

**Exercice 1**

Transformer un modèle **Source** contenant une métaclasses :

Source!Element { name : String } en un modèle **Target** contenant : Target!Item { label : String }

Règle :

**Item.label = Element.name**

**Exercice 2 : Calculer un attribut dérivé**

On a un modèle UML simple :

UML!Person { firstName : String, lastName : String }, On veut créer : OUT!User { fullName : String }

Règle :

**fullName = firstName + ' ' + lastName**

**Exercice 3 : Filtrer certains éléments**

On possède :

IN!Product { name : String, price : Real }, OUT!CheapProduct { name : String }

Transformer **seulement les produits dont le price ≤ 10.**

**Exercice 4 : Filtrer et transformer les classes UML**

Transformer uniquement les classes UML **marquées comme persistantes** en tables relationnelles.  
Chaque UML!Class contient un attribut booléen **isPersistent : Boolean**.

- Si **isPersistent = true** → créer une Table
- Sinon → ignorer la classe

**Exercice 5 : Ajouter un préfixe aux noms**

Transformer toutes les UML!Class en Rel!Table, mais ajouter un préfixe "T\_" au nom de chaque table.

Exemple :

- UML Class **Person** → Rel Table **T\_Person**

**Exercice 6 : Lazy Rule pour attributs**

Source : UML!Class { name, attribute: UML!Attribute }, Cible : Rel!Table { name, columns: Rel!Column }

- Transformer chaque Class → Table
- Chaque Attribute → Column **via une lazy rule**

### **Exercice 7 : Transformation conditionnelle**

Source : UML!Employee { name, salary }, Cible : Target!HighSalaryEmployee { name }

- Transformer seulement les employés dont salary > 5000.

### **Exercice 8: Transformation avec helper global**

Source : UML!Person { firstName, lastName }

Cible : Target!PersonInfo { initials }

- initials = première lettre du prénom + première lettre du nom (ex : "Layachi Soumia" → "LS")