< Previous

Next >

Execute Raw SQL Queries in Entity Framework 6

Entity Framework allows you to execute raw SQL queries for the underlying relational database.

The following methods can be used to execute raw SQL queries to the database using Entity Framework 6.x:

- DbSet.SqlQuery()
- DbContext.Database.SqlQuery()
- DbContext.Database.ExecuteSqlCommand()

DbSet.SqlQuery()

Use the `DbSet.SqlQuery()` method to write raw SQL queries which return entity instances. The resulted entities will be tracked by the context, as if they were returned by the LINQ query.

```
using (var ctx = new SchoolDBEntities())
{
    var studentList = ctx.Students
                        .SqlQuery("Select * from Students")
                        .ToList<Student>();
}
```

The above query executes `Select * from Students` SQL in the database to get all students and will be converted into a list of `Student` entities. The column names in the SQL query must match with the properties of an entity type, otherwise, it will throw an exception.

You can specify the parameters using the object of `SqlParameter`, as shown below.

```
using (var ctx = new SchoolDBEntities())
{
    var student = ctx.Students
                    .SqlQuery("Select * from Students where StudentId=@id", new SqlParameter("@id", 1))
                    .FirstOrDefault();
}
```

If you change the column name in SQL query, then it will throw an exception because it must match column names. The following example will throw an exception.

```
using (var ctx = new SchoolDBEntities())
{
    //this will throw an exception
    var studentName = ctx.Students.SqlQuery("Select studentid as id, studentname as name
            from Student where studentname='Steve'").ToList();
}
```

The `DbSet<TEntity>.SqlQuery()` executes the SQL query only for the table which is mapped with the specified entity (e.g. `DbSet<Student>.SqlQuery()` only returns the result from the corresponding `Students` table and not from any other table). The following will throw an exception.

```
using (var ctx = new SchoolDBEntities())
{
    //this will throw an exception
    var studentName = ctx.Students.SqlQuery("Select * from Courses").ToList();
}
```

Database.SqlQuery()

The `Database` class represents the underlying database and provides various methods to deal with the database. The `Database.SqlQuery()` method returns a value of any type.

```
using (var ctx = new SchoolDBEntities())
{
    //Get student name of string type
    string studentName = ctx.Database.SqlQuery<string>("Select studentname from Student where studentid=1")
                            .FirstOrDefault();

    //or
    string studentName = ctx.Database.SqlQuery<string>("Select studentname from Student where studentid=@id", new
SqlParameter("@id", 1))
                            .FirstOrDefault();
}
```

Database.ExecuteSqlCommand()

The `Database.ExecuteSqlCommnad()` method is useful in executing database commands, such as the Insert, Update and Delete command.

```
using (var ctx = new SchoolDBEntities())
{
    int noOfRowUpdated = ctx.Database.ExecuteSqlCommand("Update student
            set studentname ='changed student by command' where studentid=1");

    int noOfRowInserted = ctx.Database.ExecuteSqlCommand("insert into student(studentname)
            values('New Student')");

    int noOfRowDeleted = ctx.Database.ExecuteSqlCommand("delete from student
            where studentid=1");
}
```

[Download EF 6 DB-First Demo Project from Github](#)

< Previous

Next >

ENTITYFRAMEWORKTUTORIAL

Learn Entity Framework using simple yet practical examples on EntityFrameworkTutorial.net for free. Learn Entity Framework DB-First, Code-First and EF Core step by step. While using this site, you agree to have read and accepted our terms of use and privacy policy.

✉ feedback@entityframeworktutorial.net

TUTORIALS

- EF Basics
- EF Core
- EF 6 DB-First
- EF 6 Code-First

E-MAIL LIST

Subscribe to EntityFrameworkTutorial email list and get EF 6 and EF Core Cheat Sheets, latest updates, tips & tricks about Entity Framework to your inbox.

Email address GO

We respect your privacy.

HOME PRIVACY POLICY ADVERTISE WITH US

© 2020 EntityFrameworkTutorial.net. All Rights Reserved.