

SQL Function 自定义函数

目录

产生背景（已经有了存储过程，为什么还要使用自定义函数）

发展历史

构成

使用方法

适用范围

注意事项

疑问

内容

产生背景（已经有了存储过程，为什么还要使用自定义函数）

与存储过程的区别（存在的意义）：

1. 能够在 `select` 等 SQL 语句中直接使用自定义函数，存储过程不行。
2. 自定义函数可以调用其他函数，也可以调用自己（递归）
3. 可以在表列和 CHECK 约束中使用自定义函数来实现特殊列或约束
4. 自定义函数不能有任何副作用。函数副作用是指对具有函数外作用域（例如数据库表的修改）的资源状态的任何永久性更改。函数中的语句唯一能做的更改是对函数上的局部对象（如局部游标或局部变量）的更改。不能在函数中执行的操作包括：对数据库表的修改，对不在函数上的局部游标进行操作，发送电子邮件，尝试修改 目录，以及生成返回至用户的结果集。存储过程没有此限制
5. 函数只能返回一个变量。而存储过程可以返回多个

发展历史

SqlServer 2000 之后都支持用户自定义函数

构成

在 SQL Server 2000 中根据函数返回值形式的不同将用户自定义函数分为三种类型：标量函数（Scalar Function）、内嵌表值函数（Inline Function）、多声明表值函数（Multi-Statement Function）

标量函数：标量函数是对单一值操作，返回单一值。能够使用表达式的地方，就可以使用标量函数。像我们经常使用的 `left`、`getdate` 等，都属于标量函数。系统函数中的标量函数包括：数学函数、日期和时间函数、字符串函数、数据类型转换函数等

内嵌表值函数：内嵌表值函数的功能相当于一个参数化的视图。它返回的是一个表，内联表值型函数没有由 `BEGIN-END` 语句括起来的函数体。其返回的表由一个位于 `RETURN` 子句中的 `SELECT` 命令段从数据库中筛选出来。

作用

多声明表值函数：可以看作标量型和内嵌表值型函数的结合体。它的返回值是一个表，但它和标量型函数一样有一个用 `BEGIN-END` 语句括起来的函数体，返回值的表中的数据是由函数体中的语句插入的。由此可见，它可以进行多次查询，对数据进行多次筛选与合并，弥补了内联表值型函数的不足。

使用方法

SQL Server 为三种类型的用户自定义函数 提供了不同的命令创建格式。

- (1) 创建标量型用户自定义函数（Scalar functions） 其语法如下：

各参数说明如下：

owner_name：指定用户自定义函数的所有者。

function_name：指定用户自定义函数的名称。

database_name.owner_name.function_name 应是惟一的。

@parameter_name：定义一个或多个参数的名称。一个函数最多可以定义 1024 个参数每个参数前用“@”符号标明。参数的作用范围是整个函数。参数只能替代常量，不能替代表名、列名或其它数据库对象的名称。用户自定义函数不支持输出参数。

scalar_parameter_data_type：指定标量型参数的数据类型，可以为除 TEXT、NTEXT、IMAGE、CURSOR、TIMESTAMP 和 TABLE 类型外的其它数据类型。

scalar_return_data_type：指定标量型返回值的数据类型，可以为除 TEXT、NTEXT、IMAGE、CURSOR、TIMESTAMP 和 TABLE 类型外的其它数据类型。

scalar_expression：指定标量型用户自定义函数返回的标量值表达式。

function_body：指定一系列的 Transact-SQL 语句，它们决定了函数的返回值。

ENCRYPTION：加密选项。让 SQL Server 对系统表中有关 CREATE FUNCTION 的声明加密，以防止用户自定义函数作为 SQL Server 复制的一部分被发布（Publish）。

SCHEMABINDING：计划绑定选项将用户自定义函数绑定到它所引用的数据库对象如果指定了此选项，则函数所涉及的数据库对象从此将不能被删除或修改，除非函数被删除或去掉此选项。应注意的是，要绑定的数据库对象必须与函数在同一数据库中。

例 13-19: 创建工龄工资计算函数。

```
use pangu
go
create function WorkYearWage(@hiredate datetime, --hiredate 表示雇佣日期
    @today datetime, @per_wage money)
    --today 表示当前的日期, per_wage 表示每一年工龄应得的工资额
returns money
as
begin
    declare @WorkYearWage money
    set @WorkYearWage = (year(@today)-year(@hiredate))*@per_wage
    return(@WorkYearWage)
end --结束函数定义
--创建函数
go
--调用函数
select pangu.dbo.workyearwage('1991-7-1',getdate(),15)
    as work_year_wage
运行结果如下:
work_year_wage
```

```
135.0000
(1 row(s) affected)
```

上例的创建语句也可以写成如下形式

```
create function WorkYearWage(@hiredate datetime,
    @today datetime, @per_wage money)
returns money
as
begin
    return((year(@today)-year(@hiredate))*@per_wage)
end
```

(2) 创建内联表值型用户自定义函数 (Inline Table-valued Functions)
其语法如下:

```
CREATE FUNCTION [ owner_name. ] function_name
    ( [ { @parameter_name scalar_parameter_data_type [ = default ] } [ ...n ] ] )
RETURNS TABLE
[ WITH < function_option > [ ,...n ] ]
[ AS ]
RETURN [ ( ) select-stmt [ ] ]
```

各参数说明如下:

TABLE: 指定返回值为一个表。

select-stmt: 单个 **SELECT** 语句，确定返回的表的数据。

其余参数与标量型用户自定义函数相同。

例 13-20：创建返回所有订购某类产品的公司信息函数。

```
use pangu
go
create function orderfirms(@productid varchar(30))
returns table
as
return (select f.firm_id, f_name, f_intro
        from firms f, orders o
        where f.firm_id = o.firm_id
        and o.order_id in (select o.order_id
                           from products p, orders o
                           where p.p_id = @productid
                           and p.p_id = o.p_id))
```

```
go
--调用函数
select *
from orderfirms(10030001)
```

运行结果如下：

| firm_id | f_name | f_intro |
|----------|--------|------------|
| 10030001 | 010101 | 从事数字化工作的公司 |
| 10040001 | 神通 | 从事通信业务的公司 |

(2 row(s) affected)

(3) 创建多声明表值型用户自定义函数

其语法如下：

```
CREATE FUNCTION [ owner_name. ] function_name
  ( [ { @parameter_name scalar_parameter_data_type [ = default ] } [ ,...n ] ] )
RETURNS @return_variable TABLE < table_type_definition >
[ WITH < function_option > [ ,...n ] ]
[ AS ]
BEGIN
  function_body
  RETURN
END

< table_type_definition > ::=
  ( { column_definition | table_constraint } [ ,...n ] )
```

各参数说明如下：

@return_variable：一个 TABLE 类型的变量，用于存储和累积返回的表中的数据行。其余参数与标量型用户自定义函数相同。

在多声明表值型用户自定义函数的函数体中允许使用下列 Transact-SQL 语句。赋值语句（Assignment statements）；流程控制语句（Control-of-Flow statements）；定义作用范围在函数内的变量和游标的 DECLARE 语句；SELECT 语句；编辑函数中定义的表变量的 INSERT、UPDATE 和 DELETE 语句；在函数中允许涉及诸如声明游标、打开游标、关闭游标、释放游标这样的游标操作，对于读取游标而言，除非在 FETCH 语句中使用 INTO 从句来对某一变量赋值，否则不允许在函数中使用 FETCH 语句来向客户端返回数据。此外不确定性函数（Non-deterministic functions）不能在用户自定义函数中使用。所谓不确定性函数是指那些使用相同的调用参数在不同时刻调用得到的返回值不同的函数。这些函数如表 13-3 所示（全局变量也可以视为一种函数）。

表 13-3 不确定性函数

| | | |
|----------------|-------------------|-----------------|
| @@CONNECTIONS | @@CPU_BUSY | @@IDLE |
| @@IO_BUSY | @@MAX_CONNECTIONS | @@PACK_RECEIVED |
| @@PACK_SENT | @@PACKET_ERRORS | @@TIMETICKS |
| @@TOTAL_ERRORS | @@TOTAL_READ | @@TOTAL_WRITE |
| GETDATE | GETUTCDATE | NEWID |
| RAND | TEXTPTR | |

例 13-21：创建函数。

```
use pangu
go
create function chiefinfo(@inputdeptid char(8))
returns @chiefinfo table
    (d_name varchar(50),
     emp_id char(8) primary key,
     e_name varchar(20),
     wage money)
as
begin
    declare @deptname varchar(50)
    declare @e_wage money
    declare @empid char(8)
    declare @chief table
        (d_name varchar(50),
         emp_id char(8) primary key,
         e_name varchar(20),
         wage money default 0)
    select @deptname = d_name
    from department
```

```
where dept_id = @inputdeptid
insert @chief
select d_name, emp_id, e_name, e_wage
from chief_info
where chief_info.d_name = @deptname
```

```
select @empid = emp_id
from chief_info
where chief_info.d_name = @deptname
select @e_wage = e_wage
from employee
where emp_id = @empid
update @chief
set wage = e_wage
where d_name = @deptname
insert @chiefinfo
select d_name, emp_id, e_name, wage
from @chief
return
end
go
select *
from chiefinfo(1001)
```

运行结果如下：

(1 row(s) affected)

Type added.

适用范围

1. 只查询，不修改数据库的状态（修改、删除表中记录等）
2. 结果集需要通过递归等方法得到时，可以使用函数，函数比较灵活
3. 结果集需要直接被引用时，可以使用函数。需要对结果集进行再加工（指放在 select 语句中等），可以使用函数，函数可以嵌在 select 等 sql 语句中。

注意事项：

用户自定义函数不能用于执行一系列改变数据库状态的操作

在编写自定义函数时需要注意的：

对于标量函数：

1. 所有的入参前都必须加@
2. create 后的返回，单词是 returns，而不是 return
3. returns 后面的跟的不是变量，而是返回值的类型，如：int, char 等。
4. 在 begin/end 语句块中，是 return。

内嵌表值函数:

1. 只能返回 table, 所以 returns 后面一定是 TABLE
2. AS 后没有 begin/end, 只有一个 return 语句来返回特定的记录。

多语句表值函数:

1. returns 后面直接定义返回的表类型, 首先是定义表名, 表明前面要加@, 然后是关键字 TABLE, 最后是表的结构。
2. 在 begin/end 语句块中, 直接将需要返回的结果 insert 到 returns 定义的表中就可以了, 在最后 return 时, 会将结果返回。
3. 最后只需要 return, return 后面不跟任何变量。

疑问: 自定义函数不能修改数据库, 但可以调用存储过程, 那么在自定义函数中调用一个有修改数据库的操作的存储过程, 这个自定义函数能不能执行?

答: 自定义函数只能调用扩展存储过程, 但是 SQL Server 2008 的后续版本将删除该功能, 不再支持扩展存储过程, 所以应避免在开发中使用扩展存储过程。因此, 可以得出结论是: 实际开发中, 函数不会去调用存储过程, 也就无法对数据库进行修改操作了。

参考:

<http://technet.microsoft.com/zh-tw/library/ms186755.aspx>

<http://msdn.microsoft.com/zh-cn/library/ms175200.aspx>

<http://www.cnblogs.com/Athrun/archive/2007/07/27/833416.html>