**@@IDENTITY (Transact-SQL)**

Is a system function that returns the last-inserted identity value.

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

@@IDENTITY

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

**numeric(38,0)**

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

After an INSERT, SELECT INTO, or bulk copy statement is completed, @@IDENTITY contains the last identity value that is generated by the statement. If the statement did not affect any tables with identity columns, @@IDENTITY returns NULL. If multiple rows are inserted, generating multiple identity values, @@IDENTITY returns the last identity value generated. If the statement fires one or more triggers that perform inserts that generate identity values, calling @@IDENTITY immediately after the statement returns the last identity value generated by the triggers. If a trigger is fired after an insert action on a table that has an identity column, and the trigger inserts into another table that does not have an identity column, @@IDENTITY returns the identity value of the first insert. The @@IDENTITY value does not revert to a previous setting if the INSERT or SELECT INTO statement or bulk copy fails, or if the transaction is rolled back.

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.

@@IDENTITY, SCOPE\_IDENTITY, and IDENT\_CURRENT are similar functions because they all return the last value inserted into the IDENTITY column of a table.

@@IDENTITY and SCOPE\_IDENTITY return the last identity value generated in any table in the current session. However, SCOPE\_IDENTITY returns the value only within the current scope; @@IDENTITY is not limited to a specific scope.

IDENT\_CURRENT is not limited by scope and session; it is limited to a specified table. IDENT\_CURRENT returns the identity value generated for a specific table in any session and any scope. For more information, see [IDENT\_CURRENT (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms175098.aspx).

The scope of the @@IDENTITY function is current session on the local server on which it is executed. This function cannot be applied to remote or linked servers. To obtain an identity value on a different server, execute a stored procedure on that remote or linked server and have that stored procedure (which is executing in the context of the remote or linked server) gather the identity value and return it to the calling connection on the local server.

Replication may affect the @@IDENTITY value, since it is used within the replication triggers and stored procedures. @@IDENTITY is not a reliable indicator of the most recent user-created identity if the column is part of a replication article. You can use the SCOPE\_IDENTITY() function syntax instead of @@IDENTITY. For more information, see [SCOPE\_IDENTITY (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms190315.aspx)

|  |
| --- |
| **NoteNote** |
| The calling stored procedure or Transact-SQL statement must be rewritten to use the SCOPE\_IDENTITY() function which will return the latest identity used within the scope of that user statement, and not the identity within the scope of the nested trigger used by replication. |

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

The following example inserts a row into a table with an identity column (LocationID) and uses @@IDENTITY to display the identity value used in the new row.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl23_ctl00_ctl00_code');" \o "Copy Code)

USE AdventureWorks2008R2;

GO

--Display the value of LocationID in the last row in the table.

SELECT MAX(LocationID) FROM Production.Location;

GO

INSERT INTO Production.Location (Name, CostRate, Availability, ModifiedDate)

VALUES ('Damaged Goods', 5, 2.5, GETDATE());

GO

SELECT @@IDENTITY AS 'Identity';

GO

--Display the value of LocationID of the newly inserted row.

SELECT MAX(LocationID) FROM Production.Location;

GO