

第35期-执行原始SQL查询

2019年6月18日 10:59

在 EF Core 中可以使用原始 SQL 语言对数据进行查询，当无法使用 LINQ 表达要执行的查询或者因使用 LINQ 查询而导致低效时，SQL 查询非常有用，原始 SQL 查询可返回实体类型。

基本原生 SQL 查询

可以使用 FromSql 扩展方法开始查询。

```
var blogs = context.Blogs.FromSql("SELECT * FROM dbo.Blogs WHERE BlogId>1").ToList();
```

原生 SQL 查询可用于执行存储过程。

```
CREATE PROCEDURE GetMostPopularBlogs
    -- Add the parameters for the stored procedure here
    @Name VARCHAR(50)
AS
BEGIN
    -- Insert statements for procedure here
    SELECT * FROM Blogs WHERE [Name] = @Name
END
GO
```

```
var blogs = context.Blogs.FromSql("EXECUTE dbo.GetMostPopularBlogs") .ToList();
```

传递参数

原始 SQL 查询务必参数化参数，以抵御 SQL 注入攻击，可以将参数占位符包含在 SQL 查询语句中，EF CORE 会将提供的任何参数值将自动转换为 DbParameter 以防止 SQL 注入攻击。

SQL Injection

```
var user = "johndoe";

var blogs = context.Blogs
    .FromSql("EXECUTE dbo.GetMostPopularBlogsForUser {0}", user)
    .ToList();
```

上面的示例将一个参数传递到存储过程，尽管这看上去可能像 String.Format 语法，但提供的值包装在参数中，且生成的参数名称插入在指定 {0} 占位符的位置。

EF Core 2.0 及更高版本支持的字符串内插语法:

```
var user = "johndoe";

var blogs = context.Blogs
    .FromSql($"EXECUTE dbo.GetMostPopularBlogsForUser {user}")
    .ToList();
```

也可以构造 DbParameter 并将其作为参数值提供:

```
var user = new SqlParameter("user", "johndoe");

var blogs = context.Blogs
    .FromSql("EXECUTE dbo.GetMostPopularBlogsForUser @user", user)
    .ToList();
```

SQL 查询可使用具有命名的参数, 这在存储的流程具有可选参数时非常有用。

```
var user = new SqlParameter("user", "johndoe");

var blogs = context.Blogs
    .FromSql("EXECUTE dbo.GetMostPopularBlogs @filterByUser=@user", user)
    .ToList();
```

原始 SQL 查询与 LINQ 查询组合

```
var searchTerm = ".NET";

var blogs = context.Blogs
    .FromSql($"SELECT * FROM dbo.SearchBlogs({searchTerm})")
    .Where(b => b.Rating > 3)
    .OrderByDescending(b => b.Rating)
    .ToList();
```

原始 SQL 查询与查询跟踪器

当使用 FromSql 查询数据时, 遵循与 LINQ 查询完全相同的跟踪规则, 也就是默认情况下会跟踪查询到的对象状态, 参见上节课。如果需要不跟踪查询, 可使用 AsNoTracking 方式。

```
var searchTerm = ".NET";

var blogs = context.Query<SearchBlogsDto>()
    .FromSql($"SELECT * FROM dbo.SearchBlogs({searchTerm})")
    .AsNoTracking();
```

```
.ToList();
```

在原始 SQL 查询中关联导航数据

和 LINQ 查询一样，原始 SQL 查询也可使用 Include() 预先加载导航数据，当然也支持显式加载和延迟加载方式关联导航数据，因为原始查询到的对象受 EF Core 跟踪器的管理。

```
var searchTerm = ".NET";

var blogs = context.Blogs
    .FromSql($"SELECT * FROM dbo.SearchBlogs({searchTerm})")
    .Include(b => b.Posts)
    .ToList();
```

注意事项和一些限制

- SQL 查询必须返回实体的所有属性的字段。
- 结果集中的列名必须与属性映射到的列名称匹配。
- SQL 查询不能包含关联数据。
- 除 SELECT 以外的其它 SQL 语句无法运行。

FormattableString 类型支持内插法。