**Database Configurations**  
Code first olaraq işlənən proyekt daxilində C# kodları və EntityFramework vasitəsilə Database daxilində müəyyən table’lar yaranır. Bu table daxilindəki sütunlarda(column) yer alacaq datalara(row) müəyyən kodlar vasitəsi ilə konfiqurasiyalar verə bilərik.

**Data Annotation**

Konfiqurasiyalar üçün ən sadə yol Data Annotation istifadəsidir.   
Aşağıdaki Annotation’lar ən çox rastlaşa biləcəklərimizdən sadəcə bəziləridir.

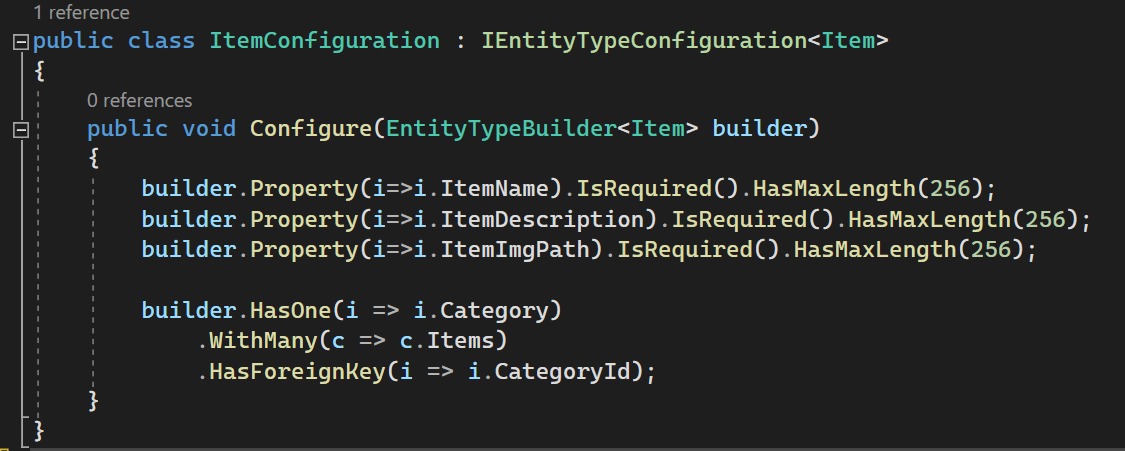
[Required(ErrorMessage=customErrorMessage)]

[MaxLength(maxLength,ErrorMessage=customErrorMessage)]

[RegularExpression(regularExpression,ErrorMessage=customErrorMessage)

**Fluent API(Separate Class)**

Daha professional yol isə Fluent API’dır. Fluent API, EntityFrameworkCore package ilə gələn IEntityTypeConfiguration<TEntity> interface’dən implement edən class daxilində hər bir entity üçün yazılır. Class’ların adlandırılması uyğun olaraq Entity adı ilə başlayıb Configuration sonluğu ilə bitir. Bu class daxilində interface’ə generic olaraq göndərilən entity’in property’ləri üçün müəyyən məhdudiyyətlər yarada bilərik. Bununla yanaşı biz Fluent API vasitəsi ilə Relationship’ləri qura bilərik. Hər bir configuration əməliyyatı implement edilən Configure method daxilində yazılır.

Aşağıdaki şəkildə database table’ın column’ları üçün bir sıra məhdudiyyət configurasiyaları və Category ilə Many to One əlaqəsi görülür.  
  


**DbContext Configuration**  
Configration class’larımızı Database tərəfi üçün işlək hala gətirmək üçün custom olaraq yaratdığımız və DbContext’dən miras alan class daxilində OnModelCreating(ModelBuilder modelBuilder); method override olunmalıdır. Method daxilində ApplyConfigurationsFromAssembly(Assembly assembly); method’u call olunmalı və uyğun Configuration class’larından birinin yer aldığı Assembly,

type of(CustomConfigurationClass).Assembly vasitəsi ilə yuxarıda qeyd olunan method’a argument şəkilində göndərilməlidir. Əgər CustomDbContext class ilə Configuration class’larımız eyni Assembly’də yerləşərsə bu zaman Assembly.GetExecutingAssembly() vasitəsi ilə əldə etdiyimiz Assembly obyektini də method’a argument olaraq göndərə bilərik.

Fluent API həmçinin OnModelCreating method’u daxilində də yazıla bilər. Lakin method’un daxili lazımsız yerə şişirilməsin deyə ayrı class’lar şəklində çıxarılır. Əgər method daxilində yazılsaydı aşağıdakı şəkildə görünəcəkdi.

