M.Sc. (Computer Science)- Second Semester
bject Name- “Practical Based on RDBMS(PL/SQL)
PL/SQL Procedure and Function

By- Prof. Dileep Kumar Sahu

Assistant Professor

Department of Computer Application

Govt. Vishwanath Yadav Tamaskar Post Graduate Autonomous College, Durg (C.G.)

Email ID: dileepksahu20@gmail.com

Contents

- PL/SQL Procedures
- PL/SQL Functions

Objective

- We will learn how to create, compile, and execute a PL/SQ procedure and Function in Database

edure in PL/SQL

SQL procedure is a reusable unit that encapsulates specific business logic
ne application. Technically speaking, a PL/SQL procedure is a
d block stored as a schema object in the Oracle Database.

:

```
 [OR REPLACE ] PROCEDURE procedure_name [ (parameter_list )]

ration statements]

tion statements]
TION
tion handler]
procedure_name ];
```

L Procedure

ove syntax:

cedure _name: specifies the name of the procedure.

REPLACE] option allows modifying an existing procedure.

optional parameter list contains name, mode and types of the meters.

epresents that value will be passed from outside and **OUT** represents this parameter will be used to return a value outside of the procedure

edure in PL/SQL : Header and Body

rocedure begins with a header that specifies its name and an onal parameter list. The procedure contains a header and a body.

der: The header contains the name of the procedure and the meters or variables passed to the procedure.

y: The body contains a declaration section, execution section and eption section similar to a general PL/SQL block.

QL Procedure Header: Passing parameter in procedure

are three ways to pass parameters in procedure:

parameters:

- is Read only.
- n IN parameter can reference inside parameter, but we can change its value.
- is a default parameter in Oracle.

JT parameters:

- is writable
- Ve can set a returned value for the OUT parameter and return it to the calling procedure.
- ote that a procedure ignores the value that you supply for an OUT parameter

OUT parameters:

- is both readable and writable.
- he procedure can read and modify it.
- ote that OR REPLACE option allows you to overwrite the current procedure with the new code.

ɭL Procedure Body

rocedure body has three parts.

executable part is mandatory whereas the declarative and exception-handling are optional.

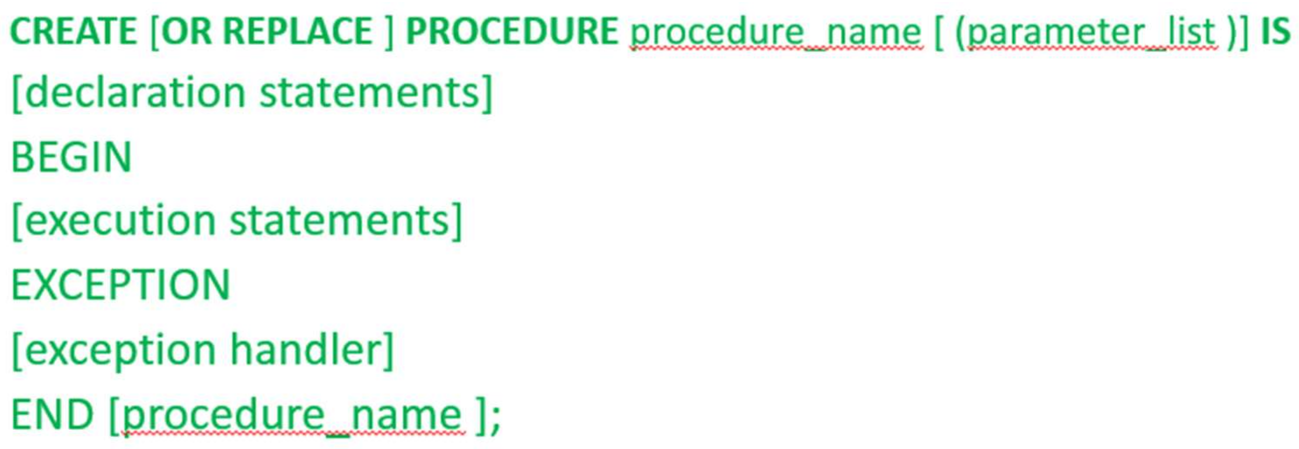
executable part must contain at least one executable statement.

eclarative Part: We can declare variables, constants, cursors, etc

ecutable Part: contains one or more statements that implement specific siness logic.

ception-handling Part: his part contains the code that handles exceptions

:



```
CREATE [OR REPLACE ] PROCEDURE procedure_name [ (parameter_list )] IS
[declaration statements]
BEGIN
[execution statements]
EXCEPTION
[exception handler]
END [procedure_name ];
```

L Procedure: Example

Creation:

***e table** person(id number(10) **primary key**, **name** varchar2(100)); .*

dure Code:

```
or replace procedure "INSERTPERSON" (id IN NUMBER,  name IN VARCHAR2)



into person values(id,name);
```

t:

lure Created

L Procedure: Example

ng a Procedure:

```
N
rtperson(111,'Rohan');
ns_output.put_line('record inserted successfully');
```

ut:

	Name
	Rohan

L Procedure: Drop Procedure

x :

OP PROCEDURE *procedure_name;*

ole of drop procedure

OP PROCEDURE *pro1;*

L Function

QL function is a reusable program unit stored as a schema object in the
le Database.

PL/SQL Function is very similar to PL/SQL Procedure.

main difference between procedure and a function is, a function must alwa
rn a value, and on the other hand a procedure may or may not return a valu

ax:

```
E [OR REPLACE] FUNCTION function_name [parameters]
meter_name [IN | OUT | IN OUT] type [, ...])]
RN return_datatype
S}

nction_body >
function_name];
```

L Function

ove syntax:

- ction_name: specifies the name of the function.
- REPLACE] option allows modifying an existing function.
- optional parameter list contains name, mode and types of the ameters.
- epresents that value will be passed from outside and **OUT** represents this parameter will be used to return a value outside of the procedure
- URN clause specifies that data type you are going to return from the ction.
- ction_body contains the executable part.
- AS keyword is used instead of the IS keyword for creating a standalone ction.

L Function: Example

```
e or replace function
(n1 in number, n2 in number)
n number

mber(8);

n1+n2;
n n3;
```

program to **call the function.**

```
DECLARE
  n3 number(2);
BEGIN
  n3 := adder(11,22);
  dbms_output.put_line('Addition is: ' || r
END;
/
```

Output:

```
Addition is: 33
```

QL Recursive Function

- ogram or a subprogram can call another subprogram.
- en a subprogram calls itself, it is called recursive call and the process is
  wn as recursion.

L Recursive Function: Example

```
LARE
m number;
ctorial number;
NCTION fact(x number)
URN number

umber;
IN
x=0 THEN
:= 1;
SE
:= x * fact(x-1);
D IF;
URN f;
;
```

```
BEGIN
   num:= 6;
   factorial := fact(num);
   dbms_output.put_line(' Factorial '|| num || ' is ' || factorial);
END;
/
```

Output:

```
      Factorial 6 is 720
```

QL Function: example using table

e customer table and have records in it.

e a PL/SQL function:

```
E OR REPLACE FUNCTION totalCustomers
N number IS
number(2) := 0;

CT count(*) into total
1 customer;
URN total;
```

Customer			
Id	Name	Department	Sal
1	Ramesh	web developer	350
2	Sohan	program developer	450
3	Mohan	web designer	350

g PL/SQL Function:

Output: Total no. of Customers: 3

```
CLARE
number(2);
GIN
:= totalCustomers();
oms_output.put_line('Total no. of Customers: ' || c);
D;
```

Thank You