Configurer la classe Transaction comme **porteuse de données** entre Compte

et Banque avec une clé primaire composée de trois propriétés

NumeroCompteFk, DABFk et Date

builder.HasOne(p => p.Compte)

.WithMany(p => p.Transactions)

.HasForeignKey(p => p.CompteFk)

.OnDelete(DeleteBehavior.Restrict);

//config de 1 \* DAB et Transaction

builder.HasOne(p => p.DAB)

.WithMany(p => p.Transactions)

.HasForeignKey(p => p.DabFK)

.OnDelete(DeleteBehavior.Restrict);

//3-b

builder.HasKey(p => new

{

p.DabFK,

p.CompteFk,

p.Date});

**//TPT**

**AM context**

modelBuilder.Entity<TransactionRetrait>()

.ToTable("TransactionRetraits");

modelBuilder.Entity<TransactionTransfert>()

.ToTable("TransactionTransferts");

**Configurer la relation Many To Many** entre Fournisseur et Produit en précisant

Facture comme étant **le nom de la table associative.**

Context.CS :

modelBuilder.Entity<Produit>()

.HasMany(p => p.Fournisseurs)

.WithMany(p => p.Produits)

.UsingEntity(p => p.ToTable("Facture"));

OR :

**// config de \* \***

ProduitsConfiguration

builder.HasMany(p => p.Fournisseurs)

.WithMany(p => p.Produits)

.UsingEntity(p => p.ToTable("Facture"));

puis call in the context. cs

modelBuilder.ApplyConfiguration(new ProduitsConfiguration());

//////////////////////////////////////////////////////////////

le **Discriminator de la hiérarchie d'héritage par défaut (TPH) e**ntre Produit, Chimique et Biologique afin qu’elle soit **nommée TypeProduit de type char** et qui prend la valeur C si le type du Produit est Chimique,

la valeur B si le type de Produit est Biologique et la valeur P sinon.

//tph => config IN Context.cs

//2 eme methode de fluent API

//modelBuilder.Entity<Produit>()

// .HasDiscriminator<char>("TypeProduit")

// .HasValue<Produit>('P')

// .HasValue<Biologique>('B')

// .HasValue<Chimique>('C');

OR :

in ProduitConfiguration:

//TPH

builder.HasDiscriminator<char>("TypeProduit")

.HasValue<Produit>('P')

.HasValue<Biologique>('B')

.HasValue<Chimique>('C');

Web:

Liste déroulante

**// GET: ProduitController/Create**

public ActionResult Create()

{

ViewBag.CategoryList = new SelectList(sc.GetAll(), "Id", "Nom");

return View();

}

// POST: ProduitController/Create

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Produit p)

{

try

{

sp.Add(p);

sp.Commit();

return RedirectToAction(nameof(Index));

}

catch

{

return View();

}

}

**in the create.html**

///////dynamique liste

<div class="form-group">

<label asp-for="CategorieFK" class="control-label"></label>

@\* <input asp-for="CategorieFK" class="form-control" />

\*@

<select asp-for="CategorieFK" class="form-control"

asp-items="ViewBag.CategoryList"

></select>

<span asp-validation-for="CategorieFK" class="text-danger"></span>

</div>

////////////// Statique liste

<div class="form-group">

<label asp-for="Type" class="control-label"></label>

<select asp-for="Type" class="form-control"

asp-items="Html.GetEnumSelectList<Domain.Type>()">

</select>

</div>

**Barre de recherche**

**fel controller fel methode get index**

**//** el Name en parametre lezem ikon nafesou fel partie html

// GET: ProduitController

public ActionResult Index(string? Name)

{

if(Name == null)

return View(sp.GetAll());

return View(sp.GetMany(p => p.Nom.Equals(Name)));

}

**////////////mba3ed net3adew lel index.html**

<form asp-action="index">

<fieldset>

<legend> Recherche</legend>

Saisir un nom : <input type="text" name="Name" />

<input type="submit" value="Serach" />

</fieldset>

</form>

**///////////Photo**

creation dossier uploads tahet wwwroot

tzideha fel index bahhdha create melfou9  <form asp-action="Create" enctype="multipart/form-data

html <input type="file" name="file" class="form-control" />

<div class="form-group">

<label asp-for="Photo" class="control-label"></label>

<input type="file" name="file" class="form-control" />

<span asp-validation-for="Photo" class="text-danger"></span>

</div>

@Post create in Controller

public AActionResult Create(Cagnotte c , IFormFile file)

c.Photo = file.FileName;

if (file != null)

{

var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "uploads",

file.FileName);

using (System.IO.Stream stream = new FileStream(path, FileMode.Create))

{

file.CopyTo(stream);

}

}

bech l photo tokhrej fel index.html lezem tzid

<td>

<img src="~/uploads/@item.Photo " width="50" height="50" />

</td>

////////////Ajouter un Lien

<td>

@Html.ActionLink(linkText: "Entreprise", actionName: "Details", controllerName: "Entreprise", routeValues: new { id = item.EntrepriseId }, htmlAttributes: null)

</td>