

Objectifs pédagogiques

- Apprendre à utiliser `$lookup` pour simuler des jointures entre collections.
 - Structurer les résultats avec `$unwind` et `$project`.
 - Réaliser une jointure simple et une jointure imbriquée multi-niveaux.
-

Lab 2 : Jointures avec `$lookup`

Étape 1 — Préparer les collections

- Créez deux collections : `students` et `courses`.
- Chaque étudiant doit avoir une liste d'identifiants de cours.

Exemple de structure attendue (sans la solution) :

```
// Collection students
{
  "_id": ObjectId "...",
  "name": "Ali",
  "courses": [ ObjectId "...", ObjectId "..."]
}

// Collection courses
{
  "_id": ObjectId "...",
  "title": "Mathématiques"
}
```

Étape 2 — Réaliser une jointure simple

Utilisez `$lookup` pour afficher la liste des étudiants avec les détails de leurs cours.

Objectif : Joindre les documents de la collection `courses` à ceux de `students`.

Étape 3 — Structurer le résultat

Ajoutez les étapes suivantes :

- `$unwind` pour transformer les tableaux en documents individuels.
- `$project` pour n'afficher que le nom de l'étudiant et le nom du cours.

Format attendu :

```
{  
  "name": "Ali",  
  "course_title": "Math matiques"  
}
```

Étape 4 — Jointure imbriquée (lookup dans un lookup)

Objectif : Simuler une jointure multi-niveaux.

- Ajouter une collection professors.
- Chaque cours contient un champ professorId.
- Chaque professeur a un name.
- Réaliser une double jointure :
 1. Étudiant → Cours.
 2. Cours → Professeur.

Format de sortie attendu :

```
{  
  "student_name": "Ali",  
  "course_title": "Math matiques",  
  "professor_name": "Dr. Karim"  
}
```

Astuce

Pour faire une jointure imbriquée, utilisez \$lookup à l'intérieur d'un pipeline dans from avec let et pipeline.

À vous de jouer !

Implémentez toutes les étapes ci-dessus sans consulter la solution. Testez vos résultats dans MongoDB Compass ou le shell.