Чтобы подключаться к базе данных, нам надо будет добавить в проект классы моделей, которые соответствуют определениям таблиц, и класс контекста данных, который соответствует БД.

Увы – у нас нет визуального средства маппинга БД на классы.

Для решения этих проблем в Entity Framework Core предусмотрена функция Reverse Engineering, которая позволяет автоматически создать все необходимые классы по базе данных.

Scaffold-DbContext "Server=(localdb)\MSSQLLocalDB;Database=TimeTable\_Data;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir ../DBO/Models

По умолчанию в полученной модели для описания ключей, ограничений , Внешних ключей и навигационных свойств используется Fluent API

Можно использовать также и аттрибуты.

**Файл конфигурации**

Оба выше представленных способа вполне работают, однако в том определении, в котором они представлены, они имеют один недостаток - строка подключения жестко определена в коде C#. И было бы неплохо, если бы она была бы определена в каком-нибудь внешнем файле подключения, где мы ее могли бы поменять без перекомпиляции приложения.

Добавим в проект новый элемент JSON File, который назовем appsettings.json.