



Basics

```
public class PlutoContext : DbContext
{
    protected override void OnModelCreating(DbModelBuilder
modelBuilder)
    {
        modelBuilder
            .Entity<Course>
            .Property(t => t.Name)
            .IsRequired();
    }
}
```

Tables

```
Entity<Course>.ToTable( "tbl_Course", "catalog");
```

Primary Keys

```
Entity<Book>.HasKey( t => t.Isbn);
Entity<Book>.Property(t => t.Isbn)
    .HasDatabaseGeneratedOption(DatabaseGeneratedOption.None);
```

Composite Primary Keys

```
Entity<OrderItem>.HasKey(t => new { t.OrderId, t.OrderId });
```

Columns

```
Entity<Course>.Property(t => t.Name)
    .HasColumnName("sName")
    .HasColumnType("varchar")
    .HasColumnOrder(2)
    .IsRequired()
    .HasMaxLength(255);
```

One-to-many Relationship

```
Entity<Author>
    .HasMany(a => a.Courses)
    .WithRequired(c => c.Author)
    .HasForeignKey(c => c.AuthorId);
```

Many-to-many Relationship

```
Entity<Course>
    .HasMany(c => c.Tags)
    .WithMany(t => t.Courses)
    .Map(m =>
    {
        m.ToTable("CourseTag");
        m.MapLeftKey("CourseId");
        m.MapRightKey("TagId");
    });
```

One-to-zero/one Relationship

```
Entity<Course>  
    .HasOptional(c => c.Caption)  
    .WithRequired(c => c.Course);
```

One-to-one Relationship

```
Entity<Course>  
    .HasRequired(c => c.Cover)  
    .WithRequiredPrincipal(c => c.Course);
```

```
Entity<Cover>  
    .HasRequired(c => c.Course)  
    .WithRequiredDependent(c => c.Cover);
```