

در نگارش‌های پیشین EF امکان استفاده از Stored Procedure ها و یا Function های SQL ای به صورت Code First وجود نداشت. ولی در نگارش 6.1 آن با استفاده از کتابخانه‌ی [EntityFramework.CodeFirstStoreFunctions](#) می‌توان آنها را فراخوانی کرد.

```
protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
    modelBuilder.Conventions.Add(new FunctionsConvention<MyContext>("dbo"));
}
```

ابتدا لازم است که قاعده‌ی FunctionsConvention را در OnModelCreating به قواعد موجود modelBuilder اضافه کنیم. سپس به طریق زیر می‌توانیم Stored Procedure و Table Value Function را فراخوانی نمائیم:

```
[DbFunction("MyContext", "CustomersByZipCode")]
public IQueryable<Customer> CustomersByZipCode(string zipCode)
{
    var zipCodeParameter = zipCode != null ?
        new ObjectParameter("ZipCode", zipCode) :
        new ObjectParameter("ZipCode", typeof(string));

    return ((IObjectContextAdapter)this).ObjectContext
        .CreateQuery<Customer>(
            string.Format("[{0}].[1]", GetType().Name,
                "[CustomersByZipCode](@ZipCode)"), zipCodeParameter);
}

public ObjectResult<Customer> GetCustomersByName(string name)
{
    var nameParameter = name != null ?
        new ObjectParameter("Name", name) :
        new ObjectParameter("Name", typeof(string));

    return ((IObjectContextAdapter)this).ObjectContext
        .ExecuteFunction<Customer>("GetCustomersByName", nameParameter);
}
```

بدنه SP و TVF استفاده شده به شرح زیر است:

```
context.Database.ExecuteSqlCommand(
    "CREATE PROCEDURE [dbo].[GetCustomersByName] @Name nvarchar(max) AS " +
    "SELECT [Id], [Name], [ZipCode] " +
    "FROM [dbo].[Customers] " +
    "WHERE [Name] LIKE (@Name)");

context.Database.ExecuteSqlCommand(
    "CREATE FUNCTION [dbo].[CustomersByZipCode](@ZipCode nchar(5)) " +
    "RETURNS TABLE " +
    "RETURN " +
    "SELECT [Id], [Name], [ZipCode] " +
    "FROM [dbo].[Customers] " +
    "WHERE [ZipCode] = @ZipCode");
```

چند نکته:

نوع خروجی تابع باید از `IQueryable<T>` باشد. T باید از جنسی باشد که معادل EDM آن موجود باشد. T می‌تواند از انواع اصلی (Primitive) باشد که در EF پشتیبانی می‌شوند (رجوع شود به [Entity Data Model: Primitive Data Types](#)) به طور مثال، `int` قابل استفاده است ولی `uint` خیر و یا می‌تواند از انواع غیر اصلی (non-primitive) باشد (enum/complex type/entity type) که در مدل شما تعریف شده‌اند. پارامترهای متد باید از نوع اسکالر (primitive یا enum) باشند که قابل map شدن به نوع‌های EF باشند.

نام متد، `DbFunction.FunctionName` و `querysting` ای که به `CreateQuery` پاس داده می‌شود باید همگی یکسان باشند. توضیحات بیشتر در [support for SPs TVFs in entityframework 6.1](#) کد پروژه در [CodePlex](#)