SQL Server中表变量的使用[C#] (2011-10-09 12:52:13)

标签： sql server 表变量 it 分类： SQLServer

```
—声明表变量
declare @t_tableBarcode    table(id int identity(1,1),BarcodeID int,Barcode varchar(50))
declare @t_iId        int
—表变量中插入值
insert into @t_tableBarcode(BarcodeID,Barcode) select BarcodeID,Barcode from tTable where ...
select @t_iId=min(id) from @t_tableBarcode
—对表变量进行循环
while @t_iId is not null
begin
  select @t_iBarcodeID=BarcodeID,@p_cBarcode =Barcode from @t_tableBarcode where id = @t_iId
 //具体应用实现
    ......
end
  select @t_iId=min(id) from @t_tableBarcode where id>@t_iId
end
```

SQL SERVER临时表的使用&表变量的使用

临时表说明：
创建临时表可以创建本地和全局临时表。本地临时表仅在当前会话中可见；全局临时表在所有会话中都可见。

本地临时表的名称前面有一个编号符 (#table_name)，而全局临时表的名称前面有两个编号符 (##table_name)。

SQL 语句使用 CREATE TABLE 语句中为 table_name 指定的名称引用临时表：

CREATE TABLE #TableName (id INT PRIMARY KEY)

INSERT INTO #TableName VALUES (1)

--临时表

begin

create table #table1(t1 int,t2 varchar(100))

```
insert into #table1 (t1) values(33);

select * from #table1;

drop table #table1

end
```

当创建本地或全局临时表时，CREATE TABLE 语法支持除 FOREIGN KEY 约束以外的其它所有约束定义。如果在临时表中指定 FOREIGN KEY 约束，该语句将返回警告信息，指出此约束已被忽略，表仍会创建，但不具有 FOREIGN KEY 约束。在 FOREIGN KEY 约束中不能引用临时表。考虑使用表变量而不使用临时表。当需要在临时表上显式地创建索引时，或多个存储过程或函数需要使用表值时，临时表很有用。通常，表变量提供更有效的查询处理。

全局临时表在创建此表的会话结束且其它任务停止对其引用时自动除去。任务与表之间的关联只在单个 Transact-SQL 语句的生存周期内保持。换言之，当创建全局临时表的会话结束时，最后一条引用此表的 Transact-SQL 语句完成后，将自动除去此表。 在存储过程或触发器中创建的本地临时表与在调用存储过程或触发器之前创建的同名临时表不同。如果查询引用临时表，而同时有两个同名的临时表，则不定义针对哪个表解析该查询。嵌套存储过程同样可以创建与调用它的存储过程所创建的临时表同名的临时表。嵌套存储过程中对表名的所有引用都被解释为是针对该嵌套过程所创建的表

表和表变量有什么区别?

表是实体，数据保存在数据库文件内。

表变量是一种特殊变量，只对当前会话有效。

什么时候用表好,什么时候用表变量好?

table 变量的行为类似于局部变量，有明确定义的作用域。该作用域为声明该变量的函数、存储过程或批处理。

在其作用域内，table 变量可像常规表那样使用。该变量可应用于 SELECT、INSERT、UPDATE 和 DELETE 语句中用到表或表的表达式的地方。但是，table 不能用在下列语句中：

INSERT INTO table_variable EXEC 存储过程。

SELECT select_list INTO table_variable 语句。

在定义 table 变量的函数、存储过程或批处理结束时，自动清除 table 变量。

·表类型声明中的 CHECK 约束、DEFAULT 值和计算列不能调用用户定义函数。

·在存储过程中使用 table 变量与使用临时表相比，减少了存储过程的重新编译量。

·涉及表变量的事务只在表变量更新期间存在。这样就减少了表变量对锁定和记录资源的需求。

·不支持在表变量之间进行赋值操作。

创建表变量：

declare @table1 table(t1 int)declare @table2table(t2 int)set @table1=@table2 --错误 不支持在表变量之间进行赋值

例子：

--表变量

begin

declare @table1 table(t1 int,t2 varchar(100));

insert into @table1 (t1) values(1);

select * from @table1;

end