


Improve Entity Framework Performance



 Bulk Insert

 Bulk Delete

 Bulk Update

 Bulk Merge

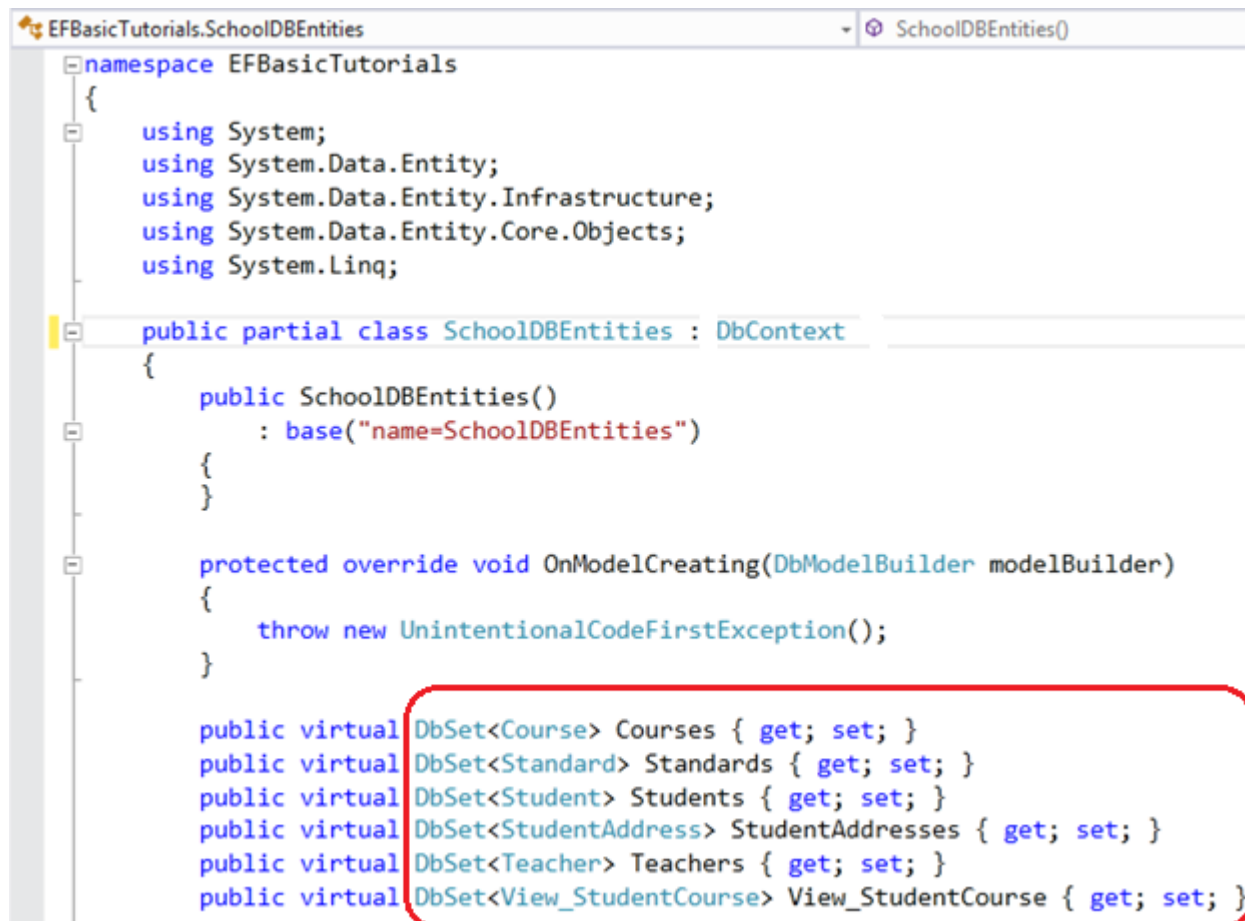
[LEARN MORE](#)

[< Previous](#)[Next >](#)

DbSet in Entity Framework 6

The `DbSet` class represents an entity set that can be used for create, read, update, and delete operations.

The context class (derived from `DbContext`) must include the `DbSet` type properties for the entities which map to database tables and views.



```

namespace EFBasicTutorials
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;
    using System.Data.Entity.Core.Objects;
    using System.Linq;

    public partial class SchoolDBEntities : DbContext
    {
        public SchoolDBEntities()
            : base("name=SchoolDBEntities")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Course> Courses { get; set; }
        public virtual DbSet<Standard> Standards { get; set; }
        public virtual DbSet<Student> Students { get; set; }
        public virtual DbSet<StudentAddress> StudentAddresses { get; set; }
        public virtual DbSet<Teacher> Teachers { get; set; }
        public virtual DbSet<View_StudentCourse> View_StudentCourse { get; set; }
    }
}

```

The following table lists important methods of the `DbSet` class:

Add	Added entity type	<p>Adds the given entity to the context with the Added state. When the changes are saved, the entities in the Added states are inserted into the database. After the changes are saved, the object state changes to Unchanged.</p> <p>Example: <code>dbcontext.Students.Add(studentEntity)</code></p>

AsNoTracking<Entity>	DBQuery<Entity>	<p>Returns a new query where the entities returned will not be cached in the DbContext. (Inherited from DbQuery.)</p> <p>Entities returned as AsNoTracking will not be tracked by DbContext. This will be a significant performance boost for read-only entities.</p> <p>Example: var studentList = dbContext.Students.AsNoTracking<Student>().ToList<Student>();</p>
Attach(Entity)	Entity which was passed as parameter	<p>Attaches the given entity to the context in the Unchanged state</p> <p>Example: dbContext.Students.Attach(studentEntity);</p>
Create	Entity	<p>Creates a new instance of an entity for the type of this set. This instance is not added or attached to the set. The instance returned will be a proxy if the underlying context is configured to create proxies and the entity type meets the requirements for creating a proxy.</p> <p>Example: var newStudentEntity = dbContext.Students.Create();</p>
Find(int)	Entity type	<p>Uses the primary key value to find an entity tracked by the context. If the entity is not in the context, then a query will be executed and evaluated against the data in the data source, and null is returned if the entity is not found in the context or in the data source. Note that the Find also returns entities that have been added to the context but have not yet been saved to the database.</p> <p>Example: Student studEntity = dbContext.Students.Find(1);</p>

Include	DBQuery	<p>Returns the included non-generic LINQ to Entities query against a DbContext. (Inherited from DbQuery)</p> <p>Example:</p> <pre>var studentList = dbcontext.Students.Include("StudentAddress").ToList<Student>(); var studentList = dbcontext.Students.Include(s => s.StudentAddress).ToList<Student>();</pre>
Remove	Removed entity	<p>Marks the given entity as Deleted. When the changes are saved, the entity is deleted from the database. The entity must exist in the context in some other state before this method is called.</p> <p>Example:</p> <pre>dbcontext.Students.Remove(studentEntity);</pre>
SqlQuery	DBSqlQuery	<p>Creates a raw SQL query that will return entities in this set. By default, the entities returned are tracked by the context; this can be changed by calling AsNoTracking on the DbSqlQuery<TEntity> returned from this method.</p> <p>Example:</p> <pre>var studentEntity = dbcontext.Students.SqlQuery("select * from student where studentid = 1").FirstOrDefault<Student>();</pre>

Visit MSND for more information on [DbSet](#).

[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.

We respect your privacy.

[HOME](#) [PRIVACY POLICY](#) [ADVERTISE WITH US](#)

© 2020 EntityFrameworkTutorial.net. All Rights Reserved.