**Uniqueidentifier vs. IDENTITY**

Uniqueidentifier :

DECLARE @GUID uniqueidentifier  
SET @GUID = NEWID()  
INSERT Item VALUES (@GUID,'Yak Hoof')

CREATE TABLE cust

(

CustomerID **uniqueidentifier** NOT NULL

DEFAULT newid(),

Company varchar(30) NOT NULL,

ContactName varchar(60) NOT NULL,

Address varchar(30) NOT NULL,

City varchar(30) NOT NULL,

StateProvince varchar(10) NULL,

PostalCode varchar(10) NOT NULL,

CountryRegion varchar(20) NOT NULL,

Telephone varchar(15) NOT NULL,

Fax varchar(15) NULL

);

IDENTITY : 插入数据时，不需要提供数值， 否则出错

CREATE TABLE new\_employees

(

id\_num int IDENTITY(1,1),

fname varchar (20),

minit char(1),

lname varchar(30)

);

INSERT new\_employees

(fname, minit, lname)

VALUES

('Karin', 'F', 'Josephs');

INSERT new\_employees

(fname, minit, lname)

VALUES

('Pirkko', 'O', 'Koskitalo');

区别：

1. Uniqueidentifier: 使用 NEWID() 获取， IDENTITY字段是自动获得
2. IDENTITY 通常是 INT 类型只占 4 个字节， 唯一识别是 16字节的字符， 对于索引性能不同
3. IDENTITY 插入完以后可以获得其值， 唯一识别只能在插入前获取其中， 然后赋值插入。
4. 唯一识别具有全局性， 全局唯一 GUID , 是个多服务器，多环境下作唯一识别用， 如多服务器数据复制合并
5. Uniqueidentifier: 是SQL 的一种数据类型， IDENTITY 通常是 INT 类型

SELECT MIN($IDENTITY), MAX($IDENTITY)

FROM Students;

SELECT MIN($IDENTITY), MAX($IDENTITY)

FROM Teachers;

SELECT IDENT\_CURRENT('students'),IDENT\_SEED('students'), IDENT\_INCR('students');

SELECT IDENT\_CURRENT('teachers'),IDENT\_SEED('teachers'), IDENT\_INCR('teachers');

这些都是跟最后一次操作有关系

select SCOPE\_IDENTITY(), @@IDENTITY

 [@@IDENTITY](http://msdn.microsoft.com/en-us/library/ms187342.aspx) returns the last identity value generated for any table in the current session, across all scopes. **You need to be careful here**, since it's across scopes. You could get a value from a trigger, instead of your current statement.

 [SCOPE\_IDENTITY()](http://msdn.microsoft.com/en-us/library/ms190315.aspx) returns the last identity value generated for any table in the current session and the current scope. **Generally what you want to use**.

 [IDENT\_CURRENT('tableName')](http://msdn.microsoft.com/en-us/library/ms175098.aspx) returns the last identity value generated for a specific table in any session and any scope. This lets you specify which table you want the value from, in case the two above aren't quite what you need (**very rare**). Also, as @[Guy Starbuck](http://stackoverflow.com/questions/42648/best-way-to-get-identity-of-inserted-row#42665) mentioned, "You could use this if you want to get the current IDENTITY value for a table that you have not inserted a record into."

1.) Find statistics on a table:

exec sp\_help 'dbo.table\_name\_old';

2.) Create a duplicate, identical new table, except add an identity field on the PK field where it had been before.

3.) Disable the identity to move data.

SET IDENTITY\_INSERT dbo.table\_name ON -- after disable, you need to privder unique id

4.) Transfer the data.

INSERT INTO dbo.table\_name\_new

(

field1, field2, etc...

)

SELECT

field1, field2, etc...

FROM

dbo.table\_name\_old;

作用范围只在此连接会话有效， 不影响其他连接会话

1)

SET IDENTITY\_INSERT Test ON;

insert test(id, name) values(60, 'Tommy');

2)

SET IDENTITY\_INSERT Test OFF;

SELECT MAX($IDENTITY) FROM Test;

insert test(name) values('Tommy');

SELECT @@IDENTITY;

删除添加字段

ALTER table customer

DROP COLUMN hiredate, salary;

ALTER Table Customer

Add

memberdate datetime not null,

credit int default(100)

CHECK : 应用在表上的字段上, 约束是在表上 , 表达式是字段

ALTER Table Customer

Add

constraint ck\_credit

check(credit > 100 AND credit < 10000)

OVER Clause

over不能单独使用，要和分析函数：lag(..), lead(..) , rank(), dense\_rank(), row\_number()等一起使用。

OVER (

[ <PARTITION BY clause> ]

[ <ORDER BY clause> ]

[ <ROW or RANGE clause> ]

)

SELECT

ROW\_NUMBER() OVER(PARTITION BY cust\_id order by odate) as rowid ,

customer.id as cid,

concat(customer.fname, ' ', customer.lname) as cname,

orders.id as oid, odate

FROM orders

INNER JOIN customer ON (orders.cust\_id = customer.id)

ORDER BY cid, rowid



PARTITION BY cust\_id - 按 cust\_id 分组

order by odate - 按 odate 顺序排序， 生成行号， 排名

SELECT

ROW\_NUMBER() OVER(order by odate) as rowid ,

customer.id as cid,

concat(customer.fname, ' ', customer.lname) as cname,

orders.id as oid, odate

FROM orders

INNER JOIN customer ON (orders.cust\_id = customer.id)

ORDER BY cid, rowid

没有PARTITION BY cust\_id

