

# E /

## Entity FrameWork Core MEO

association nationale  
pour la formation professionnelle  
des adultes

Vincent BOST



**Heritage TPH**

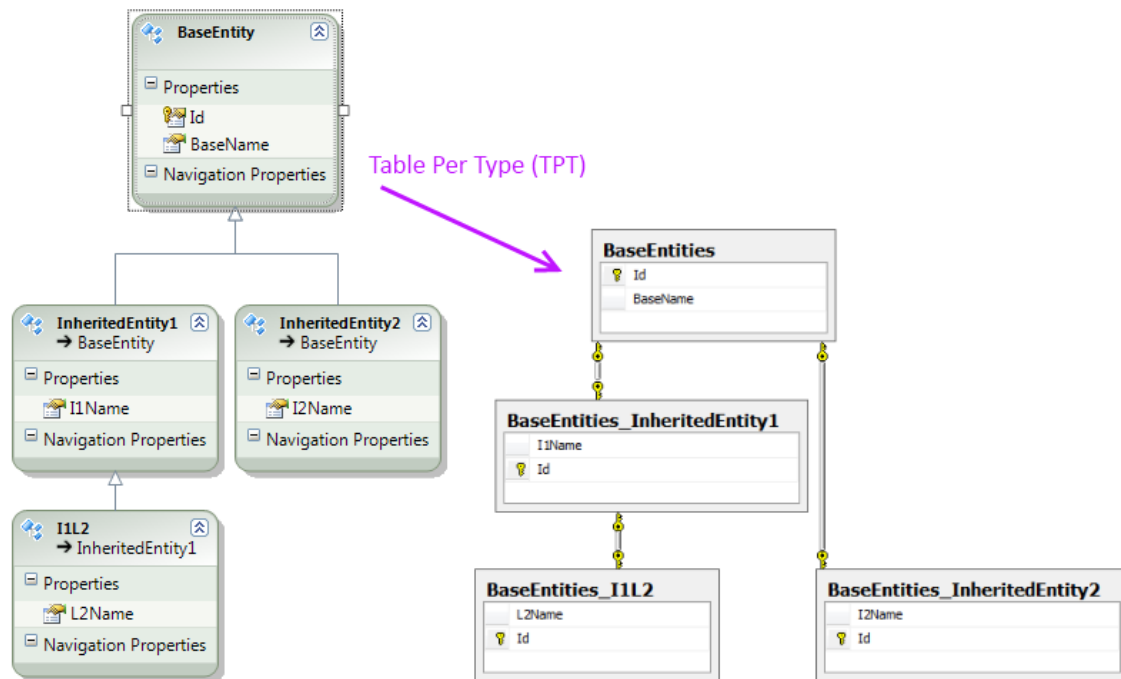
**Amender les modèles**

**Amender le contexte de base de données**

**Assurer un suivi des modifications en mode  
déconnecté / UOW**

## 3 modèles

- Le modèle TPT : Table Per Type
- Le modèle TPH : Table Per Hierarchy
- Le modèle TPC : Table Per Concret type



## Avantages

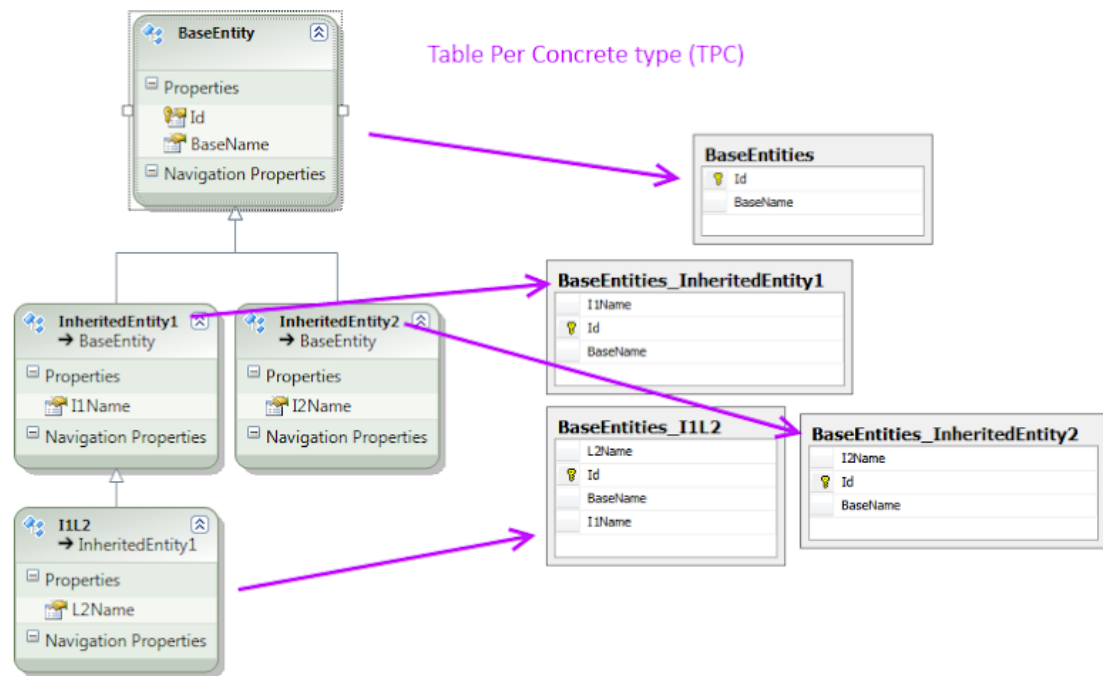
Rigueur

Respect strict des principes relationnels

## Inconvénients

Manque d'efficacité

Temps de traitements



## Avantages

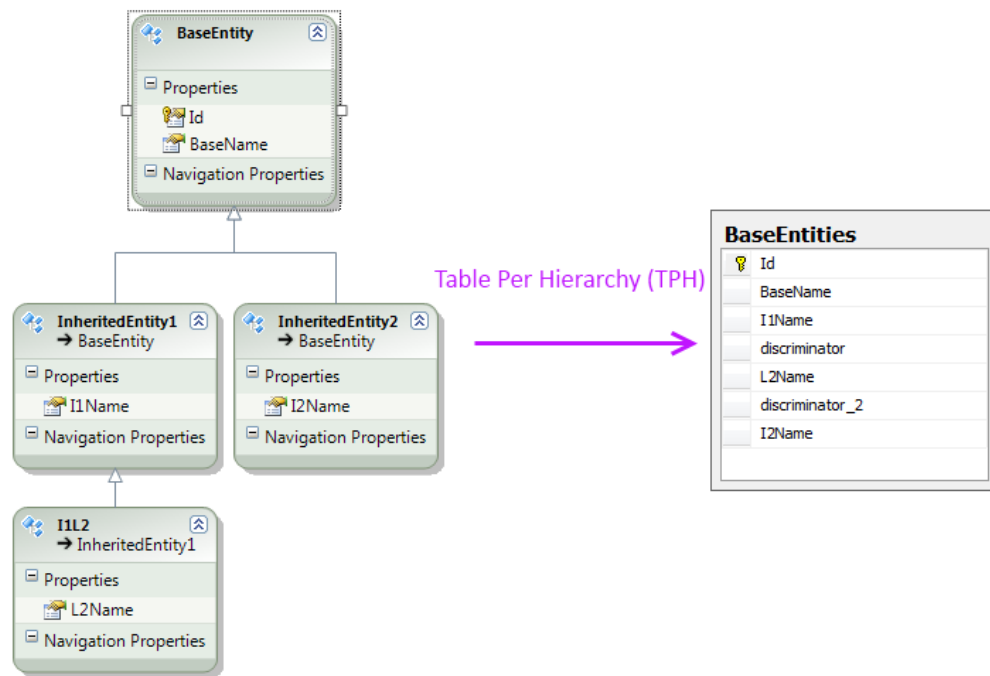
Efficiency

Tables indépendantes

## Inconvénients

Complexité

Redondance



## Avantages

Seul implémenté EF  
Core

Efficiency

Simplicité

## Inconvénients

Eloigne règles  
relationnel

Table large

Attributs nuls

## Dans le contexte de l'atelier

## 4 classes

Personne	
🔑	IdPersonne
	NomPersonne
	PrenomPersonne
	CivilitePersonne
	SexePersonne
	AdresseMail
	CatPersonne
	MatriculeCollaborateurAfpa
	MatriculeStagiaire
	DateNaissanceStagiaire

- Base abstract Personne
- Tiers : Personne
- CollaborateurAfpa : Personne
  - Matricule collaborateur
- Stagiaire : Personne
  - Matricule stagiaire
  - DateNaissanceStagiaire

## A isoler dans un code source sous forme de classes partielles

- **Créer les entités spécialisées avec :**
  - Les propriétés associées aux colonnes de la table
  - Les propriétés de navigation (clés étrangères) du type par défaut vers le type spécialisé
- **Ajouter les propriétés et méthodes non mappées avec la table**
  - Etat Entite / pour scénario complexe de mise à jour
  - Redéfinir la notion d'égalité d'objet
- **Définir le type de base (Personne) Abstract**



Au niveau des modèles d'entité, dans une classe de métadonnées associée pour les protéger d'une destruction accidentelle

- Compléter la définition des propriétés

```
[MetadataType(typeof(EtablissementMetaData))]  
public partial class Etablissement : IEntityPOCOState  
{
```

1 référence

```
public sealed class EtablissementMetaData  
{
```

```
[RegularExpression(@"^[A-Z]{1}[a-z]{1,}$",  
    ErrorMessage = "Le {0} doit commencer par  
public string DesignationEtablissement;
```

## Ajouter les DbSet des entités spécialisées

```
public virtual DbSet<Tiers> Tiers { get; set; }
```

2 références

```
public virtual DbSet<Stagiaire> Stagiaire { get; set; }
```

## Amender la configuration Mapping / déplacement des propriétés

```
modelBuilder.Entity<Tiers>();  
modelBuilder.Entity<CollaborateurAfpa>(entity =>  
{  
    entity.Property(e => e.MatriculeCollaborateurAfpa)  
        .HasMaxLength(8)  
        .IsUnicode(false)  
        .IsFixedLength();  
});
```

## Seul le type de base est mappé à une table

## Définir le discriminant

```
modelBuilder.Entity<Personne>(entity =>
{
    entity.ToTable("Personne")
        .HasDiscriminator<string>("CatPersonne")
        .HasValue<Tiers>("P")
        .HasValue<CollaborateurAfpa>("F")
        .HasValue<Stagiaire>("S");
})
```

**Au niveau des modèles d'entité en lien avec les entités spécialisées**

**Exemple : Formation**

**Amendement des propriétés de navigation pour qu'elles soient conformes au design des entités spécialisées**

**Personne → Stagiaire**

REFERENCE

```
public virtual Stagiaire IdPersonneNavigation { get; set; }
```

**Certaines méthodes ne sont accessibles que via l'opérateur de partition du type de base. Exemple Include**

```
var collaborateursAfpa2 = dbContext.
```

```
    Personne.OfType<CollaborateurAfpa>()
```

```
    .Include(o=>o.OffreFormation).ToList();
```

**Certaines méthodes propres aux DbSet mappées / Tables telle que Find qui nécessite une primary key**

# Suivi des modifications Mode déconnecté

