**Обратный инжиниринг и миграция**

**Обратный инжиниринг** – это получение классов на основе существующей БД, применяется при подходе DatabaseFirst

**Миграция** – внесение изменений в БД для ее синхронизации с моделью данных приложения, применяется при подходе CodeFirst

Microsoft.EntityFrameworkCore.Tools + провайдер данных EF Core

Scaffold-DbContext

Add-Migration

Scaffold-DbContext “строка подключения” ПровайдерEfCore – строка с минимальными настройками

Scaffold-DbContext "Data Source = mssql; Initial Catalog = ispp2101; User ID = ispp2101; Password = 2101; Trust Server Certificate=True" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -ContextDir Data -Context GameStoreContext -Tables Game, Category

Доп. Параметры:

-Context ИмяКонтекстаБД

-ContextDir ДиректорияКонтекстаБД

-OutputDir ДиректорияМоделейДанных

-Tables таблица1, таблица2, …

-Namespace ИмяПространстваКонтекстаБД

-Force Для перезаписи существующих классов моделей

-UseDatadaseNames Использовать названия таблиц и столбцов как названия классов и столбцов

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

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

modelBuilder.Entity<Category>().HasData(

new Category { CategoryId = 1, Name = "RPG" },

new Category { CategoryId = 2, Name = "notRPG" }

);

Модель можно настроить используя аннотации или Fluent API

Update-Database применяет все миграции