eBOX

BY AMPHISOFT

Stored Functions

In this session, you will learn:

- How to create a PL/SQL stored functions
- How to declare and use variables
- How to call a stored functions

Introduction to Stored Functions

- A stored function is a special kind stored program that returns a single value.

Why to use Stored Functions?

✓ increase the performance of the applications

✓ to encapsulate common formulas or business rules that are reusable

✓ improves the readability and maintainability of the procedural code

How to create a Stored Functions

The statement CREATE FUNCTION creates a new function.

Syntax

```
CREATE [OR REPLACE] FUNCTION function_name [parameters]
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
RETURN return_datatype
{IS | AS}
BEGIN
   < function_body >
END [function_name];
```

Example for Stored Function

Create a function named ‘noofCustomers’ which takes no parameters. This function must count the number of customers present in the Customer table.

1. Name of the fn – noofCustomers
2. Return datatype -- INT
3. Count of Customers – count() aggregate fn
4. Since the fn returns INT value, we need one INT variable to hold the count and return the count.

Example

```
CREATE OR REPLACE FUNCTION noofCustomers
   RETURN int IS
      total int := 0;
   BEGIN
      SELECT count(*) into total
      FROM Customer;
       RETURN total;
   END;
   /
```

Example for Stored Function

How to call a stored functions?

```
SET SERVEROUTPUT ON
DECLARE
      c int;
   BEGIN
      c := noofCustomers();
      dbms_output.put_line('Total no. of Customers: ' || c);
   END;
   /
```

or

```
Select noofCustomers() from dual;
```

Example for Stored Function

Example

```
SET SERVEROUTPUT ON
DECLARE
    num number;
    factorial number;
FUNCTION fact(x number) RETURN number
IS
    f number;
BEGIN
    IF x=0 THEN
      f := 1;
    ELSE
      f := x * fact(x-1);
    END IF;
RETURN f;
END;
BEGIN
    num:= 6;
    factorial := fact(num);
    dbms_output.put_line(' Factorial '|| num || ' is ' || factorial);
END;
/
```

→

```
Factorial 6 is 720

PL/SQL procedure successfully completed.
```

THANKS

eBOX
BY AMPHISOFT