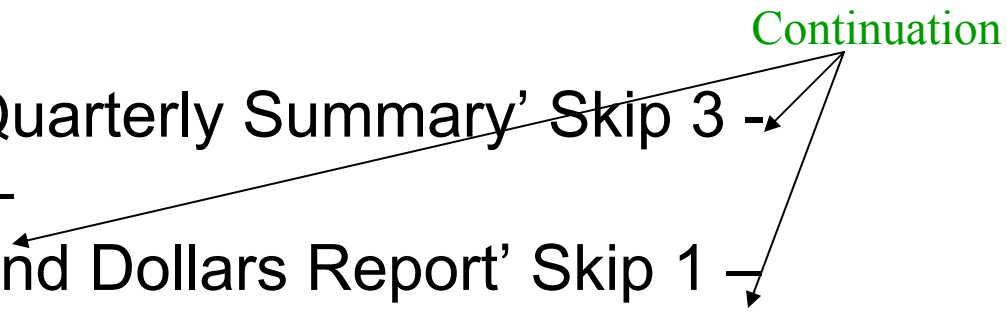
- TTitle
  - The top title of your report
  - Can span several lines, depending on how complex you make it
- BTitle
  - The bottom title of your report

Examples with TTitle

- TTitle Center 'The Quarterly Summary'
  - Displays the title centered
- TTitle Center 'The Quarterly Summary' Skip 3
  - Displays the title centered with 3 lines after it
- TTitle Center 'The Quarterly Summary' Skip 3 -

Left 'I.S. Department' –

Right ' Project Hours and Dollars Report' Skip 1 –



Note: Use of the Skip ensures carriage return, else all on one line

Examples with BTitle

- BTitle LEFT '====================' –
- Skip 1 –
- Right 'Page ' Format 999 SQL.PNO
- Note
  - Format – specifies page number
  - SQL.PNO – supplies current page number

Built Ins for SQL Plus

- SQL.PNO = current page number
- SQL.LNO = current line number
- SQL.Release = current oracle release
- SQL.SQLcode = error code returned by most recent SQL query
- SQL.User = oracle username of the user running the report

```
--Set the linesize, which must match the number of equal signs used
--for the ruling lines in the headers and footers.
SET LINESIZE 61

--Setup page headings and footings
TTITLE CENTER 'Summary Report' SKIP 3 -
      LEFT 'I.S. Department' -
      RIGHT 'Project Hours and Dollars Report' SKIP 1 -
      LEFT '============================================================='

BTITLE LEFT '============================================================' -
      SKIP 1 -
      RIGHT 'Page ' FORMAT 999 SQL.PNO

--Format the columns
COLUMN employee_name HEADING 'Employee Name' FORMAT A20 WORD_WRAPPED
COLUMN project_name HEADING 'Project Name' FORMAT A20 WORD_WRAPPED
COLUMN hours_logged HEADING 'Hours' FORMAT 9,999
COLUMN dollars_charged HEADING 'Dollars|Charged' FORMAT $999,999.99

--Execute the query to generate the report.
SELECT E.EMPLOYEE_NAME,
      P.PROJECT_NAME,
      SUM(PH.HOURS_LOGGED) hours_logged,
      SUM(PH.DOLLARS_CHARGED) dollars_charged
  FROM EMPLOYEE E,
      PROJECT P,
      PROJECT_HOURS PH
 WHERE E.EMPLOYEE_ID = PH.EMPLOYEE_ID
   AND P.PROJECT_ID = PH.PROJECT_ID
 GROUP BY E.EMPLOYEE_ID, E.EMPLOYEE_NAME,
      P.PROJECT_ID, P.PROJECT_NAME;
```

Must fit, 80 columns

Step 4: Format the Page

- Last piece involves controlling the page
- Pagesize
  - Controls # of lines per page
- NewPage
  - Controls size of top margin
- NOTE: Most printers, including laser printers will not allow you to print right up to the top of your page.

Add these lines to the top…

- --Setup pagesize parameters
- SET NEWPAGE 0
- SET PAGESIZE 55

- Can go anywhere (preferred before TTitle) but must go before the SELECT statement

Step 5: Print It!!!

- After reviewing the output on your desktop, prepare to print by SPOOLING
- Place the SPOOL commands immediately BEFORE and AFTER the SQL query

SPOOL C:_Report\proj_hours.lis

Select…..

SPOOL OFF

Some other tips….

- Did you see feedback on the screen
- To turn it off, place this in your script:

  SET FEEDBACK OFF

  SET TERMOUT OFF

Using these settings can potentially improve performance when creating large reports.

Running the Final Product

- After saving your script in NOTEPAD….

  SQL> @c:_reports\summar.sql

Questions??

Part II

Advanced Report Writing

SQL Plus and it's features..

- SQL Plus has a wide assortment of features that we will continue to look at in order to improve our output.

- The first one: BREAK command

BREAK command

- Purpose:
  1. to define page breaks based on changing column values
  2. Controls duplicate values
  3. Can be abbreviated to BRE
  4. SQL Plus only allows one per break setting but allows MULTIPLE “ONs”
  5. Tip: Sort or Group your report by the same columns specified in your BREAK

BREAK attributes

Which column to watch

- BREAK ON {column name}
  skip {lines to skip} Page
  NODUP
  DUP

How many skipped lines
At the break or break the page

Forces dups

Print column
Value only when it changes

PENNSTATE Worthington Scranton
ORACLE

Example Break Format

BREAK ON column_name action ON column_name action –
ON column_name action ON column_name action….

Try this one out. This will suppress repeating values in a report column:

BREAK ON employee_id NODUPLICATES ON employee_name NODUPLICATES
ON project_id NODUPLICATES ON project_name NODUPLICATES

Special Note: NODUPLICATES is the default setting. The above can also be….

BREAK ON employee_id ON employee_name ON project_id ON project_name

Page Lines and Breaks

- For ease of reading, you may want to change a report so that each page starts with a new column.

Use of the SKIP action allows this.

Adding a Page Break

- Showing each's employee's data on a new sheet:

Let's try this…..

```
BREAK ON employee_id SKIP PAGE NODUPLICATES –
      ON employee_name NODUPLICATES –
      ON project_id NODUPLICATES –
      ON project_name NODUPLICATES
```

Adding a Line Break

- The following adds 2 blank lines between projects:

```
BREAK ON employee_id SKIP PAGE NODUPLICATES –
      ON employee_name NODUPLICATES –
      ON project_id SKIP 2 NODUPLICATES –
      ON project_name NODUPLICATES
```

Totals and Subtotals

- SQL Plus takes you a little further to allow for computing of column totals.
- What does COMPUTE do?
  - Defines summary calculations for a reports
  - Tells what columns to summarize and over what range of records
- The COMPUTE in conjunction with the BREAK can be a good mix.

Printing a SubTotal

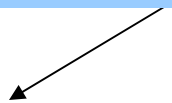
- Apply the following to your report:

Clear Computes
Compute SUM LABEL ‘Totals’ OF hours_logged ON project_id
Compute SUM LABEL ‘Totals’ OF dollars_charged ON project_id
Compute SUM LABEL ‘Totals’ OF hours_logged ON employee_id
Compute SUM LABEL ‘Totals’ OF dollars_charged ON employee_id

Printing a Grand Total

- We use a keyword known as REPORT to control displaying totals for an entire report

Note that the keyword Report is used vs. the column



```
COMPUTE SUM LABEL 'Grand Totals' OF hours_logged ON REPORT
COMPUTE SUM LABEL 'Grand Totals' OF dollars_charged ON REPORT
```

Final Piece

- Add a BREAK command
- Apply this to the report body….

```
BREAK ON REPORT
    ON employee_id SKIP PAGE NODUPLICATES –
    ON employee_name NODUPLICATES –
    ON project_id SKIP 2 NODUPLICATES –
    ON project_name NODUPLICATES
```

RepFooter & RepHeader

- Work like page Headers and Footers
- Only print ONCE
  - Header: After the First Page Title, before detail.
  - Footer: After the last detail line and before the final page footer.

Example of Use

BEFORE REPHEADER…….

```
--Setup page headings and footings
TTITLE CENTER 'The Fictional Company' SKIP 3 -
      LEFT 'I.S. Department' -
      RIGHT 'Project Hours and Dollars Detail' SKIP 1 -
      LEFT
     '==========================================================
     ===========' -
      SKIP 2 'Employee: ' FORMAT 9999 emp_id_var ' ' emp_name_var SKIP 3

BTITLE LEFT
     '==========================================================
     ===========' -
      SKIP 1 -
      LEFT report_date -
      RIGHT 'Page ' FORMAT 999 SQL.PNO
```

With headers…

```
--Setup page headings and footings
TTITLE OFF
REPFOOTER '***End of Hours and Dollars Report ***'
REPHEADER CENTER 'The Fictional Company' SKIP 3 -
      LEFT 'I.S. Department' -
      RIGHT 'Project Hours and Dollars Detail' SKIP 1 -
      LEFT
     '==========================================================
     ==========' -
      SKIP 2 'Employee: ' FORMAT 9999 emp_id_var ' ' emp_name_var SKIP 3

BTITLE LEFT
     '==========================================================
     ==========' -
      SKIP 1 -
      LEFT report_date -
      RIGHT 'Page ' FORMAT 999 SQL.PNO
```