

分栏模式

注册 登录 会员 搜索 帮助

CNOUG » Oracle开发应用 » Oracle中,Procedure 与 Function 有什么区别？

« 上一主题 | 下一主题 »

➡ 回复

✚ 新帖 ▾

Oracle中,Procedure 与 Function 有什么区别？

打印

HotTea

发表于 2005-11-30 15:12 只看该作者

小 中 大 1楼



甲骨文四段



个人空间

发短消息

加为好友

当前离线

Oracle中,Procedure 与 Function 有什么区别？

Procedure 和 Function 有什么区别啊？

TOP

zlowly

发表于 2005-11-30 21:13 只看该作者

小 中 大 2楼



甲骨文三段



个人空间

发短消息

加为好友

当前离线

一个没有返回值，一个必须返回值.

TOP

yz\_qing

发表于 2005-12-13 14:38 只看该作者

小 中 大 3楼



甲骨文二段



个人空间 发短消息  
加为好友 当前离线

现在存储过程也可以返回值，OUT 就是输出参数，IN就是输入参数，而FUNCTION可以更显示的返回一个值 Return Value。

TOP

niit

发表于 2005-12-13 15:05 [只看该作者](#)

[小](#) [中](#) [大](#) **4楼**



甲骨文六段



工人阶级

个人空间 发短消息  
加为好友 当前离线

FUNCTION只有一个返回值 Return Value

TOP

zcc@weichai.com

发表于 2005-12-13 15:31 [只看该作者](#)

[小](#) [中](#) [大](#) **5楼**

OO-SQL



甲骨文六段



个人空间 发短消息  
加为好友 当前离线

FUNCTION也可以有多个OUT参数!

zcc@weichai.com

发表于 2005-12-13 16:33 只看该作者

小 中 大 6楼

OO-SQL



甲骨文六段



个人空间

发短消息

加为好友

当前离线

有什么不同，测试如下：

```
1 create or replace function f_add(p_1 in out number, p_2 number) return number
2 is
3 begin
4   p_1 := p_1 + p_2;
5   return p_1;
6* end;
SQL> /
```

函数已创建。

```
1 create or replace procedure p_add(p_1 in out number, p_2 number)
2 is
3 begin
4   p_1 := p_1 + p_2;
5* end;
SQL> /
```

过程已创建。

```
var num number
exec :num:=1;
```

1). 函数可以用在表达式中：

```
SQL> exec :num:=1;
```

PL/SQL 过程已成功完成。

```
SQL> print :num;
```

NUM

-----

1

SQL> exec :num := f\_add(:num, 2);

PL/SQL 过程已成功完成。

SQL> print :num;

NUM

-----

3

2),作为表达式使用:

SQL> select f\_add(1,2) from dual;

F\_ADD(1,2)

-----

3

3) 过程通过对参数的负作用返回值:

SQL> exec :num:=1;

PL/SQL 过程已成功完成。

SQL> exec p\_add(:num, 2);

PL/SQL 过程已成功完成。

SQL> print :num

NUM

-----

3

4) , 过程不能用在表达式中:

BEGIN :num := p\_add(:num, 2); END;

\*

ERROR 位于第 1 行:

ORA-06550: 第 1 行, 第 15 列:

PLS-00222: 在此范围中不存在名为 'P\_ADD' 的函数

ORA-06550: 第 1 行, 第 7 列:

PL/SQL: Statement ignored

5) 。也不能做为表达式来用:

```
SQL> select p_add(:num,2) from dual;
```

```
select p_add(:num,2) from dual
```

\*

ERROR 位于第 1 行:

ORA-00904: "🤔\_ADD": 无效的标识符

6) , 函数不能直接EXEC

```
SQL> exec f_add(:num, 2);
```

```
BEGIN f_add(:num, 2); END;
```

\*

ERROR 位于第 1 行:

ORA-06550: 第 1 行, 第 7 列:

PLS-00221: 'F\_ADD' 不是过程或尚未定义

ORA-06550: 第 1 行, 第 7 列:

PL/SQL: Statement ignored

所以说, 最主要的不同还是在使用上。

TOP

zcc@weichai.com

发表于 2005-12-13 16:33 [只看该作者](#)

小 中 大 **7楼**

OO-SQL

有什么不同, 测试如下:

```
1 create or replace function f_add(p_1 in out number, p_2 number) return number
```

```
2 is
```

```
3 begin
```



甲骨文六段



个人空间  
加为好友

发短消息  
当前离线

```
4   p_1 := p_1 + p_2;
5   return p_1;
6* end;
SQL> /
```

函数已创建。

```
1 create or replace procedure p_add(p_1 in out number, p_2 number)
2 is
3 begin
4   p_1 := p_1 + p_2;
5* end;
SQL> /
```

过程已创建。

```
var num number
exec :num:=1;
```

1). 函数可以用在表达式中:  
SQL> exec :num:=1;

PL/SQL 过程已成功完成。

```
SQL> print :num;
```

```
      NUM
-----
      1
```

```
SQL> exec :num := f_add(:num, 2);
```

PL/SQL 过程已成功完成。

```
SQL> print :num;
```

```
NUM
-----
3
```

2),作为表达式使用:

```
SQL> select f_add(1,2) from dual;
```

```
F_ADD(1,2)
-----
3
```

3) 过程通过对参数的负作用返回值:

```
SQL> exec :num:=1;
```

PL/SQL 过程已成功完成。

```
SQL> exec p_add(:num, 2);
```

PL/SQL 过程已成功完成。

```
SQL> print :num
```

```
NUM
-----
3
```

4) , 过程不能用在表达式中:

```
BEGIN :num := p_add(:num, 2); END;
```

\*

ERROR 位于第 1 行:

ORA-06550: 第 1 行, 第 15 列:

PLS-00222: 在此范围中不存在名为 'P\_ADD' 的函数

ORA-06550: 第 1 行, 第 7 列:

PL/SQL: Statement ignored

5) 。也不能做为表达式来用:

```
SQL> select p_add(:num,2) from dual;
```

```
select p_add(:num,2) from dual
```

\*

ERROR 位于第 1 行:

ORA-00904: "🤔\_ADD": 无效的标识符

6) , 函数不能直接EXEC

```
SQL> exec f_add(:num, 2);
```

```
BEGIN f_add(:num, 2); END;
```

\*

ERROR 位于第 1 行:

ORA-06550: 第 1 行, 第 7 列:

PLS-00221: 'F\_ADD' 不是过程或尚未定义

ORA-06550: 第 1 行, 第 7 列:

PL/SQL: Statement ignored

所以说, 最主要的不同还是在使用上。

TOP

« 上一主题 | 下一主题 »

➡ 回复

✚ 新帖 ▾



Powered by **Discuz!** 6.0.0 © 2001-2007 Comsenz Inc.

当前时区 GMT+8, 现在时间是 2010-4-21 20:25 沪ICP备06036178号  
清除 Cookies - 联系我们 - ORACLE 中国用户组 - Archiver - WAP - TOP