

Exercice : 1 Nous avons un méta modèle **Source** composé d'un seul élément :

Méta modèle Source (MM1)

- **Person**
 - name : String

Méta modèle Cible (MM2)

- **User**
 - username : String

Objectif de la transformation :

Transformer chaque **Person** du modèle source en un **User** dans le modèle cible, avec : User.username = Person.name

Exercice : 2 Transformation d'un modèle “Livre” vers un modèle “Document”

Métamodèle Source (LibraryMM)

Book

- title : String
- pages : Integer

Métamodèle Cible (DocMM)

Document

- name : String
- length : Integer

Objectif de la transformation :

Pour chaque **Book**, créer un **Document**, avec :

- Document.name = Book.title
- Document.length = Book.pages

Exercice 3 .

Objectif

- Transformer **UML::Class** → **Table**
- Transformer **UML::Attribute** → **Column**
- Conserver le **même nom** et le **même type**

Exercice 4: Transformation Student → Learner (avec condition)

Métamodèle Source (SchoolMM)

Student

- name : String
- age : Integer

Métamodèle Cible (TrainingMM)

Learner

- fullname : String
- category : String (“adult” ou “minor”)

Objectif :

Transformer chaque **Student** en un **Learner** :

- Le champ fullname = name
- Le champ category dépend de l'âge :
 - si age \geq 18 → “adult”
 - sinon → “minor”

Exercice 5**Exercice 5 (QVT-Operational) — Student → Learner****Source (SchoolMM)**

- Student : name : String, age : Integer

Cible (TrainingMM)

- Learner : fullname : String, category : String // "adult" ou "minor"

Objectif

Pour chaque Student du modèle source, créer un Learner dans le modèle cible :

- Learner.fullname = Student.name
- Learner.category = "adult" si age >= 18 sinon "minor"

Exercice 6

Écrire une transformation QVT-Operational qui :

1. transforme chaque **UML::Class** en **Java::JavaClass**
2. transforme chaque **Attribute** en :
 - une **JavaField**
 - deux méthodes :
 - **getter** : getNomAttribut()
 - **setter** : setNomAttribut(type value)
3. génère un corps de méthode simple (du texte).