

第34期-客户端与服务端求值&跟踪与非跟踪查询

2019年6月17日 21:05

客户端与服务端求值

在下面的示例中，一个辅助方法用于标准化从 SQL Server 数据库中返回的博客的 URL。由于 SQL Server 提供程序不了解此方法的实现方式，因此不可以将其转换为 SQL。除了在客户端上是通过执行该方法来返回 URL，查询的其余部分都是在数据库中执行的。

```
var blogs = context.Blogs
    .OrderByDescending(blog => blog.Rating)
    .Select(blog => new
    {
        Id = blog.BlogId,
        Url = StandardizeUrl(blog.Url)
    })
    .ToList();

public static string StandardizeUrl(string url)
{
    url = url.ToLower();

    if (!url.StartsWith("http://"))
    {
        url = string.Concat("http://", url);
    }

    return url;
}
```

客户端求值性能问题

```
var blogs = context.Blogs
    .Where(blog => StandardizeUrl(blog.Url).Contains("dotnet"))
    .ToList();
```

默认情况下，当执行客户端求值时，EF Core 将记录警告，客户端求值引发异常。

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder
        .UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=EFQuerying;Trusted_Connection=True;")
        .ConfigureWarnings(warnings =>
            warnings.Throw(RelationalEventId.QueryClientEvaluationWarning));
}
```

跟踪与非跟踪查询

在 EF Core 中查询数据时，EF Core 内置的跟踪器将对查询返回的实体进行跟踪，当修改实体时，会将检测到的任何更改

都会在 SaveChanges() 期间永久保存到数据库。

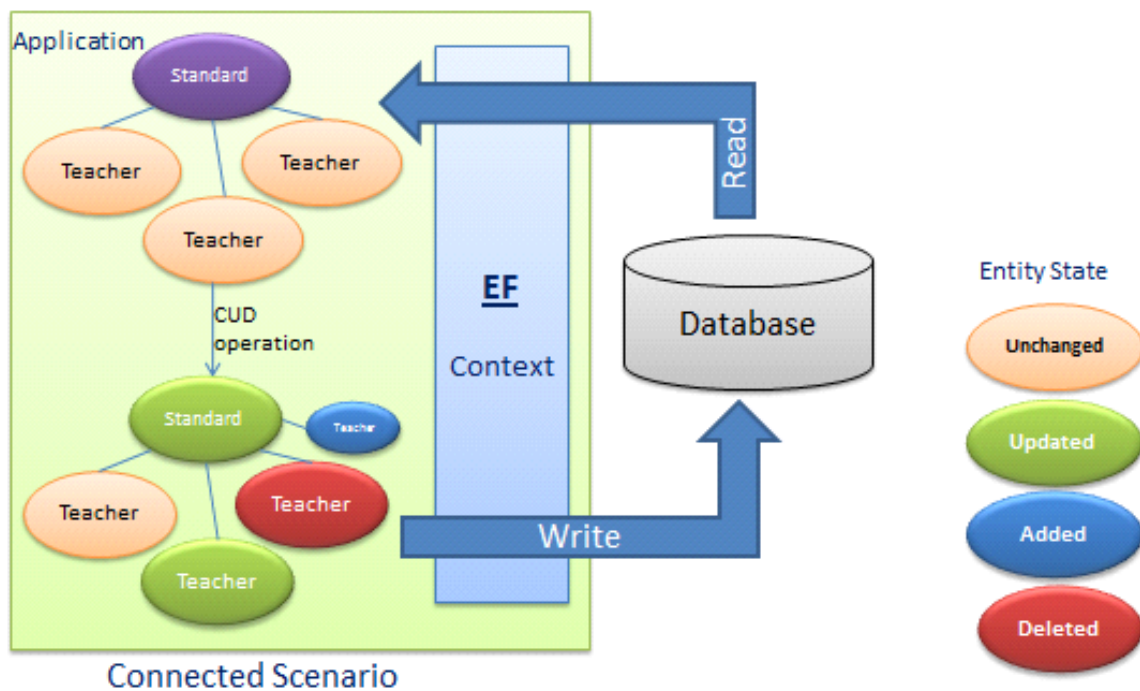
EF CORE 跟踪器对一个对象跟踪以下四种状态。

Added: 实体在数据库中不存在，调用 SaveChanges 方式时会把实体 Insert 插入到数据库。

unchanged: 实体存在于数据库中，但客户端查询出来后未进行过修改，当调用 SaveChanges 时将忽略。

updated: 实体存在于数据库中，查询出来后进行了修改，当调用 SaveChanges 方法时，将 Update 更新到数据库。

Deleted: 实体存在于数据库，单客户端调用了删除方法从内存中移除，用 SaveChanges 方法时，将从数据库中 Delete 移除。



跟踪查询

```
using (var context = new BloggingContext())
{
    var blog = context.Blogs.SingleOrDefault(b => b.BlogId == 1);
    blog.Rating = 5;
    context.SaveChanges();
}
```

虽然跟踪器非常方便，但当查询数据时，上下文会创建实体的快照，实体从而被跟踪比较变更。跟踪的目的是监控变更，当提交 SaveChanges 时，保存到数据库，但有时候，我们只查询数据，只读数据，不写入，不需要修改数据，这个时候，我们跟踪器就多余的了，但跟踪器会占用快照内存和跟踪本身的性能消耗，所以不跟踪最好，可以这样做。

```
using (var context = new BloggingContext())
{
    var blogs = context.Blogs
        .AsNoTracking()
        .ToList();
}
```

以上针对某次查询中不跟踪设置，还可全局设置查询不跟踪，这样就方便一次查询多张表，不用每个都 AsNoTracking

了。

```
using (var context = new BloggingContext())
{
    context.ChangeTracker.QueryTrackingBehavior = QueryTrackingBehavior.NoTracking;
    var blogs = context.Blogs.ToList();
}
```

跟踪与投影

即使查询的结果类型不是实体类型，但如果结果包含实体类型，则默认情况下也会跟踪这些实体类型。在以下返回匿名类型的查询中，会跟踪结果集中 Blog 的实例。

```
using (var context = new BloggingContext())
{
    var blog = context.Blogs
        .Select(b =>
            new
            {
                Blog = b,
                Posts = b.Posts.Count()
            });
}
```

如果结果集不包含任何实体类型，则不会执行跟踪。在以下返回匿名类型（具有实体中的某些值，但没有实际实体类型的实例）的查询中，不会执行跟踪。

```
using (var context = new BloggingContext())
{
    var blog = context.Blogs
        .Select(b =>
            new
            {
                Id = b.BlogId,
                Url = b.Url
            });
}
```