**Problem Sets**

1. What is an anonymous block

Answer:-

Anonymous block is a dummy block of sql commands that has no name associated with it.  the anonymous block is missing the header section altogether. Instead it simply uses the DECLARE reserved word to mark the beginning of its optional declaration section.

anonymous blocks serve as scripts that execute PL/SQL statements, including calls to procedures and functions. Anonymous blocks can also serve as nested blocks inside procedures, functions, and other anonymous blocks

[ DECLARE

... optional declaration statements ... ]

BEGIN

... executable statements ...

[ EXCEPTION

... optional exception handler statements ... ]

END;

2. What is nested sub-program?

Answer:-

A subprogram created inside a PL/SQL block is a nested subprogram. A nested subprogram is stored in the database only if it is nested within a standalone or packaged subprogram.

3. Name the 3 parts of the sub-program with their purposes?

Answer:-

1. Declarative part (optional):-It is an optional part. However, the declarative part for a subprogram does not start with the DECLARE keyword. It contains declarations of types, cursors, constants, variables, exceptions, and nested subprograms. These items are local to the subprogram and cease to exist when the subprogram completes execution.
2. Executable part (required):-This part contains one or more statements that assign values, control execution, and manipulate data.
3. Exception-handling part (optional):-This part contains code that handles runtime errors.

4. Create a procedure inside an anonymous block.

Answer:-

set serveroutput on

declare

v\_emp\_id employees.emp\_id%type;

v\_emp\_name employees.emp\_name%type;

procedure emp (p\_emp\_id in number,p\_emp\_name out varchar)

is

begin

select emp\_name into p\_emp\_name from employees where emp\_id=p\_emp\_id ;

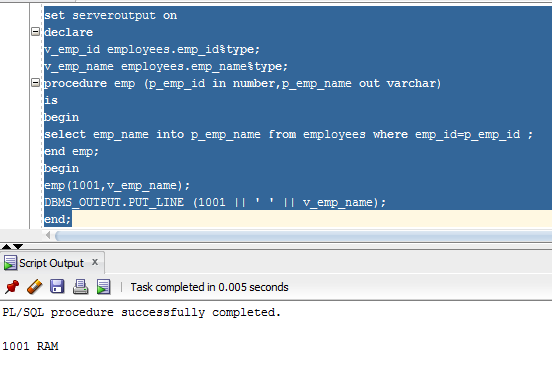
end emp;

begin

emp(1001,v\_emp\_name);

DBMS\_OUTPUT.PUT\_LINE (1001 || ' ' || v\_emp\_name);

end;



5. Create  a function (at the same time) inside an anonymous block.

Answer:-

set serveroutput on

declare

salary number(10);

function get\_salary(f\_emp\_id in number)

return number

is

sal number(10);

begin

select salary into sal

from employees

where f\_emp\_id = emp\_id;

return sal;

end;

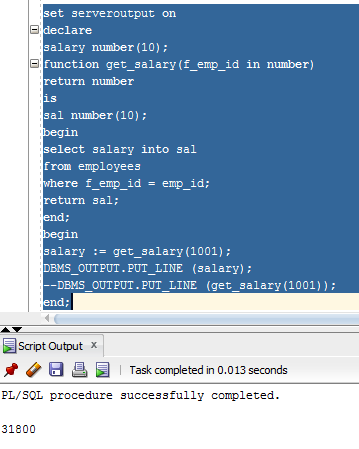
begin

salary := get\_salary(1001);

DBMS\_OUTPUT.PUT\_LINE (salary);

--DBMS\_OUTPUT.PUT\_LINE (get\_salary(1001));

end;



6. Create a nested sub-procedure with forward declaration.

Answer:-

set serveroutput on

declare

v\_emp\_name employees.emp\_name%type;

v\_bus\_no drives.bus\_no%type;

s\_bus\_no number:= 101;

procedure emp1;

procedure emp2 (p\_emp\_id in number,p\_emp\_name out varchar)

is

begin

select emp\_name into p\_emp\_name from employees where emp\_id=p\_emp\_id;

emp1;

end emp2;

procedure emp1

is

begin

select count(bus\_no) into v\_bus\_no from drives where bus\_no = s\_bus\_no;

DBMS\_OUTPUT.PUT\_LINE ('count of bus number 101 = ' || v\_bus\_no);

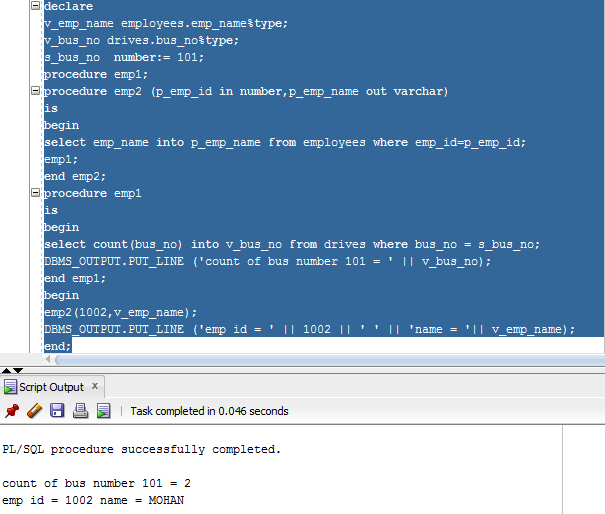
end emp1;

begin

emp2(1002,v\_emp\_name);

DBMS\_OUTPUT.PUT\_LINE ('emp id = ' || 1002 || ' ' || 'name = '|| v\_emp\_name);

end;



7. Create an example of sub-program with default values

Answer:-

set serveroutput on

declare

v\_emp\_id employees.emp\_id%type;

v\_emp\_name employees.emp\_name%type;

procedure emp (p\_emp\_id in number default 1002)

is

begin

select emp\_name into v\_emp\_name from employees where emp\_id=p\_emp\_id ;

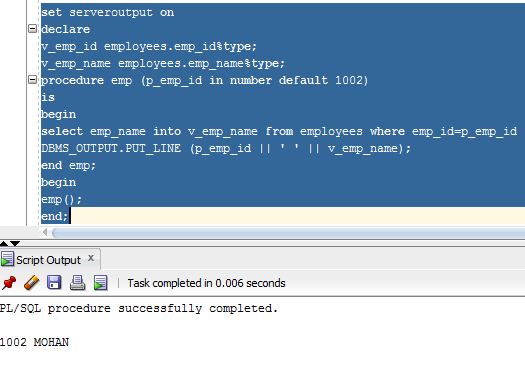
DBMS\_OUTPUT.PUT\_LINE (p\_emp\_id || ' ' || v\_emp\_name);

end emp;

begin

emp();

end;



8. How many types of notations in parameter exists in oracle pl/sql

Answer:-

1. IN

* An IN parameter lets you pass a value to the subprogram.
* It is a read-only parameter. Inside the subprogram, an IN parameter acts like a constant.
* It cannot be assigned a value. You can pass a constant, literal, initialized variable, or expression as an IN parameter.
* You can also initialize it to a default value. It is the default mode of parameter passing.

1. OUT

* An OUT parameter returns a value to the calling program.
* Inside the subprogram, an OUT parameter acts like a variable.
* You can change its value and reference the value after assigning it.
* The actual parameter must be variable and it is passed by value.

1. IN OUT

* An IN OUT parameter passes an initial value to a subprogram and returns an updated value to the caller.
* It can be assigned a value and its value can be read.
* The actual parameter corresponding to an IN OUT formal parameter must be a variable, not a constant or an expression.
* Formal parameter must be assigned a value. Actual parameter is passed by value.

8.1 Question to the students to try on their own  
Now to call the above function from SQL; Find what is dual here, its purpose

Answer:-

The DUAL table is a special one-row, one-column table present by default in Oracle and other database installations.

The DUAL table provide a table for joining in internal views or tables.

9. Explain how overloading a subprogram can be done in pl/sql

Answer:-

PL/SQL allows to overload local subprograms, packaged subprograms, and type methods. We can use the same name for several different subprograms as long as their formal parameters differ in number, order, or data type family.

set serveroutput on

declare

v\_emp\_id employees.emp\_id%type;

v\_emp\_name employees.emp\_name%type;

procedure emp (p\_emp\_id in number default 1002)

is

begin

select emp\_name into v\_emp\_name from employees where emp\_id=p\_emp\_id ;

DBMS\_OUTPUT.PUT\_LINE (p\_emp\_id || ' ' || v\_emp\_name);

end emp;

begin

emp();

end;

--overloading

set serveroutput on

declare

v\_emp\_name employees.emp\_name%type;

v\_sal employees.salary%type;

procedure emp1 (p\_emp\_id in number,p\_emp\_name out varchar)

is

begin

select emp\_name into v\_emp\_name from employees where emp\_id = p\_emp\_id;

DBMS\_OUTPUT.PUT\_LINE ('emp id = ' || p\_emp\_name || ' ' || 'name = '|| v\_emp\_name);

end emp1;

procedure emp1 (p\_emp\_id in number,p\_salary out number)

is

begin

select salary into v\_sal from employees where emp\_id = p\_emp\_id;

DBMS\_OUTPUT.PUT\_LINE ('salary = ' || v\_sal);

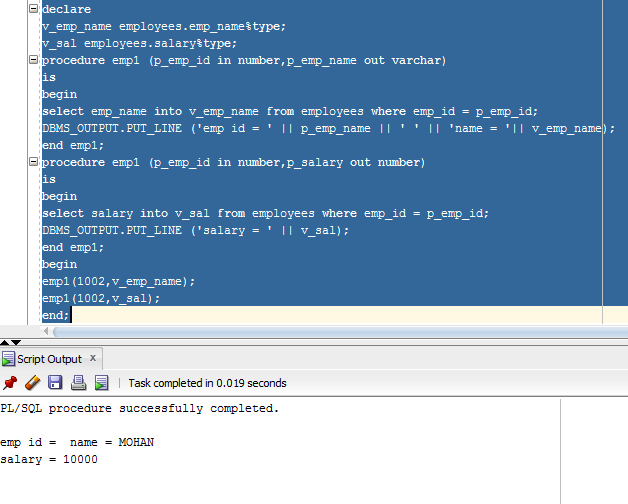
end emp1;

begin

emp1(1002,v\_emp\_name);

emp1(1002,v\_sal);

end;

  
10. What are the restrictions in subprogram overloading

Answer:-

We cannot overload the following programs:-

1. Standalone subprograms
2. Subprograms whose formal parameters differ only in mode .ex:-

PROCEDURE s (p IN VARCHAR2);

PROCEDURE s (p OUT VARCHAR2);

1. Subprograms whose formal parameters differ only in subtype ex:-

PROCEDURE s (p INTEGER);

PROCEDURE s (p REAL);

1. Functions that differ only in return value data type, even if the data types are in different families. ex:-

FUNCTION f (p INTEGER) RETURN BOOLEAN;

FUNCTION f (p INTEGER) RETURN INTEGER;

11. Create a simple package.

Answer:-

CREATE PACKAGE service\_info AS

PROCEDURE service;

END service\_info;

create or replace procedure service

is

s flight%rowtype;

begin

for s in (

select f.fid,f.fname

from flight f,flight\_services fs,service\_master sm

where f.fid = fs.fid and fs.service\_code = sm.service\_code and sm.service\_name = 'NURSE')

loop

DBMS\_OUTPUT.PUT\_LINE (s.fid || ' ' || s.fname);

end loop;

end service;

