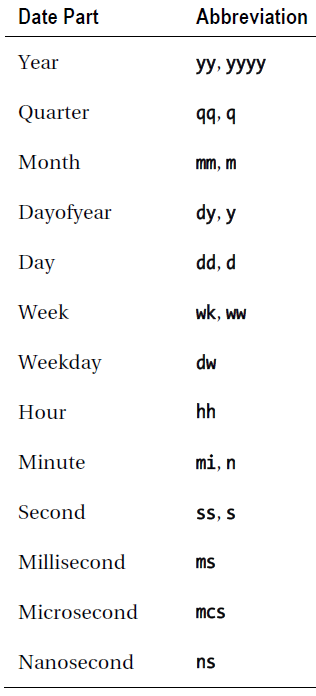
1. /\*\*/注释多行， --注释单行。
2. 聚集索引。表数据是按聚集索引的顺序来存储的，一张表只能有一个聚集索引。非聚集索引是单独的结构，与表数据存储顺序无关。
3. GO不加分行，把代码分为不同的批次。
4. 在不符合命名规则的表名、列名加[]来使用它们.
5. WHERE子句的计算结果有true、false、unknown. 仅返回结果为true的行。
6. SELECT <column1>,<column2> FROM <schema>.<table> WHERE <column> [NOT] BETWEEN <value1> AND <value2>; [value1,value2]
7. SELECT <column1>,<column2> FROM <schema>.<table> WHERE <column> [NOT] LIKE 'value[a-c]' 'value[a,b,c]' value[^d]; %代表任意多个字符。\_代表一个字符。[]限制可选的一个字符.^限制不想要的一个字符
8. WHERE子句中AND优先级高于OR.
9. SELECT <column1>,<column2> FROM <schema>.<table> WHERE <column> [NOT] IN (<value1>,<value2>);
10. SELECT <column1>,<column2> FROM <schema>.<table> WHERE <column> IS [NOT] NULL;
11. SELECT <column1>,<column2> FROM <schema>.<table> ORDER BY <column1>[<sort direction>],<column2> [<sort direction>];
12. The concatenation operator (+) allows you to add together two strings. The syntax is simple: <string or column name> + <string or column name>.当+连接一个数字和一个字符串时，会将字符串转换为数字，如果字符串不能转换为数字，将出错。
13. ISNULL(value, replacement)如果value是null，则返回replacement。COALESCE(value1,value2,value3,……)返回第一个非null值。COALESCE是ANSI标准。
14. CAST和CONVERT进行类型转换。CAST(<value> AS <new data type>) CONVERT(<new data type>,<value>). CAST是ANSI标准。
15. 当两个不同优先级的数据进行操作时，结果类型为优先级高的那一个。
16. LEFT(string, <number of character>) RIGHT(string, <number of character>) get the specify number of character from the left or right side of string.
17. LEN(<string>)字符串中的字符数，DATALENGTH(<STRING>)返回字符串中的字节数。
18. CHARINDEX搜索字符串在另一个字符串中的位置。CHARINDEX(<search string>,<target string>[,<start location>])
19. SUBSTRING(<string>,<start location>,<length>)
20. REVERSE(<string>)
21. UPPER(<string>) LOWER(<string>)
22. REPLACE(<string>,<string to replace>,<replace>)返回当前系统时间
23. -----------------------------------date function-----------------------------------------------------
24. GETDATE() SYSDATETIME()
25. DATEADD(<date part>, <number>,<date>) ex: SELECT DATEADD(month,-1,'1/29/2009');
26. CONVERT(<data type, usually varchar>, <date>, <style>) used to format a date.

CONVERT(VARCHAR,OrderDate,1) AS "1",

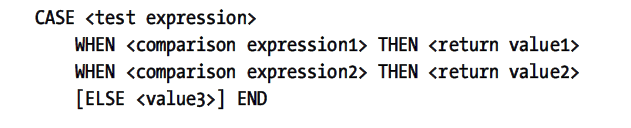
CONVERT(VARCHAR,OrderDate,101) AS "101",

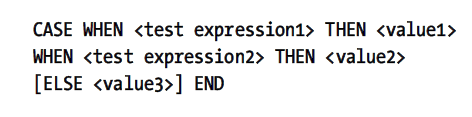
CONVERT(VARCHAR,OrderDate,2) AS "2",

CONVERT(VARCHAR,OrderDate,102) AS "102"



1. DATEDIFF(<date part>,<early date>,<later date>)
2. DATENAME(<date part>,<date>) DATEPART(<date part>,<date>)返回date中指定的部分。
3. DAY(<date>) MONTH(<date>) YEAR(<date>)返回day month year 部分。
4. -------------------------------------数学函数---------------------------------------------------------
5. ABS(<number>) POWER(<number>,<power>) SQUARE(<number>) SQRT(<number>)
6. ROUND(<number>,<length>[,<function>])
7. RAND() returns a float value between 0 and 1.
8. CASE WHEN 计算第一个表达式为真的值。



1. 
2. SELECT DB\_NAME(),HOST\_NAME(),CURRENT\_USER,USER\_NAME(),APP\_NAME()
3. INNER JOIN:当Joining列匹配的时候，包含在查询结果中。

SLELECT <select list> FROM <table1> [INNER] JOIN <table2> ON <table1>.<col1> = <table2>.<col2>

SLELECT <select list> FROM <table1> [INNER] JOIN <table2> ON <table1>.<col1> = <table2>.<col2> AND <table1>.<col3> = <table2>.<col4>

SLELECT <select list> FROM <table1> [INNER] JOIN <table2> ON <table1>.<col1> = <table2>.<col2> [INNER] JOIN <table3> ON <table2>.<col2> = <table3>.<col3>

1. OUTER JOIN获取JOIN表中一个的所有行，即使某些行没有对应的JOIN COLUMN在另一个表中。

SELECT <select list> FROM <table1> LEFT|RIGHT [OUTER] JOIN <table2> ON <table1>.<col1> = <table2>.<col2>;

1. Using OUTER JOIN to find rows with no match:

SELECT <select list> FROM <table1> LEFT [OUTER] JOIN <table2> ON <table1>.<col1> = <table2>.<col2> WHERE <col2> IS NULL;

1. SELECT <select list> FROM <table1> FULL [OUTER] JOIN <table2> ON <table1>.<col1> = <table2>.<col2>;没有相同col的项都会出现在结果中。

SELECT <SELECT list> FROM <table1> CROSS JOIN <table2>; every row from one table matched to every row from another table.

1. Self join: SELECT <select list> FROM <table1> AS a LEFT [OUTER] JOIN <table1> AS b ON a.<col1> = b.<col2>;使用left只是说明，可以用right或者inner join
2. 子查询：SELECT <select list> FROM <table1> WHERE <col1> [not] IN (SELECT <col2> FROM <table2>);
3. UNION: SELECT <col1>,<col2>,<col3> FROM <table1> UNION [ALL] SELECT <col4>,<col5>,<col6> FROM <table2>; UNION是去重的，UNION ALL是不去重的。必须确保union的列数目和类型是相同的。
4. SELECT <select list> FROM <table1> [INNER] JOIN (SELECT <select list> FROM <table2>) AS B ON <table1>.<col1> = B.<col2>
5. COMMON TABLE EXPRESSION:

WITH <CTE Name> AS (SELECT <select list> FROM <table1>) SELECT <select list> FROM <table2> [INNER] JOIN <CTE Name> ON <table2>.<col1> = <CTE Name>.<col2>;可以定义多个with

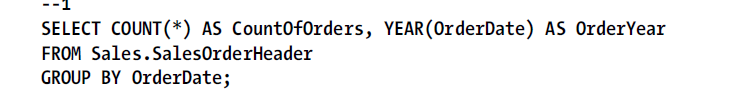
1. 聚集函数：SELECT <aggregate function>(col1> FROM <table>

COUNT: Counts the number of rows or the number of non-NULL values in a column.

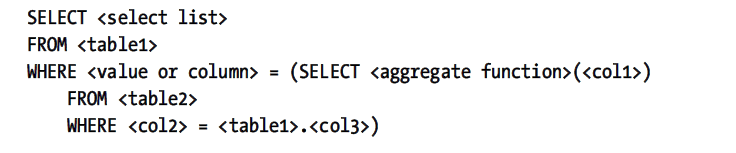
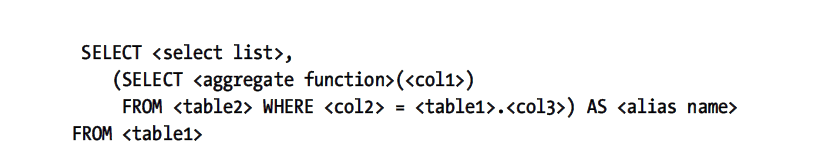
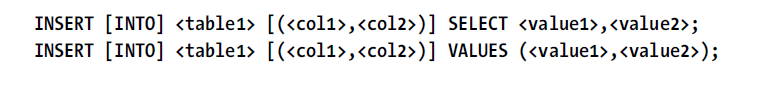
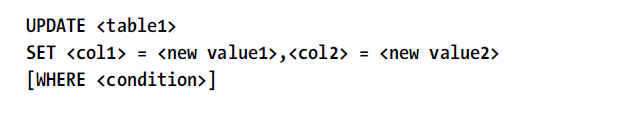
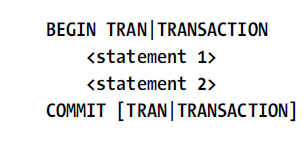
SUM:Add up the values in numeric or money data

AVG MIN MAX:average min max

聚集函数忽略NULL值，除了COUNT(\*)，COUNT(\*)包含所有的行

1. GROUP和聚集函数一起使用：SELECT <aggregate function>(<col1>), <col2> FROM <table> GROUP BY <col2>;在select中的非聚集的列必须出现在group子句中
2. 

SELECT <aggregate function>(<col1>), <col2> FROM <table> GROUP BY <col2> ORDER BY <col2>; If a nonaggregate column appears in the ORDER BY clause, it must also appear in the GROUP BY clause, just like the SELECT list.

1. SELECT <aggregate function1>(<col1>),<col2> FROM <table1> GROUP BY <col2> HAVING <aggregate function2>(<col3>) = <value>;having可以包含或者不包含出现在select中的聚集函数。也可以包含非聚集的出现在group by的列。Having是消除一些group的。
2. Fwghso
3. Distinct ： select distinct col from table; select sum(distinct col) from table;
4.  
5. 
6. DELETE [FROM] <table1> [WHERE <condition>];
7. 
8. 事物应该尽量的短，因为事物会锁定相关的表。
9. 

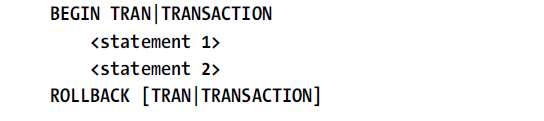
事务的例子：

BEGIN TRAN

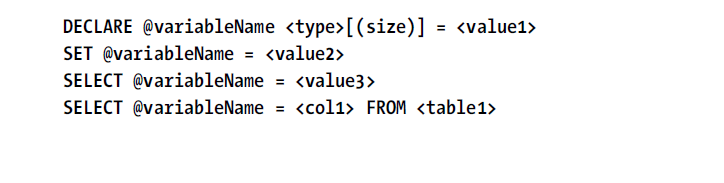
INSERT INTO table(col) VALUES(val);

INSERT INTO table(col) VALUES(val);

COMMIT TRAN;



1. T-SQL中使用变量必须先声明。



Set一次只能设置一个值，select可以一次设置多个值。

IF @i > 10 BEGIN

PRINT 'i > 10';

END

ELSE IF @i = 10 BEGIN

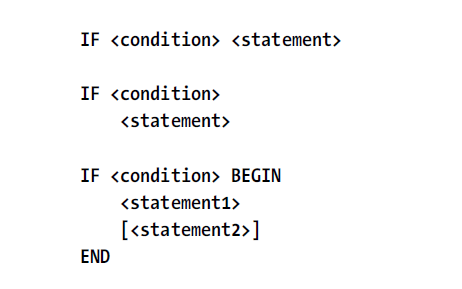
PRINT 'i = 10';

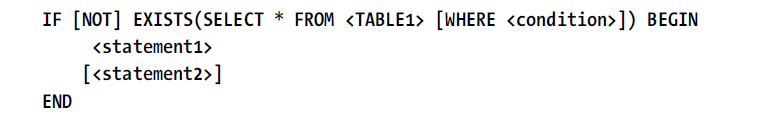
END

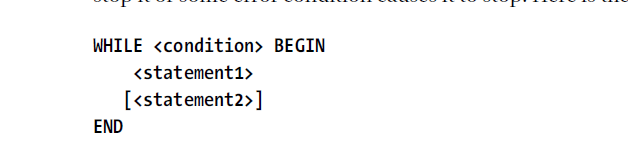
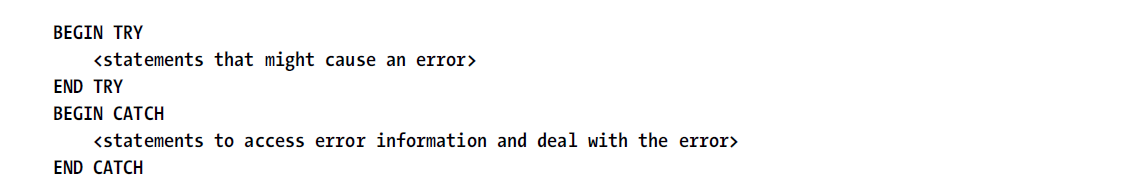
ELSE BEGIN

PRINT 'i < 10';

END;

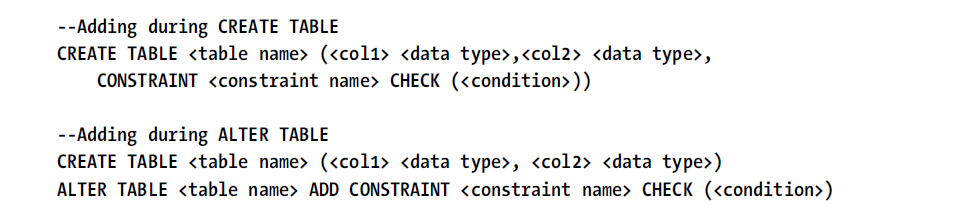
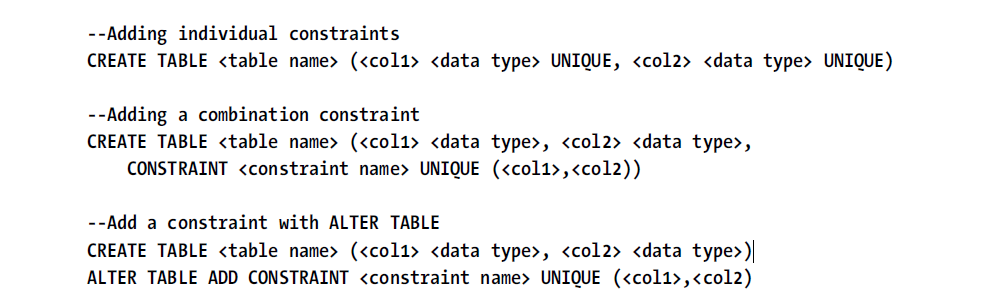


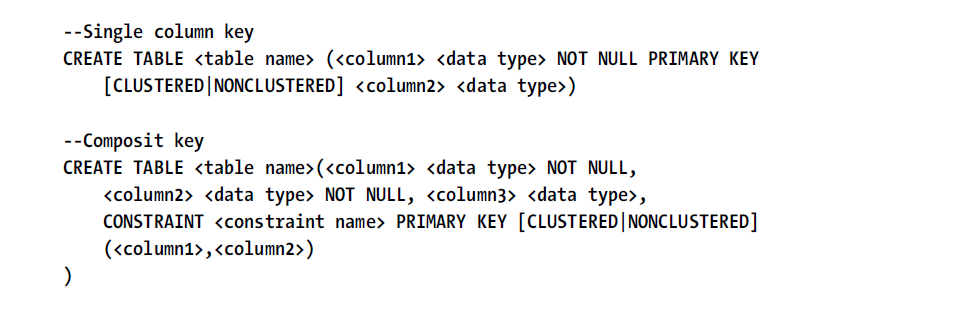
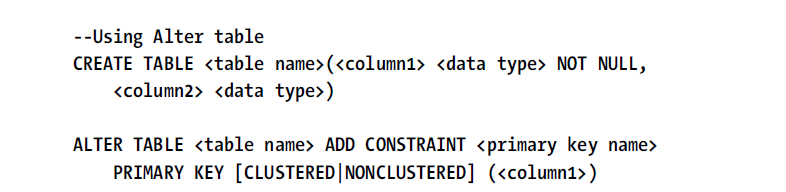
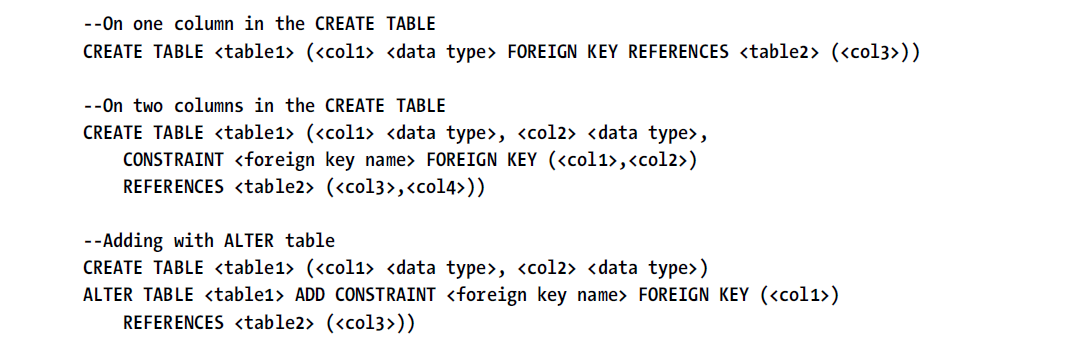
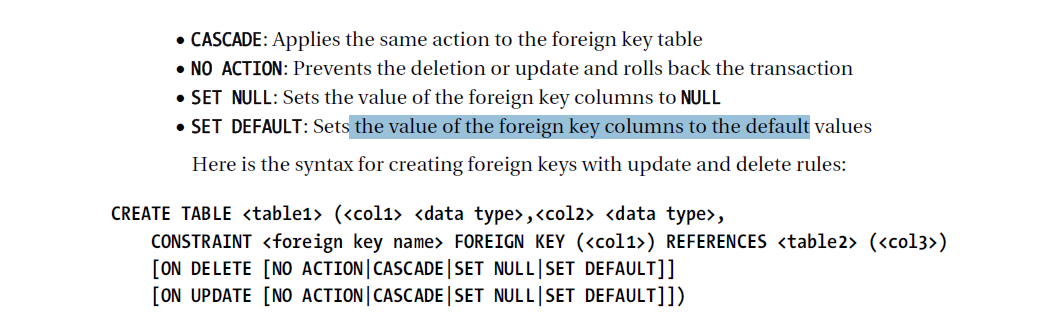
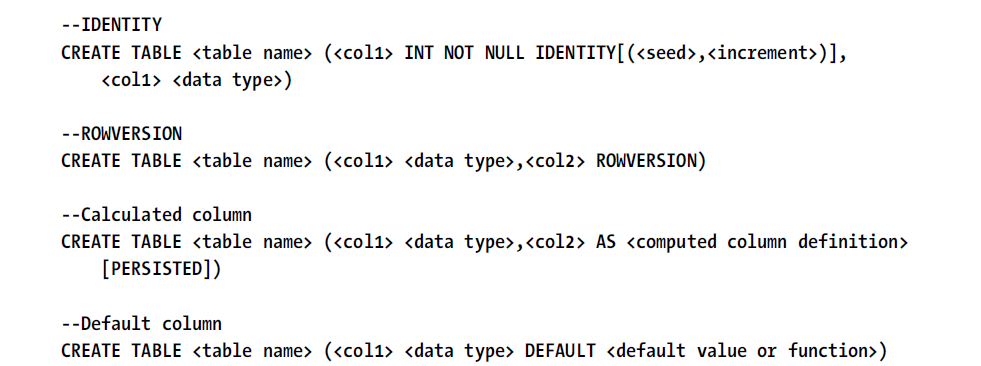
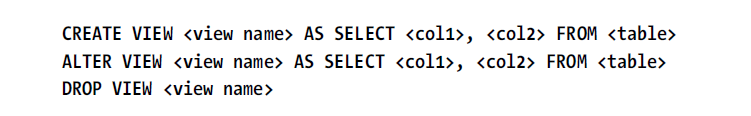


1. 
2. BREAK 和 CONTINUE退出或者继续循环。
3. @@ERROR全局变量，当它大于0，则表示出错。
4. GOTO语句
5. 

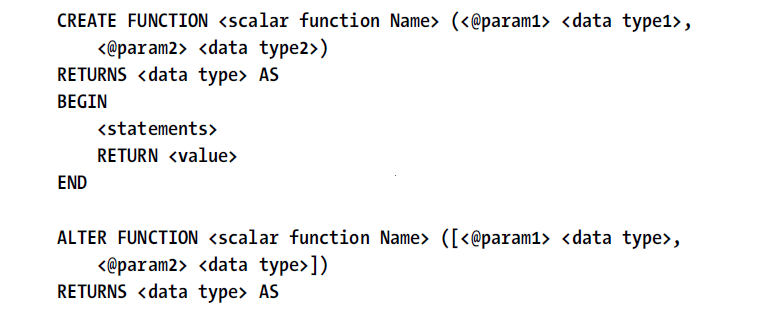
CREATE TABLE #tableName (<col1> <data type>,<col2> <data type>)

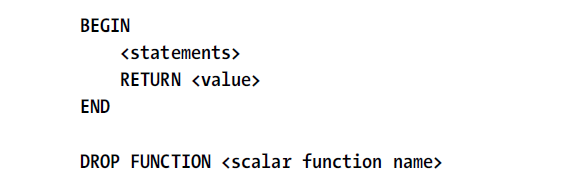
创建临时表#

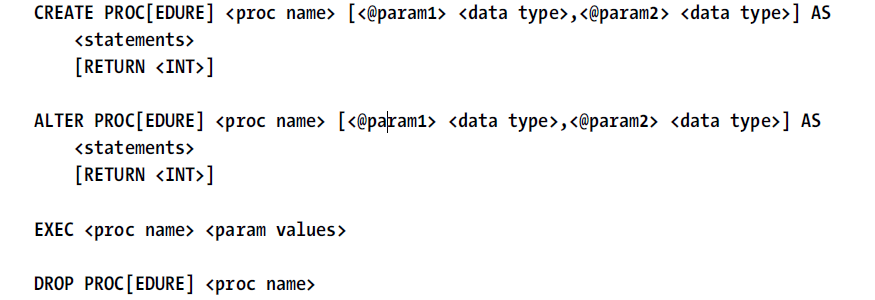
1. 
2. 

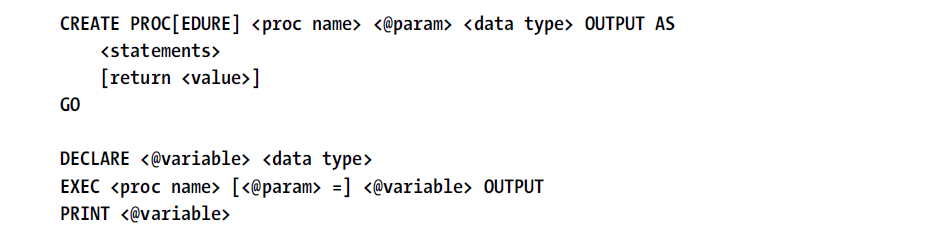
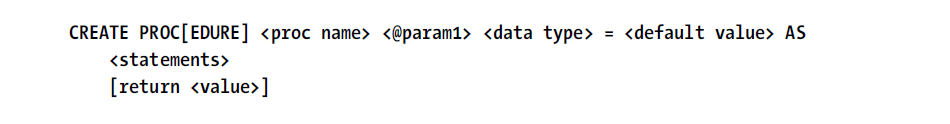
     

1. UDF：用户定义函数。UDF不能修改数据库数据，UDF可以调用其它UDF.UDF必须包含一个返回值。









1. 多个CTE. WITH <cteName1> AS (SELECT <col1> FROM <table1>),

<cteName2> AS (SELECT <col2> FROM <table2>),

<cteName3> AS (SELECT <col3> FROM <table3>)

SELECT <col1>,<col2>,<col3> FROM <cteName1> INNER JOIN <cteName2> ON <join condition1> INNER JOIN ON <join condition2>;

1. ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
2. BEGIN END定义了一个语句块，但没有定义作用域。