

Заголовок украли цыгане

Интеграция контекстов и LINQ-запросов в Entity Framework Core

Интеграция 2 контекстов

```
using UltimateProject.Data;

using UltimateProject.Models;

using Microsoft.EntityFrameworkCore;

using System.IO;

namespace UltimateProject
{
    internal class Program
    {
        static void Main()
        {
            // 🌀 СОЗДАЕМ ДВА НЕЗАВИСИМЫХ КОНТЕКСТА

            using var genericContext = new GenericContext();

            using var alterContext = new AlterContext();

            // 🗑 ОЧИСТКА И СОЗДАНИЕ БАЗ
```

```
genericContext.Database.EnsureDeleted();

genericContext.Database.EnsureCreated();

alterContext.Database.EnsureDeleted();

alterContext.Database.EnsureCreated();


// 📁 ИНФОРМАЦИЯ О БАЗАХ

Console.WriteLine($"Generic база:
{Path.GetFullPath("generic_database.db")}");

Console.WriteLine($"Alter база:
{Path.GetFullPath("alter_database.db")}");


// 📁 СОЗДАНИЕ ДАННЫХ В GENERIC

var genericEntities = new List<MainEntity>

{

    new() { Name = "Generic Конференция", Description = "IT
события", Location = "Москва" },

    new() { Name = "Generic Фестиваль", Description = "Живая
музыка", Location = "СПб" }

};


genericContext.MainEntities.AddRange(genericEntities);

genericContext.SaveChanges();


// 📁 СОЗДАНИЕ ДАННЫХ В ALTER

var alterEntities = new List<MainEntity>
```

```

    {
        new() { Name = "Alter Симпозиум", Description = "Научные
исследования", Location = "Новосибирск" },

        new() { Name = "Alter Выставка", Description =
"Искусство", Location = "Казань" }

    };

    alterContext.MainEntities.AddRange(alterEntities);

    alterContext.SaveChanges();

    // 🔥 LINQ ЗАПРОСЫ ДЛЯ КАЖДОГО КОНТЕКСТА

    Console.WriteLine("\n≡ GENERIC КОНТЕКСТ ≡");

    var genericAll = genericContext.MainEntities.ToList();

    genericAll.ForEach(g => Console.WriteLine($"    {g.Name} -
{g.Location}"));

    Console.WriteLine("\n≡ ALTER КОНТЕКСТ ≡");

    var alterAll = alterContext.MainEntities.ToList();

    alterAll.ForEach(a => Console.WriteLine($"    {a.Name} -
{a.Location}"));

    Console.WriteLine("\n≡ ОБЪЕДИНЕННЫЕ РЕЗУЛЬТАТЫ ≡");

    var allEntities = genericAll.Concat(alterAll)

```

```
        .OrderBy(e => e.Name)

        .ToList();

        allEntities.ForEach(e => Console.WriteLine($" {e.Name}
({e.Location})"));
    }

}

}
```



Интеграция 1 контекста

```
using UltimateProject.Data;

using UltimateProject.Models;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.IO;

namespace UltimateProject
{
    internal class Program
    {

```

```
static void Main()

{

    using var context = new GenericContext();

    // 🗑 Очистка и создание БД

    context.Database.EnsureDeleted();

    context.Database.EnsureCreated();

    var connection = context.Database.GetDbConnection();

    if (connection.DataSource != null)

        Console.WriteLine($"База данных:
{Path.GetFullPath(connection.DataSource)}");


    // 📄 СОЗДАНИЕ ДАННЫХ

    var mainEntities = new List<MainEntity>

    {

        new() { Name = "Технологическая Конференция",
Description = "IT события", Location = "Москва" },

        new() { Name = "Музыкальный Фестиваль", Description =
"Живая музыка", Location = "Санкт-Петербург" },

        new() { Name = "Научный Симпозиум", Description =
"Исследования", Location = "Новосибирск" }

    };


    var relatedEntities = new List<RelatedEntity>

    {

        new() { Name = "Анна Иванова", Email = "anna@test.com"
```

```
},

        new() { Name = "Борис Петров", Email = "boris@test.com"
    },

        new() { Name = "Светлана Сидорова", Email =
"svetlana@test.com" },

        new() { Name = "Дмитрий Козлов", Email =
"dmitry@test.com" }

    };

    context.MainEntities.AddRange(mainEntities);

    context.RelatedEntities.AddRange(relatedEntities);

    context.SaveChanges();

    // 📎 СОЗДАНИЕ СВЯЗЕЙ

    mainEntities[0].RelatedEntities.Add(relatedEntities[0]);
    mainEntities[0].RelatedEntities.Add(relatedEntities[1]);
    mainEntities[0].RelatedEntities.Add(relatedEntities[2]);
    mainEntities[1].RelatedEntities.Add(relatedEntities[1]);
    mainEntities[1].RelatedEntities.Add(relatedEntities[3]);
    mainEntities[2].RelatedEntities.Add(relatedEntities[0]);
    mainEntities[2].RelatedEntities.Add(relatedEntities[2]);
    mainEntities[2].RelatedEntities.Add(relatedEntities[3]);

    context.SaveChanges();
```

```
// 🔥 LINQ ЗАПРОСЫ  
  
...  
  
}  
  
}  
  
}
```



⚡ ЭКСПРЕСС-КОД

Создание БД + данные

```
using var context = new AppContext();  
  
context.Database.EnsureDeleted();  
  
context.Database.EnsureCreated();  
  
var mains = new List<MainEntity>  
{  
  
    new() { Name = "Объект 1", Description = "Описание 1" },  
  
    new() { Name = "Объект 2", Description = "Описание 2" }  
  
};  
  
var relateds = new List<RelatedEntity>  
{
```

```
new() { Name = "Связанный 1", Email = "test1@mail.com" },  
new() { Name = "Связанный 2", Email = "test2@mail.com" }  
};  
  
context.MainEntities.AddRange(mains);  
  
context.RelatedEntities.AddRange(relateds);  
  
context.SaveChanges();
```

Связи

```
mains[0].RelatedEntities.Add(relateds[0]);  
mains[0].RelatedEntities.Add(relateds[1]);  
mains[1].RelatedEntities.Add(relateds[0]);  
  
context.SaveChanges();
```

Запросы

```
context.MainEntities.ToList().ForEach(x => Console.WriteLine(x.Name));  
  
context.MainEntities  
    .Include(m => m.RelatedEntities)  
    .ToList()  
    .ForEach(m => Console.WriteLine($"{m.Name}:  
{m.RelatedEntities.Count}"));
```



```
context.MainEntities
    .GroupBy(m => m.Name)
    .Select(g => new { Name = g.Key, Count = g.Count() })
    .ToList()
    .ForEach(x => Console.WriteLine($"{x.Name}: {x.Count}"));
```

```
context.MainEntities
    .Where(m => m.RelatedEntities.Count > 1)
    .ToList()
    .ForEach(m => Console.WriteLine(m.Name));
```



Лinq запросы

◆ Базовые

```
var all = context.MainEntities.ToList();

var byId = context.MainEntities.Find(1);

var firstN = context.MainEntities.Take(5).ToList();

var sorted = context.MainEntities.OrderBy(m => m.Name).ToList();
```

◆ Связи

```
var withRelations = context.MainEntities.Include(m =>
m.RelatedEntities).ToList();

var withCounts = context.MainEntities

    .Select(m => new { m.Name, Count = m.RelatedEntities.Count })

    .ToList();

var withManyRelations = context.MainEntities

    .Where(m => m.RelatedEntities.Count >= 2)

    .ToList();
```

◆ Группировки

```
var grouped = context.MainEntities

    .GroupBy(m => m.Description)

    .Select(g => new { Key = g.Key, Count = g.Count() })

    .ToList();

var complexGroup = context.MainEntities

    .GroupBy(m => m.Description)

    .Select(g => new {

        Description = g.Key,

        Total = g.Count(),
```

```
AvgRelations = g.Average(m => m.RelatedEntities.Count)

}))

.ToList();
```



Ошибки(распространенные)

Database is locked

→ Отключись от базы (DBeaver/Rider/DataGrip) и перезапусти код

No such table

→ Проверь имена таблиц в `OnModelCreating`

Lazy Loading не работает

→ Добавь `virtual` к навигационным свойствам и `UseLazyLoadingProxies()`



Полезные методы (ну или не очень полезные кому как)

```
// Получение низкоуровневого соединения с базой данных
var connection = context.Database.GetDbConnection();

// Вывод информации о сервере базы данных (например: localhost, имя
// файла SQLite и т.д.)
Console.WriteLine($"База: {connection.DataSource}");

// Получение метаданных всех сущностей (таблиц), зарегистрированных в
// контексте
var tables = context.Model.GetEntityTypes();

// Перебор всех таблиц и вывод их имен
foreach (var table in tables)
```

```
Console.WriteLine(table.GetTableName());
```

```
// Создание LINQ-запроса (пока только определение, без выполнения)  
var query = context.MainEntities.Where(m => m.Id == 1);
```

```
// Получение SQL-кода, который будет выполнен для этого запроса  
Console.WriteLine(query.ToQueryString());
```



Алгоритм

1. Создай модели и контекст
2. Заполни данными и создай связи
3. Сделай 3–5 LINQ-запросов
4. Проверь и выведи результаты



Схемы баз данных

Фильмы и актёры

<https://ibb.co/Jj8Rz8M0> 

Error parsing Mermaid diagram!

Cannot read properties of null (reading 'getBBox')

События и участники

<https://ibb.co/kVYmS2sb> 

Error parsing Mermaid diagram!

Cannot read properties of null (reading 'getBBox')



ПАКЕТЫ



SQLite

- ◆ Microsoft.EntityFrameworkCore.Sqlite

MS SQL

- ◆ Microsoft.EntityFrameworkCore.SqlServer

PostgreSQL

- ◆ Npgsql.EntityFrameworkCore.PostgreSQL

MySQL

- ◆ Pomelo.EntityFrameworkCore.MySql

Общие

- ◆ Microsoft.EntityFrameworkCore
- ◆ Microsoft.EntityFrameworkCore.Abstractions
- ◆ Microsoft.EntityFrameworkCore.Analyzers
- ◆ Microsoft.EntityFrameworkCore.Design
- ◆ Microsoft.EntityFrameworkCore.InMemory
- ◆ Microsoft.EntityFrameworkCore.Proxies