

date
27/09/24

unit - 5

num

PL / SQL

num

- notes

marks

1. Introduction:-

1. The full form of PL/SQL is procedure language structured query language.
2. It is a high level language that combines the features of procedure programming language with SQL.
3. It is an extension of SQL developed by oracle corporation.
4. Procedure is a named block that can take parameters, execute SQL statements and return values.

Structure of PL/SQL:-

It consists of three blocks they are

1. Declaration block.
2. Execution block.
3. Exception block.

Declaration block :-

It is optional section and begins with the keyword "declare". It contains declaring variables, constants that can be used within the block.

Execution Block :-

It is mandatory section and begins with the keyword "begin" and ends with "end" keyword. It contains statements that are run when the block is executed.

Exception Block :-

It is optional section. It begins with "exception" keyword. This block consists actions to perform when error occurs.

Syntax:-

Declare

// Variables and constants.

Begin

// statements are to be executed.

Exception

// statements contains error.

End.

Simple PL/SQL program:-

Set Serveroutput on

Begin

Dbms_output.put_line('hi');

End;

/

In the above example "set serveroutput on" is a command used in SQL that enables the display of output from PL/SQL block. Dbms_output.put_line is the print statement used in PL/SQL.

Data types:-

These are similar to SQL datatypes. int, float, char, varchar,

data, time

Control structures:-

These are also known as decision making statements,

These are used to execute specific blocks of code based on conditions. There are 3 types of control structures. They are-

① If then

② If then else

③ If then else if elif

1. If then :-

It is a condition then statement. It contains only one condition. If the condition is true, the if block will provide results. otherwise there is no output.

Syntax:-

If <condition> then .

End if ,

Ex:-

Set Serverpoint Serveroutput on

Declare

a int ; = 10;

begin

'if a=10 then

dbms-output.put-line(a);

End if;

End;

/

2. If then else :-

The condition in if block is true then the output returns otherwise the else block will return output.

Syntax:-

If ~~Condition~~ then .

Else

End if;

Ex:-

Set Serveroutput on

Declare

a int := 10;

b int := 20;

Begin

If a < b then

dbms-output.put-line ('hi');

Else

dbms-output.put-line ('hello');

End if;

End;

/

3. If then elsif :-

It contains multiple elsif conditions

Syntax:-

If condition 1 then

Elsif condition 2 then

Else

End if;

Ex:-

Set Serveroutput on

Declare

```
a int := 10;  
b int := 20;
```

Begins

If a < b then

```
dbms-output.put-line('hi');
```

Else If a=b then

```
dbms-output.put-line('hello');
```

Else If a>b then

```
dbms-output.put-line('bca');
```

Else

```
dbms-output.put-line('sdhr');
```

End If;

End;

/

Cursors :-

'cursor' is a temporary work area created in system memory. It stores results of queries that is executed by SQL select statements.

they are 2 types of cursors they are

1. Implicit cursor.

2. Explicit cursor.

3. 1- Implicit cursor

These are automatically created by SQL Software internally. When SQL statement is executed in SQL Software the cursor automatically allocates memory.

Explicit Cursors:-

These are the user-defined cursors. We can create our own cursors with any user-defined name. It is used to process query results one row at a time. To create user defined cursors we have to follow four steps.

1. Create the cursor and declare variables in cursor.
2. Open the cursor to allocate memory. (Temporary)
3. Fetch the cursor into cursor variables.
4. Close the cursor to release allocated memory.

Ex:-

```
create table emp (id int, name varchar(5), sal int);
insert into emp values(1, 'raju', 2000);
insert into emp values(2, 'rani', 3000);
select * from emp;
```

Set serveroutput on.

```
declare
    a_id int;
    a_name varchar(5);
cursor a_emp is select id, name from emp;
begin
    open a_emp;
    fetch a_emp into a_id, a_name;
    dbms_output.put_line('id:' || a_id);
    dbms_output.put_line('name:' || a_name);
    close a_emp;
end;
```

Procedure:-

It is a block of code which performs specific tasks. These are reusable code blocks that perform specific actions. They have been defined with a specified name. After creating a procedure, we have to call the procedure using its procedure name.

There are 3 types of parameters used in procedures. They are in, out, inout.

In parameter:-

It is used to send data to the procedure.

Out parameter:-

It returns data from the procedure.

Inout parameter:-

It is a combination of in and out parameter.

Syntax:-

Create or replace procedure procedure name, (parameters)

Inapt · Inout Datatypes);

is / AS

Begin

..... procedure body

end;

/

Ex:-

Procedure without parameters:-

Create or replace procedure (bea.)

is

Begin

dbms_output.put_line('hi');

End;

/

Output:-

Procedure successfully created.

Calling the procedure

begin

bca;

end;

Output:-

Hi;

Ex:- Procedure with parameters (In, Out)

Create or replace procedure bca (a in int, b in int,

c out int)

is

Begin

c=a+b;

end;

/

Calling the procedure:-

declare

c int;

begin

bca (5,3,c);

dbms_output.put_line('The value of c is : ' || c);
end;

/

Output:

The value of c is 8.

Ex:- procedure with parameters (In)

create or replace procedure bca (a inout int);

is

begin

a := out;

end;

,

Calling the procedure

declare

a int;

begin

bca (s)

dbms_output.put_line ('The value of a is : ' || a);

end;

/

Functions:-

1. These are similar to procedures.
2. Functions are used to perform specific tasks.
3. Function contains return type.

Syntax:-

create or replace function function Name (in
parameter datatype)

return datatype

is

Begin

-- function body

end;

Ex:-

create or replace function bca (a in int, b in int)

return int

is

c int;

Begin

c := a+b;

return c;

end;

Calling the function:-

Declare

c int;

begin

c := bca (5,6);

dbms_output.putline (c);

end;

/

Exceptions:-

It is an unexpected error that occurs during the execution of a program. There are two types of exceptions.

1. pre-defined exceptions:-

These are automatically identified oracle server,

they are two types of pre-defined exceptions.

i. NO data found.

ii. exceptions.

* NO data found exception occurs when there is no rows are selected from table.

* zero divide exception occur when a value is divided by zero.

Eg:-

```
Create table emp (id int, name varchar(5));
```

```
insert into emp values (1, 'soni');
```

```
insert into emp values (2, 'ravi');
```

```
Select * from emp;
```

Output:-

ID	name
1	Soni
2	ravi

Set server output on

```
Declare
```

```
a int;
```

```
begin
```

```
Select id into a from emp where id=3;
```

```
exception
```

who no-data-found then

dbms-output.put-line ('there is no data');

End;

Zero divide exception ex:

Set serveroutput on

Declare

a int := 0;

b int := 10;

begin

a := b/a;

Exception

when zero-divide then

dbms-output.put-line ('hi');

End;

2. user defined Exceptions:-

users can also create their own exceptions by using exception name. raise keyword is used to give signal to user defined exception.

Example:-

Set serveroutput on

Declare

a int := 10;

Sony exception;

begin

if a=10 then

```
raise sony;  
end if;  
Exception  
when sony then  
dbms_output.put_line('hi');  
end;
```

1
Packages:-

Packages are a ways to organise and combine procedures, functions into a single unit. It is a collection of related procedures and functions.

⇒ A package contains two parts they are package specification & package body.

Package specification:-

It includes declaration of procedure. it defines what a package can perform procedures & functions.

Package body:-

package body contains the implementation details of the package. It includes the coding of the procedures (or) functions which are declared in the package specification.

Q:- Package specification.

Create or replace package sony is

procedure sdhi(name in varchar);

End sony;

Eg:-

Package body.

Create or replace package body Sony is
procedure Sdhr(name in varchar)
is
begin
dbms_output.put_line(name);
end Sdhr;
End Sony;

Execution of package :-

Set serveroutput on

begin
Sony.Sdhr ('ravi');
End;

Triggers:-

Trigger is a predefined program that automatically executes to specific events occurring in a database.

⇒ A trigger is stored in database. It works automatically when any events occur.

Types of triggers :-

There are three types of triggers.

1. DDL triggers.
2. DML triggers.
3. Database event trigger.

1. DDL trigger :-

These triggers automatically execute when we perform DDL operations (Create, Alter, Drop, Truncate, Remove).

2. DML trigger :-

These triggers automatically execute when we perform DML operations (Insert, Delete, Update).

3. Database event Trigger :-

It executes for actions such as login and logoff to the database.