**IDENT\_CURRENT (Transact-SQL)**

Returns the last identity value generated for a specified table or view. The last identity value generated can be for any session and any scope.

Topic link icon[Transact-SQL Syntax Conventions](http://msdn.microsoft.com/en-us/library/ms177563.aspx)

http://i.msdn.microsoft.com/Global/Images/clear.gif Syntax

IDENT\_CURRENT( '*table\_name*' )

http://i.msdn.microsoft.com/Global/Images/clear.gif Arguments

*table\_name*

Is the name of the table whose identity value is returned. *table\_name* is **varchar**, with no default.

http://i.msdn.microsoft.com/Global/Images/clear.gif Return Types

**numeric(38,0)**

http://i.msdn.microsoft.com/Global/Images/clear.gif Exceptions

Returns NULL on error or if a caller does not have permission to view the object.

In SQL Server, a user can only view the metadata of securables that the user owns or on which the user has been granted permission. This means that metadata-emitting, built-in functions such as IDENT\_CURRENT may return NULL if the user does not have any permission on the object. For more information, see [Metadata Visibility Configuration](http://msdn.microsoft.com/en-us/library/ms187113.aspx) and [Troubleshooting Metadata Visibility](http://msdn.microsoft.com/en-us/library/ms190785.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Remarks

IDENT\_CURRENT is similar to the SQL Server 2000 identity functions SCOPE\_IDENTITY and @@IDENTITY. All three functions return last-generated identity values. However, the scope and session on which *last* is defined in each of these functions differ:

* IDENT\_CURRENT returns the last identity value generated for a specific table in any session and any scope.
* @@IDENTITY returns the last identity value generated for any table in the current session, across all scopes.
* SCOPE\_IDENTITY returns the last identity value generated for any table in the current session and the current scope.

When the IDENT\_CURRENT value is NULL (because the table has never contained rows or has been truncated), the IDENT\_CURRENT function returns the seed value.

Failed statements and transactions can change the current identity for a table and create gaps in the identity column values. The identity value is never rolled back even though the transaction that tried to insert the value into the table is not committed. For example, if an INSERT statement fails because of an IGNORE\_DUP\_KEY violation, the current identity value for the table is still incremented.

Be cautious about using IDENT\_CURRENT to predict the next generated identity value. The actual generated value may be different from IDENT\_CURRENT plus IDENT\_INCR because of insertions performed by other sessions.

http://i.msdn.microsoft.com/Global/Images/clear.gif Examples

**A. Returning the last identity value generated for a specified table**

The following example returns the last identity value generated for the Person.Address table in the AdventureWorks2008R2 database.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl27_ctl00_ctl00_code');" \o "Copy Code)

USE AdventureWorks2008R2;

GO

SELECT IDENT\_CURRENT ('Person.Address') AS Current\_Identity;

GO

**B. Comparing identity values returned by IDENT\_CURRENT, @@IDENTITY and SCOPE\_IDENTITY**

The following example shows the different identity values that are returned by IDENT\_CURRENT, @@IDENTITY, and SCOPE\_IDENTITY.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl27_ctl00_ctl01_code');" \o "Copy Code)

USE AdventureWorks2008R2;

GO

IF OBJECT\_ID(N't6', N'U') IS NOT NULL

DROP TABLE t6;

GO

IF OBJECT\_ID(N't7', N'U') IS NOT NULL

DROP TABLE t7;

GO

CREATE TABLE t6(id int IDENTITY);

CREATE TABLE t7(id int IDENTITY(100,1));

GO

CREATE TRIGGER t6ins ON t6 FOR INSERT

AS

BEGIN

INSERT t7 DEFAULT VALUES

END;

GO

--End of trigger definition

SELECT id FROM t6;

--IDs empty.

SELECT id FROM t7;

--ID is empty.

--Do the following in Session 1

INSERT t6 DEFAULT VALUES;

SELECT @@IDENTITY;

/\*Returns the value 100. This was inserted by the trigger.\*/

SELECT SCOPE\_IDENTITY();

/\* Returns the value 1. This was inserted by the

INSERT statement two statements before this query.\*/

SELECT IDENT\_CURRENT('t7');

/\* Returns value inserted into t7, that is in the trigger.\*/

SELECT IDENT\_CURRENT('t6');

/\* Returns value inserted into t6. This was the INSERT statement four statements before this query.\*/

-- Do the following in Session 2.

SELECT @@IDENTITY;

/\* Returns NULL because there has been no INSERT action

up to this point in this session.\*/

SELECT SCOPE\_IDENTITY();

/\* Returns NULL because there has been no INSERT action

up to this point in this scope in this session.\*/

SELECT IDENT\_CURRENT('t7');

/\* Returns the last value inserted into t7.\*/