**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**فیلد identity**

* به صورت اتوماتیک مقدار دهی خواهد شد و نمی توانیم این فیلد را مقدار دهی کنیم

**فیلد identity دو خصوصیت دارد**

* **Identity Increment:** که مشخص می کند مقدار فیلد چند تایی افزایش پیدا کند، به طور پیش فرض مقدار آن یک است و می تواند اعداد مثبت و همین طور منفی نیز بگیرد.
* Identity Seed: مشخص می کند مقدار این فیلد از چه عددی شروع شود، به طور پیش فرض از یک شروع می شود ولی می تواند از اعداد منفی و هر عدد مثبتی برای شروع فیلد Identity استفاده کرد.

**تغیر حالت Identity به حالت On – Off**

|  |
| --- |
| SET IDENTITY\_INSERT [Table Name] OFF |
| SET IDENTITY\_INSERT [Table Name] ON |

برای استفاده از فیلد identity در sql از ساختار زیر استفاده مکنیم :

|  |  |
| --- | --- |
| CREATE TABLE Persons  (  P\_Id int PRIMARY KEY IDENTITY(-200,1),  LastName varchar(255) NOT NULL,  ) | CREATE TABLE Persons  (  P\_Id int PRIMARY KEY IDENTITY,  LastName varchar(255) NOT NULL,  ) |

**نکته مهم**

اگر رکوردهای جدول را با Delete حذف کنیم مقدار آخرین Identity حفظ خواهد شد و درج با عدد بعدی آن صورت   
می گیرد، اگر بخواهیم بعد از حذف داده ها درج با مقدار Seed تعریف شده صورت گیرد به جای Delete از دستور **TRUNCATE table\_name** استفاده می کنیم، به این ترتیب لاگ هم ذخیره نخواهد شد، ضمن اینکه سرعت حذف رکوردها بسیار بیشتر است.

**تابع DBCC CHECKIDENT**

جهت RESEED ایدی بکار میرود ، در صورت نبود کلیک خطا صادر میشود.

**پارامترهای مربوط به این دستور عبارتند از**

* : **Table\_Name**این پارامتر نام جدولی را مشخص می‌کند که می‌خواهید مقدار آن را دستکاری نمایید.
* : **NORESEED**در صورتیکه از این کلمه کلیدی به عنوان پارامتر دوم استفاده شود. باعث می‌شود که تغییر در مقدار Identity بوجود نیاید.
* : **RESEED**در صورتیکه از این کلمه کلیدی به عنوان پارامتر سوم استفاده شود. باعث می‌شود که تغییر در مقدار Identity با توجه به پارامتر چهارم بوجود آید.
* : **new\_reseed\_value**این پارامتر مشخص می‌کند که Identity باید به چه مقداری Reset شود و مقدار جدید با new\_reseed\_value+1 خواهد بود.

|  |
| --- |
| DBCC CHECKIDENT ('Test2', NORESEED) |
| DBCC CHECKIDENT ('Test2', RESEED, 0) |
| CREATE TABLE Test2  (  [ID] [smallint] PRIMARY KEY IDENTITY(1,1)  )  INSERT INTO Test2 DEFAULT VALUES;  GO 50    SELECT [ID] FROM Test2    SELECT IDENT\_CURRENT('Test2') |

**نحوه بدست آوردن مقادیر Identity**

مقدار Identity پس از درج به آن تخصیص می‌یابد چنانچه بخواهید به این مقدار دسترسی پیدا کنید چندین روش به ازای اینکار وجود دارد

* استفاده از متغییر سیستمی  **@@Identity**
* استفاده از تابع **Scope\_Identity ()**
* استفاده از تابع **Ident\_Current**

برای بدست آوردن یک **Identity** کافی است که پس از درج رکورد در جدول مورد نظر متغییر سیستمی **@@Identity** و یا توابع **Scope\_Identity** و یا **Ident\_Current** را Selectکنید.

هر سه این توابع مقدار Identity ایجاد شده برای جداول را نمایش می‌دهند. **اما تفاوت هایی باهم دارند**

**متغییر سیستمی Identity@@ :**

این متغییر سیستمی حاوی آخرین Identity ایجاد شده به ازای Session جاری شما است. لازم به ذکر است اگر به واسته Insert شما، Identity دیگری در یک حوزه دیگر مانند یک Trigger ایجاد شود مقدار موجود در این متغییر حاوی آخرین Identity ایجاد شده است. یعنی Identity ایجاد شده توسط آن تریگر و نه خود جدول.

**تابع ()Scope\_Identity :**

با استفاده از این تابع می‌توانیم آخرین Identify ایجا دشده به ازای Session جاری را بدست آوریم. لازم به ذکر است مقادیر Identity ایجاد شده توسط سایر حوزه‌ها تاثیر در مقدار بازگشتی توسط این تابع ندارد.

**تابع ident\_Current :**

این تابع آخرین مقدار Identity موجود در یک جدول را نمایش می‌دهد. ذکر این نکته ضروری است که Identity ایجاد شده توسط سایر Session ها هم روی خروجی این تابع تاثیرگذار است. چون این تابع آخرین Identity موجود در جدول را به شما نمایش می‌دهد و نه Identity ایجاد شده به ازای یکSession   را.

**مثال**

|  |
| --- |
| CREATE TABLE Employees1  (  ID int IDENTITY(1,1),  FirstName NVARCHAR(50),  LastName NVARCHAR(50)  ) |
| INSERT INTO Employees1 (FirstName,LastName) VALUES (N'محمدحسین',N'فخرآوری')  GO  SELECT @@IDENTITY AS [@@IDENTITY]  SELECT SCOPE\_IDENTITY() AS [SCOPE\_IDENTITY()]  SELECT IDENT\_CURRENT('Employees1') AS [IDENT\_CURRENT('Employees1')] |

[**بررسی دستور TRUNCATETable و Delete**](http://www.dotnettips.info/post/1348/%d8%a8%d8%b1%d8%b1%d8%b3%db%8c-%d8%af%d8%b3%d8%aa%d9%88%d8%b1-truncate-table-%d9%88-delete)

دستور Delete دارد این است که این دستور Log کلیه رکوردهایی را که قرار است حذف کند در Log File می‌نویسد. این Log شامل اصل رکورد، تاریخ و زمان حذف، نام کاربر و... می‌باشد. شاید الان متوجه شوید که دستور Delete چرا در برخی از مواقع که قرار است حجم زیادی از اطلاعات را حذف نماید به کٌندی این کار را انجام می‌دهد. چون باید Log رکوردهای حذف شده در Log File نوشته شود.

**دستور TRUNCATE**

|  |
| --- |
| TRUNCATE TABLE **Customers** |

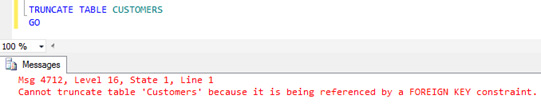
Truncate در لغت به معنی بریدن و کوتاه کردن می‌باشد. با استفاده دستور Truncate Table می‌توانید محتوای کلیه رکوردهای موجود در یک جدول را در کسری از ثانیه حذف کنید.

نکته مهمی که باید درباره دستور Truncate Table بدانید این است که تاثیر استفاده از این دستور بر روی کلیه رکوردها بوده و به هیچ عنوان نمی‌توان برای این دستور شرط (Where) اعمال نمود.

**اما  در مورد دستور Truncate Table و  Delete باید به نکات زیر توجه کنید**

* دستور Truncate Table فاقد قسمت شرط Where Clause می‌باشد در صورتیکه دستور Delete دارای قسمت شرط Where Clause است .
* دستور Truncate Table در Log File آدرس Page و مقدار فضای آزاد شده کمترین میزان Log را می‌نویسد  اما در صورتیکه دستور Delete در Log هر رکوردی را که قرار است حذف شود را در Log File ثبت می‌نماید.
* دستور Truncate Table باعث می‌شود که رکوردهای موجود در جدول واقعاً حذف نشوند بلکه Extentهای مربوط به آن Pageها علامت Empty خورده تا دفعات بعد مورد استفاده قرار گیرند اما دستور Delete به طور فیزیکی محتوای Pageها  مربوط به جدول را خالی می‌کند.
* دستور Truncate Table مقدار Identity را Reset کرده و آن را به Seed (هسته/مقدار اولیه) بر می‌گرداند. در صورتیکه دستور Delete تاثیری بر روی مقدار Identity ندارد
* دستور Truncate Table تنها توسط کاربرانی قابل اجرا است که نقش DB\_Owner و یا SysAdmin را داشته باشند در صورتیکه دستور Delete توسط هر کاربری که مجوز Delete بر روی جدول را  داشته باشد قابل اجرا می‌باشد.
* پس از اجرای دستور Truncate Table تعداد رکوردهای حذف شده نمایش داده نمی‌شود. در صورتیکه هنگام اجرای دستور Delete تعداد رکوردهای حذف شده نمایش داده می‌شود.

**نکته : پس از Truncate شدن رکوردها امکان بازگشت آنها وجود ندارد.**

[](http://www.dotnettips.info/file/image?name=Truncate01.jpg)

در صورتیکه در دستور Delete امکان حذف رکوردها به ازای جداولی که دارای Relation هستند وجود دارد. فقط باید به این نکته توجه کنید که ترتیب حذف رکوردها از جداول Master و Detail را رعایت کنید.

