

اگر بخواهیم اولین رکورد از یک جدول را توسط EF درخواست نماییم از متد First یا FirstOrDefault استفاده می‌شود. برای مثال واکشی اولین رکورد از جدول Student به صورت زیر است:

```
var student=context.Students.FirstOrDefault();
```

در این حالت اولین رکورد از جدول student واکشی می‌شود و اگر رکوردی موجود نباشد یک مقدار null بازگشت داده می‌شود. حال اگر بخواهید به جای اولین رکورد آخرین رکورد را واکشی نمایید چطور؟ برای یافتن آخرین رکورد در لیست‌های Generic و کلا لیست‌های Enumerable از متد LastOrDefault استفاده می‌شود. با این حال این متد توسط Entity Framework پشتیبانی نمی‌شود و در صورتی که از کد زیر استفاده کنید برنامه با خطا متوقف خواهد شد:

```
var student=context.Students.LastOrDefault();
```

دو راه حل برای رفع این مشکل به ذهن می‌رسد:

روش اول: می‌توان خروجی را ابتدا به یک نوع Enumerable مانند List تبدیل کرد و سپس از متد LastOrDefault استفاده کرد. کد زیر را در نظر بگیرید:

```
var student=context.Students.ToList().LastOrDefault();
```

در کد بالا ابتدا رکوردهای جدول Student از درون بانک اطلاعات به صورت کامل واکشی شده و سپس رکورد آخر از میان آنها جدا می‌شود. این حالت در حالی که رکوردهای کمی در جدول وجود داشته باشد روش بدی به حساب نمی‌آید ولی اگر تعداد رکوردها زیاد باشد (اکثر مواقع نیز به همین شکل است) روش مناسبی نمی‌باشد و باعث کندی برنامه می‌شود.

روش دوم: با توجه به اینکه تنها به یک رکورد (آخرین رکورد) نیاز داریم بهتر است یک رکورد هم واکشی شود. در این روش برای اینکه بتوان به آخرین رکورد رسید ابتدا رکوردهای جدول را به صورت نزولی مرتب می‌کنیم و سپس از متد FirstOrDefault برای واکشی آخرین رکورد استفاده می‌نماییم. برای مثال:

```
var student=context.Students.OrderByDescending(s=>s.Id).FirstOrDefault();
```

در کد بالا ابتدا رکوردها را بر اساس فیلد مورد نظر به صورت نزولی مرتب کرده ایم (در نظر داشته باشید عملیات مرتب سازی را می‌توان بر اساس فیلدی که مورد نظر است انجام داد) و پس از آن با توجه به اینکه رکوردها به صورت نزولی مرتب شده اند و رکورد آخر به اول منتقل شده است از متد FirstOrDefault جهت دسترسی به آخرین رکورد که در حال حاضر اول لیست رکوردها است استفاده شده است. سرعت این روش به مراتب از روش اول بیشتر می‌باشد. برای بالا رفتن سرعت مرتب سازی در جداول بزرگ نیز می‌توان از تدابیری همچون Index گذاری بر روی فیلدها در DataBase استفاده کرد (با توجه به فیلدی که قرار است مرتب سازی بر اساس آن انجام شود).

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۰۱ ۸:۳۸

ممنون. روش دوم به 1 select top در حین استفاده از SQL Server ترجمه میشه.

نویسنده: سامان هاشمی
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۱:۵۴

مورد اول اصلا توصیه نکنید بعدها به دلیل مشکل کارآیی که داره خیلی اذیت میکنه همون مورد دوم تنها گزینه و بهترین گزینه است!

نویسنده: ابوالفضل
تاریخ: ۱۳۹۲/۰۵/۰۲ ۱۲:۳۴

یک نکته اینکه : زمانی که قصد داریم آخرین رکورد افزوده شده رو به این طریق و بر اساس فیلدی غیر از کلید واکنشی کنیم (به فرض تاریخ فاکتور و ...) حتما باید برای آن فیلد در صورت کلید نبودنش ، ایندکس ایجاد کنیم تا واکنشی در کوتاهترین زمان ممکن در حجم بالای اطلاعات صورت گیرد .

نویسنده: باغبان
تاریخ: ۱۳۹۲/۰۵/۰۳ ۱۵:۳۹

ممنون از آموزش خوبتون می‌خواستم بپرسم در حالت دوم فقط یک رکورد از دیتابیس واکنشی میشه؟ یا اینکه از میون رکوردهای واکنشی شده یک رکورد را انتخاب می‌کنه؟

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۲/۰۵/۰۳ ۱۷:۸

در روش دوم فقط یک رکورد واکنشی می‌شود.