REPORT GENERATION USING iSQL*PLUS COMMANDS

- What is a report?
- Sample report
- Report script
- Break command
- Compute command
- Column command
- Ttitle and Btitle commands
- Clear command
- System variables

What is a report?

A report is information provided in a neat and understandable format. Report contains details as well as summary information. We have so far seen different clauses of SELECT command. But no clause allows you to display details as well as summary information.. For instance, GROUP BY can be used to get summary information but it cannot display details. Simple SELECT can display details but cannot display summary information.

iSQL*PLUS is a tool that is used to send SQL command to Oracle Instance.

It is an environment in which you enter SQL commands. It also provides some extra commands such as DESCRIBE, EDIT etc., which are called as iSQL*PLUS commands.

iSQL*PLUS environment also provides a set of commands which can be used to generate report.

The following is a sample report that we are going to generate in this chapter. This report is generated with iSQL*PLUS commands that are specifically meant for this purpose and SELECT command, which is used to retrieve data from database.

Sample Report

The following is the report to be generated. We will see in the remaining sections of this chapter how to generate this report.

The following are the various iSQL*PLUS commands that are used to generate the following report. The report is used to display the details of payment made by students. The details are divided into groups based on batch and then course. The totals are displayed for each batch, each course and also for the entire report.

Payments Report

COURS	BATCH	ROLLNO	Student Name	PHONE	Date of Joining	Date of Payment	AMOUNT
asp	b2	3	Andy Roberts	433554	11-JAN-01	13-JAN-01	2,000
		3	Andy Roberts	433554	11-JAN-01	20-JAN-01	3,000
		4	Malcom Marshall	653345	16-JAN-01	30-JAN-01	2,000
		4	Malcom Marshall	653345	16-JAN-01	16-JAN-01	3,000
		5	Vivan Richards	641238	16-JAN-01	16-JAN-01	5,000
	*****						-------
	sum						15,000
*****							-------
sum							15,000
c	b3	6	Chirs Evert		14-JAN-01	14-JAN-01	3,500
		7	Ivan Lendal	431212	15-JAN-01	15-JAN-01	3,500
	*****						-------
	sum						7,000
*****							-------
sum							7,000
java	b5	9	Richard Marx	876567	06-APR-01	07-APR-01	3,000
		11	Jody Foster	234344	07-APR-01	10-APR-01	3,500
		11	Jody Foster	234344	07-APR-01	07-APR-01	1,000
		10	Tina Turner	565678	06-APR-01	10-APR-01	4,500
	*****						-------
	sum						12,000

*****							-------
sum							12,000
ora	b1	1	George Micheal	488333	10-JAN-01	10-JAN-01	4,500
		2	Micheal Douglas	334333	11-JAN-01	11-JAN-01	3,500
		2	Micheal Douglas	334333	11-JAN-01	17-JAN-01	1,000
	*****						-------

Payments Report

COURS	BATCH	ROLLNO	Student Name	PHONE	Date of Joining	Date of Payment	AMOUNT
	sum						9,000
ora	b7	12	Louis Figo	535555	12-JUL-01	12-JUL-01	3,000
		13	Marshall Brain	223222	13-JUL-01	13-JUL-01	2,500
		13	Marshall Brain	223222	13-JUL-01	15-JUL-01	1,000
	*****						-------
	sum						6,500
*****							-------
sum							15,500
xml	b4	8	George Micheal	488333	01-MAR-01	01-MAR-01	2,000
		8	George Micheal	488333	01-MAR-01	02-MAR-01	2,000
	*****						-------
	sum						4,000
*****							-------
sum							4,000
							-------
sum							53,500

Report Script

The following is the report script. The script is used to generate the report shown above. The following sections will explain the commands used in the script.

Type the script in any text editor under the name like payreport.sql and then run it from SQL prompt of iSQL*Plus as follows.

payreport.sql

```
rem ************************************************
rem Purpose : Script to generate Payments Report
rem ************************************************
rem set break and compute settings
break on report on course skip page on batch skip 2
compute sum of amount on batch course report
set pagesize 24
set linesize 90
set feedback off
column amount format 99,999
column name format a20 heading 'Student Name'
column dj heading 'Date of|Joining'
column dp heading 'Date of|Payment'
ttitle skip 1 right 'Page:' format 99 sql.pno skip 1 center 'Payments Report' skip 2

select c.ccode course, b.bcode batch, p.rollno, s.name name, phone, dj, dp,
amount
from batches b, students s, payments p, courses c
where b.ccode = c.ccode and b.bcode = s.bcode and s.rollno = p.rollno
order by course, batch;

set feedback on
rem clear settings
clear compute
clear break
clear column
ttitle off
```

BREAK Command

Specifies how which column(s) the data selected by SELECT command is to be grouped (broken). This is also used to specify what should be done when break is given.

BRE[AK] [ON expression [action]] . . .

Expression is the column on which the data is to be grouped or ROW or REPORT keyword.
Action specifies what action should be taken when break is issued. The following are the possible actions.

SKI[P] n	Prints specified number of empty lines.
SKI[P] page	Issues a page break at break.
DUP[LICATE]	Prints the value of a break column in every selected row.
NODUP[LICATE]	Prints blanks instead of the value of a break column when the value is a duplicate of the column's value in the preceding row.

Note: BREAK command alone displays the current break settings
The following BREAK command issues break whenever it detects a change in BCODE column and starts a new page.
break on bcode skip page

Note: It is important to give ORDER BY clause on the columns that are used in BREAK command.
When multiple columns are used in BREAK, iSQL*PLUS issues break starting from rightmost column to leftmost.

break on country skip page on city skip 2
First it issues break on CITY and then issues break on COUNTRY. You need to makes sure the order is right.

BREAK command in script

The script above used the following break.
break on report on course skip page on batch skip 2
The above BREAK issues break on three different levels. First whenever there is a change in BATCH column, second whenever there is a change in COURSE column and finally at the end of the report.
It is important to know the order in which columns are to be given – most specific to most common.
For this break to work properly the ORDER BY clause of the SELECT must be given as follows:
order by course, batch

COMPUTE Command

This is used to compute and print summary information. The FUNCTION is any of the functions listed in Table 1.

COMP[UTE] [function [LABEL text] OF {column}... ON {column | REPORT | ROW} . . .]

If you give more than one function, use spaces to separate the functions.

FUNCTION	COMPUTES	APPLIES TO DATATYPES
AVG	Average of non-null values	NUMBER
COU[NT]	Count of non-null values	all types
MAX[IMUM]	Maximum value	NUMBER, CHAR, VARCHAR, VARCHAR2
MIN[IMUM]	Minimum value	NUMBER, CHAR, VARCHAR, VARCHAR2
NUM[BER]	Count of rows	all types
STD	Standard deviation of non-null values	NUMBER
SUM	Sum of non-null values	NUMBER
VAR[IANCE]	Variance of non-null values	NUMBER

Table 1: List of functions used in COMPUTE.

Note: Every COMPUTE requires a corresponding BREAK. COMPUTE displays summary information, only when there is a break on the given level.

The following COMPUTE will display the subtotal and grand total of amount.
compute sum of amount on bcode report

For the above COMPUTE to function there must be a corresponding BREAK as follows.
break on bcode skip page on report

LABEL keyword specifies the text to be displayed as label for the computed value. The maximum length of the label is 500 characters.
compute sum label 'Grand Total' of amount on report

NOTE: The label is truncated to the size of first column in the SELECT.

COMPUTE command in Script

The following compute command is used in the script to generate the report.
compute sum of amount on batch course report

It will display the sum of AMOUNT column at the end of each batch, course and at the end of report.

COLUMN Command
Specifies the display attributes of a column. The following are the possible options:

- Text for the column heading
- Alignment of the column heading
- Format of NUMBER type columns
- Wrapping of text

COL[UMN**] [{column|expr} [option ...]]**

The following is the list of a few commonly used options of COLUMN command.
FOR[MAT] format
HEA[DING] text
JUS[TIFY] {L[EFT]|C[ENTER]|C[ENTRE]|R[IGHT]}
NUL[L] char
WRA[PPED]|WOR[D_WRAPPED]|TRU[NCATED]

FORMAT

Specifies the display format of the column.
The **format** is a string containing format characters. Format characters depend on the data type of the column.
For **CHAR** type the only available format is :
A*n*
Where *n* is the number of locations used on the screen to display the value. You can specify what action you want to take, when *n* is less than the length of the data. The valid options are:

WRAPPED	Wraps the remaining string to next line.
WORD_WRAPPED	Same as WRAPPED but moves the entire word to next line.
TRUNCATED	Truncates the extra portion of the string.

To format DATE type column we have to use TO_CHAR function.

The following are the available options to format numeric columns.

Element	Example	Description
9	9999	Represents a digit.
0	0999	Displays leading zeroes.
$	$9999	Prefixes a dollar sign to a value.
B	B9999	Displays a zero value as blank.
MI	9999MI	Displays "-" after a negative value.
PR	9999PR	Displays a negative value in angle brackets.
comma	9,999	Displays a comma in the position indicated.
period	99.99	Aligns the decimal point in the position indicated.
V	999V99	Multiplies value by 10n, where n is the number of "9s" after the "V"
EEEE	9.999EEEE	Displays in scientific notation. There should be exactly EEEE.

Table 2: List of Formats for Numeric Data.

The following format will display FEE column in the given format.
column fee format 99,999.00

HEADING
Specifies the heading to be displayed. If no heading is given, the name of the column is displayed as heading.
If heading separator (|) is used then heading is split into multiple lines.
JUSTIFY is used to align heading either to LEFT, CENTER or RIGHT.
column ccode heading 'Course|Code' justify center

NULL
Substitutes null values with the given value in the display. If not given then null is displayed as blank.
COLUMN command may also be used to show display attributes of a particular column as follows:

SQL> column name
COLUMN name ON
HEADING 'Student | Name ' headsep '|'
JUSTIFY center

COLUMN command in script
The following COLUMN commands in the script are used to format columns.

```
column amount  format 99,999
column name    format a20 heading 'Student Name'
column dj      heading 'Date of|Joining'
column dp      heading 'Date of|Payment'
```

AMOUNT column is formatted as 99,999.

NAME column is displayed with a width of 20 columns and heading is set to “Student Name”.
DJ’s heading is set to two lines heading where first line is “Date of” and second line is “Joining”. The same is the case with DP column.

TTITLE and BTITLE Commands

TTITLE specifies the text to be displayed at the top of each printed page. BTITILE displays text at the bottom of the printed page.

TTI[TLE] [options [text]] [ON | OFF]
BTI[TLE] [options [text]] [ON | OFF]

The following is the list of options:

Option	Meaning
BOLD	Prints text in bold print.
COL n	Prints title after n columns in the current line.
ENTER	Aligns title to center.
LEFT	Aligns title to left.
RIGHT	Aligns title to right.
SKIP n	Skips n number of lines before printing title.

TTITLE center 'First line ' skip 1 center 'Second line'
Prints the text First Line in the first line and after skipping one line then displays text Second Line.

TTITLE left ‘Payments Report’ right sql.pno
Prints the text Payments Report on the left and current page number on the right. SQL.PNO returns the current page number. The other variables that you can use are: SQL.USER – current username, SQL.RELEASE – current Oracle release number.

TTITLE off
Will turn off display of top title.

TTITLE command in script
The following TTITLE command is used to display the top title for payments report.

ttitle skip 1 right 'Page:' format 99 sql.pno skip 1 center 'Payments Report' skip 2
First string “Page:” is displayed followed by the page number of the current page. These two are right justified to the page. Then one line is skipped and then at the center of the line title “Payments Report” is displayed. Then two lines are skipped

before the data of the page is displayed.

CLEAR Command
Resets the current value of the specified option.

CLEAR option
Where option may be any of the following.

Option	What it clears?
BRE[AKS]	Break settings set by BREAK.
COMP[UTES]	Compute setting set by COMPUTE.
SCR[EEN]	Content of SQL*PLUS window.
COL[UMNS]	Column setting set by COLUMN.
BUFF[ER]	Text in the current buffer.

The following CLEAR command will rest all summary information specified by COMPUTE.
CLEAR COMPUTES

SET Variables
Set variables/system variables are used to configure iSQL*PLUS environment. For instance, system variables can be used to set page pause, the length of page, size of line and so on.

SET command is used to change the value of system variables.
SET system_variable value

The following is a list of a few system variables. For the complete list of system variables and all the possible options, please see on-line help.
Some of the system variables can store more than one value. For instance, FEEDBACK stores a number and ON/OFF state.

System variable	Meaning
AUTOCOMMIT	If set to ON, automatically commits changes made by SQL command.
FEEDBACK	If set to ON, displays the number of rows selected by SELECT, if number of rows is >= the number specified by FEEDBACK variable.
LINESIZE	Specifies the number of characters that can be displayed in a single line.
PAGESIZE	Specifies the number of lines to be displayed in a single page.
NUMFORMAT	Specifies the default numeric format for numbers.
PAUSE	If set to ON, gives pause at the end of each page. Also contains the text to be displayed when pause is issued.
SERVEROUTPUT	If set to ON, enables the display of output by DBMS_OUTPUT package.
TERMOUT	If set to OFF, suppresses the display of output of start file.

Table 3: SET Variables.

The following SET commands will configure iSQL*Plus for report.

```
set pagesize 50
set linesize 132
set feedback off
set pause off
```

Summary

SQL*PLUS environment, apart from providing SQL commands, has its own set of commands called as SQL*PLUS commands. These commands are basically to generate report and to change the working environment.