

استفاده از گزینه ی DELETE CASCADE برای رسیدگی به Child key به Removal در ارتباطات کلید خارجی

استفاده از گزینه ی DELETE CASCADE در تعاریف محدودیت کلید خارجی به معنای اجرای بهتر و کد کمتر در هنگام حذف رکوردها از جدول هایی است که دارای یک رابطه ی parent-child تعریف شده می باشند.

توضیحات

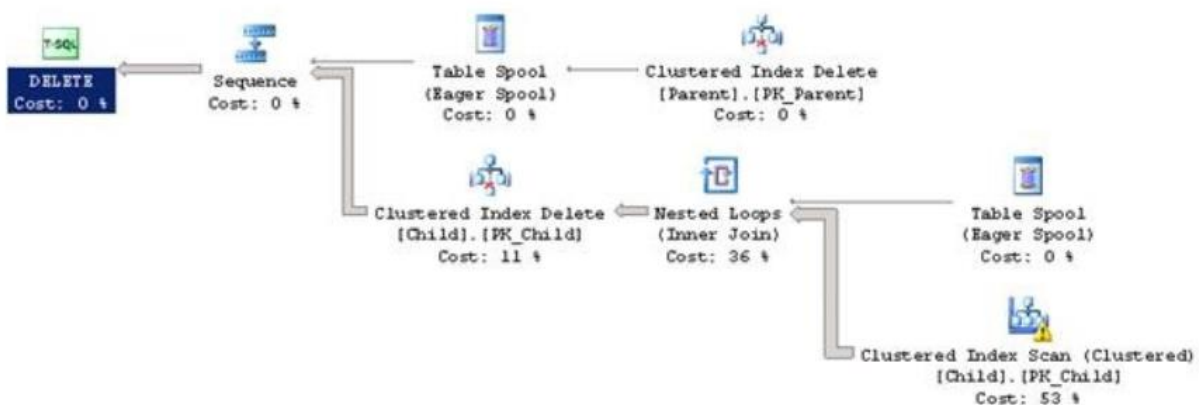
ابتدا اجازه بدهید که تایید کنیم که نمودار جاری ما دارای گزینه ی تعریف شده ی DELETE CASCADE روی کلید خارجی، بین جدول Parent و Child می باشد. در اینجا وضعیت SQL به همراه نتیجه را مشاهده می کنید.

name	delete_referential_action_desc
FK_Child_Parent	CASCADE

اکنون که تایید کرده ایم که این گزینه ی تعریف شده را داریم، اجازه بدهید با استفاده از عبارت SQL زیر یک رکورد را از جدول Parent حذف کنیم.

```
DELETE FROM [dbo].[Parent] where ParentID=82433
```

با نگاه به explain plan برای این query، متذکر می شویم که SQL Optimizer ابتدا رکوردهای child را حذف کرده و سپس delete را روی جدول Parent اجرا می کند. به این خاطر لازم است به هر جدول تنها یک بار دسترسی داشته باشد.



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

اکنون اجازه بدهید گزینه ی **DELETE CASCADE** را از تعریف کلید خارجی (**foreign key**) حذف کرده و تفاوت ها را مشاهده کنیم. برای این کار به حذف و ایجاد مجدد کلید خارجی بدون گزینه ی **DELETE CASCADE** نیاز داریم. در اینجا عبارت های **SQL** برای انجام این تغییرات را مشاهده می کنید:

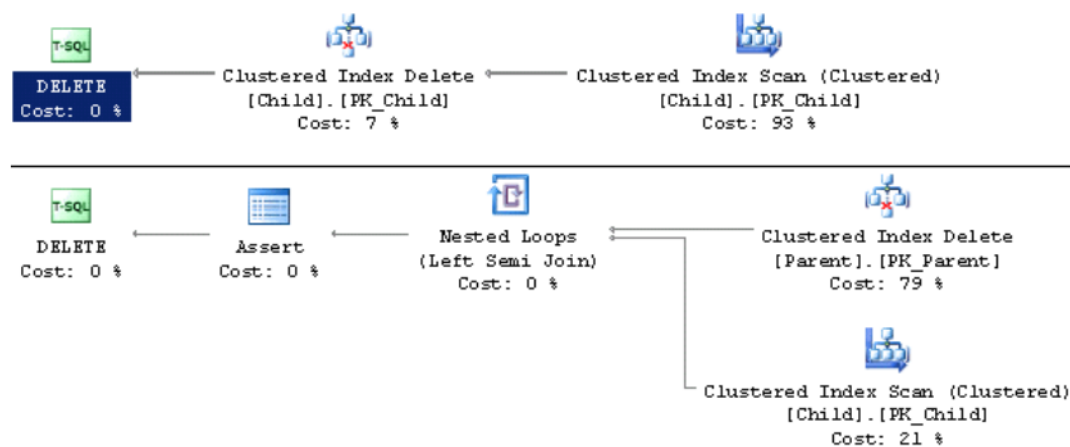
```
ALTER TABLE [dbo].[Child] DROP CONSTRAINT [FK_Child_Parent]
```

```
ALTER TABLE [dbo].[Child] WITH CHECK
ADD CONSTRAINT [FK_Child_Parent] FOREIGN KEY([ParentID])
REFERENCES [dbo].[Parent] ([ParentID])
```

زمانی که کلید خارجی دوباره ایجاد شده می توانیم یک عملکرد حذف دیگر نیز اجرا کنیم تا اگر تفاوتی در اجرا وجود دارد، مشاهده کنیم. یک موضوع قابل ذکر این است که بدون گزینه ی **DELETE CASCADE** نیاز به اجرای یک عبارت **delete** دیگر داریم تا رکوردها را از اولین جدول **Child** حذف کنیم. در اینجا عبارت های **SQL** را برای اجرای **delete** مشاهده می کنید:

```
DELETE FROM [dbo].[Child] where ParentID=62433
DELETE FROM [dbo].[Parent] where ParentID=62433
```

با نگاه کردن به **explain plan** برای این عبارت ها، مشاهده می کنیم که آنها کاملاً شبیه هستند، تنها تفاوت این است که چون ما در حال اجرای عبارت های مجزای **delete** هستیم، لازم است جدول **Child** برای بار دوم محدودیت کلید خارجی را برای بار دوم، هنگام حذف از جدول **Parent**، چک کند.



با استفاده از نتایج **SQL Profiler** از هر **query**، می توانیم این اسکن اضافه از جدول **Child** را تایید کنیم، که در واقع به این

معناست که گزینه ی **DELETE CASCADE** بهتر اجرا می کند. در زیر مشاهده می کنیم که گزینه ی **DELETE CASCADE** در هر رده از منابع کمتری استفاده می کند و حدود 20 درصد سریعتر اجرا می کند.

	CPU	Reads	Writes	Duration
No Delete Cascade	344	28488	0	399
Delete Cascade	250	14249	0	312



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>