Обратный инжиниринг и Миграция

*В память Погребному Ивану <3*

Обратный инжиниринг – получение классов, на основе существующей БД. Применяется при подходе к DataBaseFirst.

Миграция – это внесение изменений в БД, для ее синхронизации с ее моделью данных в приложении. Применяется при подходе CodeFirst.

Для работы нужен пакет Microsoft.EntityFrameworkCore.Tools

Scaffofd-DbContext – для обратного инжиниринга.

Команды можно писать в консоли диспетчера пакетов или в PowerShell (ну или в cmd).

Scaffold-DbContext “строка подключения” ПровайдерEfCore

-Context ИмяКонтекстаБД

-ContextDir ДиректорияКонтекстаБД

-OutputDir ДиректорияМоделейДанных

-Tables таблица1, таблица2

-Namespace ИмяПространстваМоделейДанных

-ContextNamespace ИмяПространстваКонтекстаБД

Scaffold-DbContext "Data Source = mssql; Initial Catalog = ispp2111; User ID = ispp2111; Password = 2111; Trust Server Certificate=True" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -ContextDir Data -Context SessionContext -Tables Customer, Ticket, Genre, Film

-Force используется для перезаписи существующих файлов (если указать меньше таблиц, нежели в исходном, то они не удаляются)

-UseDatabaseNames использовать названия таблиц и столбцов как названия классов и свойств

По умолчанию классы моделей генерируются с модификатором partial, поэтому можно расширить их функциональность в отдельных файлах

Метод OnModelCreating создается в контексте. Он с помощью Fluent API позволяет описать схему БД и указать начальные данные

Для добавления данных используется HasData (OnModelCreating)

modelBuilder.Entity<Category>().HasData(

new Category {CategoryId = 1, Name = “RPG”}

new Category {CategoryId = 2, Name = “notRPG”}

);

Модель можно настроить используя нотации или Fluent API

Если составной первичный ключ, то пишется в [PrimaryKey]

Если один, то [Key]

[Column(TypeName = “varchar(10)”)]

[Required] – то же что и null! (обязательно)

[MaxLength(50)]

[StringLength(50, MinimumLength = 2)]

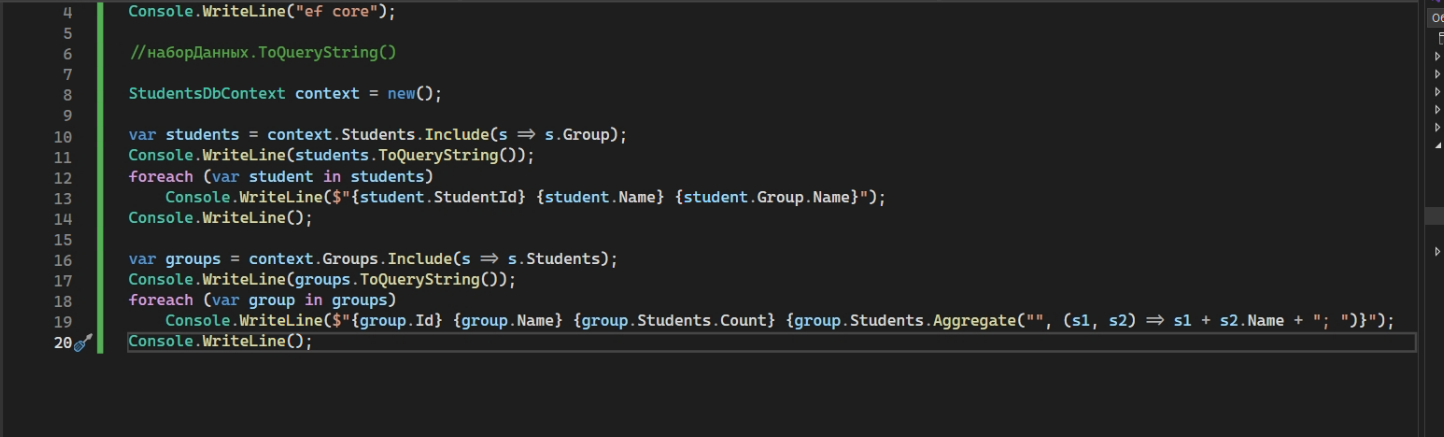
[MinLength(2)]

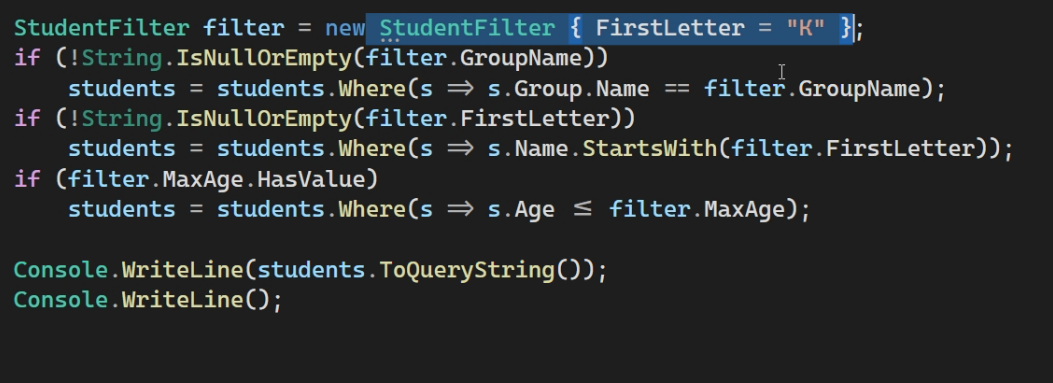
[Range(10, 100)]

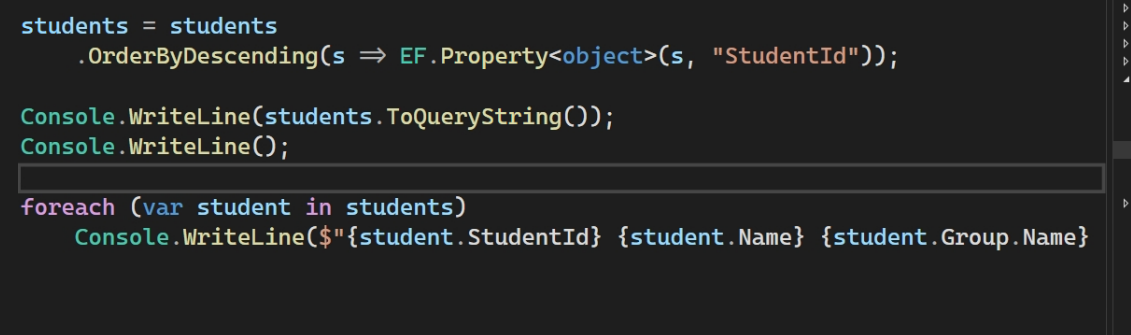
[ForeignKey(“Group”)]

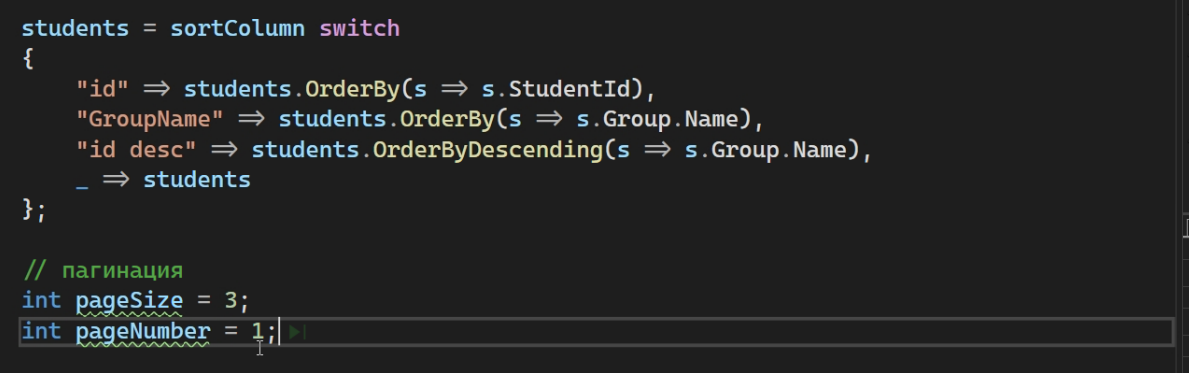
Получение и модификация данных

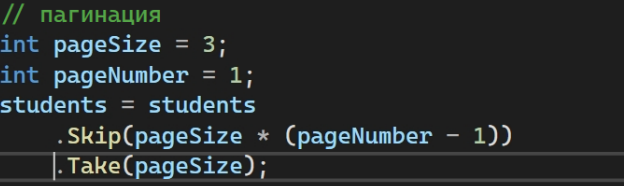
наборДанных.ToQueryString()







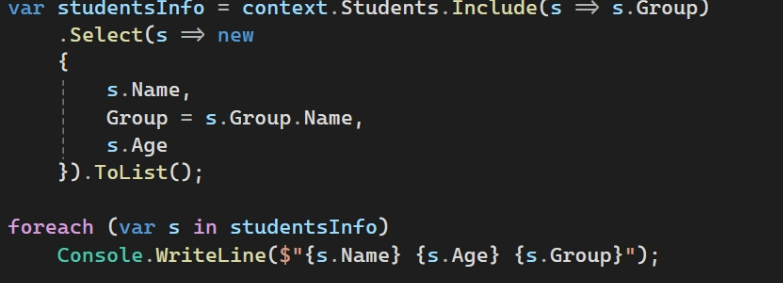




SELECT and ИНТРОВЕРЦИЯ

Получить часть данных, или данные удобные для пользователя

Для удобства можно создать классы DTO ( Data Transfer Object)



DTO для строк:

