

بر اساس [رفتار پیش فرض](#) در دیتابیس SQL Server، در زمان انجام دادن یک دستور که منجر به ایجاد تغییرات در اطلاعات موجود در جدول می‌شود (برای مثال دستور Update)، جدول مربوطه به صورت کامل Lock می‌شود، ولو آن دستور Update، فقط با یکی از رکوردهای آن جدول کار داشته باشد.

در سیستم‌های با تعداد تراکنش بالا و دارای تعداد زیاد کلاینت، این رفتار پیش فرض موجب ایجاد صفی از تراکنش‌های در حال انتظار بر روی جداولی می‌شود که ویرایش‌های زیادی بر روی آنها رخ می‌دهد. اگر چه که بنظر این مشکل [راه حل‌های زیادی دارد](#)، لکن آن راه حلی که همیشه موثر عمل می‌کند استفاده از SQL Server Table Hints است.

SQL Server Table Hints به تمامی آن دستوراتی گفته می‌شود که هنگام اجرای دستور اصلی (برای مثال Select و یا Update) رفتار پیش فرض SQL Server را بر اساس Hint ارائه شده تغییر می‌دهند.

لیست کامل این Hint ها را می‌توانید در [اینجا مشاهده کنید](#).

این Hint ای که در اینجا برای ما مفید است، آن است که به SQL Server بگوییم هنگام اجرای دستور Update، به جای Lock کردن کل جدول، فقط رکورد در حال ویرایش را Lock کند، و این باعث می‌شود تا باقی تراکنش‌ها، که ای بسا با سایر رکوردهای آن جدول کار داشته باشند متوقف نشوند، که البته این مسئله کمی به افزایش مصرف حافظه می‌انجامد، لکن مقدار افزایش بسیار ناچیز است.

این Hint که rowlock نام دارد در تراکنش‌های با Isolation Level تنظیم شده بر روی Snapshot باید با یک Table Hint دیگر با نام updlock ترکیب شود.

توضیحات مفصل‌تر این دو Hint در لینک مربوطه آمده است.

بنابر این، بجای دستور

```
update products
set Name = "Test"
Where Id = 1
```

داریم

```
update products with (nolock,updlock)
set Name = "Test"
where Id = 1
```

تا اینجا مشکل خاصی وجود ندارد، آنچه که از اینجا به بعد اهمیت دارد این است که در هنگام کار با Entity Framework، اساسا ما نویسنده دستورات Update نیستیم که به آنها Hint اضافه کنیم یا نه، بلکه دستورات SQL بوسیله Entity Framework ایجاد می‌شوند.

در Entity Framework، مکانیزمی تعبیه شده است با نام Db Command Interceptor که به شما اجازه می‌دهد دستورات SQL ساخته شده را [Log کنید](#) و یا قبل از اجرا [تغییر دهید](#)، که برای اضافه نمودن Table Hint ها ما از این روش استفاده می‌کنیم، برای انجام این کار داریم: (توضیحات در ادامه)

```
public class UpdateRowLockHintDbCommandInterceptor : IDbCommandInterceptor
{
    public void NonQueryExecuting(DbCommand command, DbCommandInterceptionContext<Int32> interceptionContext)
    {
        if (command.CommandType != CommandType.Text) return; // (1)
        if (!(command is SqlCommand)) return; // (2)
        SqlCommand sqlCommand = (SqlCommand)command;
        String commandText = sqlCommand.CommandText;
        String updateCommandRegularExpression = "(update) ";
```

```

        Boolean isUpdateCommand = Regex.IsMatch(commandText, updateCommandRegularExpression,
        RegexOptions.IgnoreCase | RegexOptions.Multiline); // You may use better regular expression pattern
        here.
        if (isUpdateCommand)
        {
            Boolean isSnapshotIsolationTransaction = sqlCommand.Transaction != null &&
            sqlCommand.Transaction.IsolationLevel == IsolationLevel.Snapshot;
            String tableHintToAdd = isSnapshotIsolationTransaction ? " with (rowlock , updlock) set
            " : " with (rowlock) set ";
            commandText = Regex.Replace(commandText, "^(set) ", (match) =>
            {
                return tableHintToAdd;
            }, RegexOptions.IgnoreCase | RegexOptions.Multiline);
            command.CommandText = commandText;
        }
    }
}

```

این کد در قسمت (1) ابتدا تشخیص می‌دهد که آیا این یک Command دارای Command Text است یا خیر، برای مثال اگر فراخوانی یک Stored Procedure است، ما با آن کاری نداریم.

در قسمت دوم تشخیص می‌دهیم که آیا با SQL Server در حال تعامل هستیم، یا برای مثال با Oracle و .... که ما برای Table Hintها فقط با SQL Server کار داریم.

سپس باید تشخیص دهیم که آیا این یک دستور update است یا خیر؟ برای این منظور از Regular Expressionها استفاده کرده ایم، که خیلی به بحث آموزش این پست مربوط نیست، به صورت کلی از Regular Expressionها برای یافتن و بررسی و جایگزینی عبارات با قاعده در هنگام کار با رشته‌ها استفاده می‌شود.

ممکن است Regular Expression ای که شما می‌نویسید بسیار بهتر از این نمونه باشد، که در این صورت خوشحال می‌شوم در قسمت نظرات آنرا قرار دهید.

در نهایت با بررسی Transaction Isolation Level مربوطه که Snapshot است یا خیر، به درج یک یا هر دو Table Hint مربوطه اقدام می‌نماییم.