چگونه ترتیب اتصال روی query plan تاثیر می گذارد

ترتیبی که طبق آن جدول ها در query های شما به یکدیگر متصل هستند، می تواند یک تاثیر نمایشی روی چگونگی اجراهای query شما داشته باشد. اگر query در ابتدا به جدول های بزرگتر و سپس به جدول های کوچکتر متصل شود، این امر باعث انجام پردازش های غیرضروری بسیاری توسط موتور SQL می شود.

به طور کلی SQL Server Optimizer در ابتدا سعی خواهد کرد به جدول هایی متصل شود که اجازه ی کار با کوچکترین مجموعه ی نتیجه ی ممکن را به آن می دهند. برای انجام این کار بهینه ساز (optimizer) سعی خواهد کرد بررسی کند که کدام ترتیب اتصال کوچکترین مجموعه نتیجه را در اولین پردازش ها، ارائه می دهد؛ اما در query های خیلی پیچیده همه ی ترکیبات ممکن را امتحان نمی کند، چرا که ممکن است باعث فشردگی در منبع شود (resource intensive). به عنوان بهترین تمرین باید سعی کنید که اتصالات جدول خود را طوری تنظیم کنید که مجموعه نتیجه را که در ابتدا ملحق می شود، به حداقل برساند. قبل از شروع اجازه بدهید یک ایندکس به ستون در جدولی که به عنوان شرط اتصال استفاده خواهیم کرد، اضافه کنیم (به این مسئله بیشتر خواهیم پرداخت.).

```
CREATE NONCLUSTERED INDEX idxChild_ParentID
ON [dbo].[Child] ([ParentID])

-- cleanup statements
--DROP INDEX Child.idxChild_ParentID
```

از آنجایی که در بیشتر موارد این مسئله زمانی مطرح می شود که queryها واقعا پیچیده می شوند و بهینه ساز برنامه های زیادی برای ارزیابی دارد، به عنوان مثال اتصالات چند جدول، برای توضیح واضح تر این مسئله از نیروی ترتیب با یک query ساده استفاده خواهیم کرد. در اینجا کدی را مشاهده می کنید که ترتیب اتصالات ضعیف ما را توضیح می دهد:

```
SELECT P.ParentID,C.ChildID,S.SmallID
  FROM [dbo].[Parent] P INNER JOIN
     [dbo].[Child] C ON C.ParentID=P.ParentID INNER JOIN
     [dbo].[Small] S ON S.SmallID=C.ParentID
 OPTION (FORCE ORDER)
```

با دقت به توضیح برنامه برای این query می بینیم که جدول های Parent و Child در ابتدا به هم متصل شده و نتیجهِ آن 1899980 ردیف می باشد که پس از آن به جدول Small پیوسته که مجموعه ی رکورد نهایی را به 95 ردیف کاهش می دهد.

اکنون اجازه بدهید آنها را به ترتیب درست متصل کنیم، بنابراین کوچکترین جدول در ابتدا ملحق می شود. در اینجا وضعیت SQL را مشاهده می کنید:

```
SELECT P.ParentID,C.ChildID,S.SmallID
  FROM [dbo].[Small] S INNER JOIN
     [dbo].[Parent] P ON S.SmallID=P.ParentID INNER JOIN
     [dbo].[Child] C ON P.ParentID=C.ParentID
```

با دقت به explain plan برای این query مشاهده می کنیم که جدول Parent ابتدا به جدول Small ملحق شده و نتیجه 5 ردیف می باشد که پس از آن به جدول Child ملحق می شود که رکورد نهایی که 95 ردیف می باشد را تولید می کند.

تنها نگاه کردن به explain plans اطلاعات کافی در مورد query دوم به ما ارائه می دهد و درمی یابیم که query دوم بهتر عمل می کند. اما اجازه بدهید فقط برای تایید به استاتیک های SQL Profiler نگاهی داشته باشیم. همانطور که ار تصویر زیر در می یابیم اتصال به جدول Small در ابتدا به طور واضح تعداد داده های query را که باید پردازش شوند، کاهش می دهد.بنابراین کاهش منابع نیاز به اجرای این query دارد.

	CPU	Reads	Writes	Duration
Poor Join Order	265	5935	0	309
Good Join Order	0	35	0	0