**[数据库中游标的使用](http://www.cnblogs.com/zijinguang/archive/2007/12/17/1003281.html)**

1. 为何使用游标：

         使用游标(cursor)的一个主要的原因就是把集合操作转换成单个记录处理方式。用SQL语言从数据库中检索数据后，结果放在内存的一块区域中，且结果往往是一个含有多个记录的集合。游标机制允许用户在SQL server内逐行地访问这些记录，按照用户自己的意愿来显示和处理这些记录。

2. 如何使用游标：

     一般地，使用游标都遵循下列的常规步骤：

      (1)  声明游标。把游标与T-SQL语句的结果集联系起来。  
      (2)  打开游标。  
      (3)  使用游标操作数据。  
      (4)  关闭游标。

2.1. 声明游标

DECLARE CURSOR语句SQL-92标准语法格式：

DECLARE 游标名 [ INSENSITIVE ] [ SCROLL ] CURSOR

FOR  sql-statement

Eg:

Declare MycrsrVar  Cursor

FOR Select \*  FROM tbMyData

2.2  打开游标

OPEN MycrsrVar

当游标被打开时，行指针将指向该游标集第1行之前，如果要读取游标集中的第1行数据，必须移动行指针使其指向第1行。就本例而言，可以使用下列操作读取第1行数据：

     FETCH FIRST from E1cursor

     或 FETCH NEXT from E1cursor

2.3      使用游标操作数据

下面的示例用@@FETCH\_STATUS控制在一个WHILE循环中的游标活动

/\* 使用游标读取数据的操作如下。\*/

DECLARE E1cursor cursor      /\* 声明游标，默认为FORWARD\_ONLY游标 \*/

FOR SELECT \* FROM c\_example

OPEN E1cursor                /\* 打开游标 \*/

FETCH NEXT from E1cursor     /\* 读取第1行数据\*/

WHILE @@FETCH\_STATUS = 0     /\* 用WHILE循环控制游标活动 \*/

BEGIN

          FETCH NEXT from E1cursor   /\* 在循环体内将读取其余行数据 \*/

END

CLOSE E1cursor               /\* 关闭游标 \*/

DEALLOCATE E1cursor          /\* 删除游标 \*/

2.4     关闭游标

     使用CLOSE语句关闭游标

CLOSE { { [ GLOBAL ] 游标名 } | 游标变量名 }

使用DEALLOCATE语句删除游标，其语法格式如下：

DEALLOCATE { { [ GLOBAL ] 游标名 } | @游标变量名

3.  FETCH操作的简明语法如下：

FETCH

           [ NEXT | PRIOR | FIRST | LAST]

FROM

{ 游标名  | @游标变量名 } [ INTO @变量名 [，…] ]

参数说明：

NEXT   取下一行的数据，并把下一行作为当前行(递增)。由于打开游标后，行指针是指向该游标第1行之前，所以第一次执行FETCH NEXT操作将取得游标集中的第1行数据。NEXT为默认的游标提取选项。

INTO @变量名[,…]  把提取操作的列数据放到局部变量中。列表中的各个变量从左到右与游标结果集中的相应列相关联。各变量的数据类型必须与相应的结果列的数据类型匹配或是结果列数据类型所支持的隐性转换。变量的数目必须与游标选择列表中的列的数目一致。

--------------------------------------------------------------------------------------------------------------------------------

每执行一个FETCH操作之后，通常都要查看一下全局变量@@FETCH\_STATUS中的状态值，以此判断FETCH操作是否成功。该变量有三种状态值：

・  0  表示成功执行FETCH语句。

・ -1  表示FETCH语句失败，例如移动行指针使其超出了结果集。

・ -2  表示被提取的行不存在。

由于@@FETCH\_STATU是全局变量，在一个连接上的所有游标都可能影响该变量的值。因此，在执行一条FETCH语句后，必须在对另一游标执行另一FETCH 语句之前测试该变量的值才能作出正确的判断。