Application

Other DBMS

* + PL/SQL is portable
  + You can declare variable
  + You can program with procedural language control structure
  + Can error handle

Benefits of subprograms (modulation)

* + Easy maintained
  + Improved data security and integrity
  + Improve performance
  + Improved code clarity

**PL/SQL Block structure**

DECLARE (optional)

Variable, cursors, user-defined exception

BEGIN (mandatory)

-SQL statement

-PL/SQL statement

EXCEPTION(optional)

Action to perform error occurs

END; (mandatory)

Note – DBMS\_OUTPUT.PUT\_LINE(variable)

We need to instruct to server to show output values

Prompt – SET SERVEROUTPUT ON

Inherit the columns data types

<table\_name>.<col\_name>%TYPE

<table\_name>%ROWTYPE

Block types

Anonymos

Procedure

Functions

SQL cursers

A curser is a private SQL work area

* Implicit curser-to parse and execute your SQL statement
* Explicit curser-declared by programmer

SQL%ROWCOUNT - # of rows affected by most recent SQL statemet

SQL%FOUND – Boolean attribute that evaluate to TRUE if the most recent SQL statement affected one or more rows

SQL%ISOPEN – always evaluate to FLASE because PL/SQL close implicit cursor immediate after they are executing

IF condition THEN

Statement;

[ELSIF condition THEN

Statemet;]

[ELSE

Statement;]

END IF;

Loops

Basic,for,while

Basic loop

LOOP

Statement 1;

…

EXIT [WHEN condition];

END LOOP;

DECLARE

v\_row number;

v\_col number;

BEGIN

v\_row := 0;

v\_col := 0;

LOOP

v\_col := 0;

v\_row := v\_row+1;

LOOP

v\_col := v\_col+1;

DBMS\_OUTPUT.PUT('\*');

EXIT WHEN v\_col = v\_row;

END LOOP;

dbms\_output.new\_line;

EXIT WHEN v\_row = 10;

END LOOP;

END;

While condition

Statement 1;

End while;

FOR counter IN [Reverse]

Lower\_bound … upper\_bound LOOP

Statement1;

Statement1;

END LOOP: