

TP 6 : LES JOINTURES SOUS MONGODB

Le but de ce TP6 est d'étudier l'équivalent des jointures (SGBDR) en MongoDB avec \$lookup



Afin de voir \$lookup en situation, nous allons utiliser trois collections qui stockeront des élèves, leurs devoirs individuels et les projets sur lesquels ils travaillent à plusieurs :

1. LES TRAVAUX INDIVIDUELS

Commençons par les travaux individuels des étudiants, voici les collections impliquées, aucun index n'a été mis pour garder les choses les plus simples possibles.

- Créer une collection **élèves** qui comporte les élèves suivants :

```
db.eleves.insertMany([
  {"nom": "Zakaria ABOURIDA", "code": "NAT123"},
  {"nom": "Salma KAMAR", "code": "NAT125"},
  {"nom": "Mohamed BOUSSELHAM", "code": "NAT120"},
  {"nom": "Rani KAROUI", "code": "NAT127"}
])
```

- Créer une collection **devoirs** qui comporte les notes des devoirs des élèves de différentes matières correspondant au codes élèves suivants :

```
db.devoirs.insertMany([
  {"code": "NAT123", "matiere": "SQL", "note": 12},
  {"code": "NAT123", "matiere": "JAVA", "note": 10},
  {"code": "NAT125", "matiere": "JAVA", "note": 11.75},
  {"code": "NAT120", "matiere": "Dev_Mobile", "note": 18},
  {"code": "NAT127", "matiere": "NoSQL", "note": 19}
])
```

- Nous allons partir d'**élèves** pour aller vers **devoirs**, dans les deux cas notre nom de champ de « jointure » est **code**:

```
db.eleves.aggregate([
  {
    $lookup: {
      "from": "devoirs",
      "localField": "code",
      "foreignField": "code",
      "as": "detail_notes"
    }
  }
])
```

TP6 : LES JOINTURES AVEC LOOKUP SOUS MONGODB

Cela donnera un résultat qui exprime l'idée générale : les notes sont « raccrochées » aux noms des élèves :

```
{ "_id" : ObjectId("5ff39dd65a96c99956e55441"), "nom" : "Zakaria ABOURIDA", "code" : "NAT123", "detail_notes" : [ { "_id" : ObjectId("5ff39e025a96c99956e55445"), "code" : "NAT123", "matiere" : "SQL", "note" : 12 }, { "_id" : ObjectId("5ff39e025a96c99956e55446"), "code" : "NAT123", "matiere" : "JAVA", "note" : 10 } ] }
{ "_id" : ObjectId("5ff39dd65a96c99956e55442"), "nom" : "Salma KAMAR", "code" : "NAT125", "detail_notes" : [ { "_id" : ObjectId("5ff39e025a96c99956e55447"), "code" : "NAT125", "matiere" : "JAVA", "note" : 11.75 } ] }
{ "_id" : ObjectId("5ff39dd65a96c99956e55443"), "nom" : "Mohamed BOUSSELHAM", "code" : "NAT120", "detail_notes" : [ { "_id" : ObjectId("5ff39e025a96c99956e55448"), "code" : "NAT120", "matiere" : "Dev_Mobile", "note" : 18 } ] }
{ "_id" : ObjectId("5ff39dd65a96c99956e55444"), "nom" : "Rani KAROUI", "code" : "NAT127", "detail_notes" : [ { "_id" : ObjectId("5ff39e025a96c99956e55449"), "code" : "NAT127", "matiere" : "NoSQL", "note" : 19 } ] }
```

- Si nous trouvons que l'affichage n'est pas assez clair, nous pouvons ajouter une étape **project** pour rendre cet affichage plus correct en éliminant les id et les champs redondants :

```
db.eleves.aggregate([
  {
    $lookup: {
      "from": "devoirs",
      "localField": "code",
      "foreignField": "code",
      "as": "detail_notes"
    }
  },
  {
    $project: {
      "_id": 0,
      "detail_notes._id": 0,
      "detail_notes.code": 0
    }
  }
])
```

Dont voici le résultat

```
{ "nom" : "Zakaria ABOURIDA", "code" : "NAT123", "detail_notes" : [ { "matiere" : "SQL", "note" : 12 }, { "matiere" : "JAVA", "note" : 10 } ] }
{ "nom" : "Salma KAMAR", "code" : "NAT125", "detail_notes" : [ { "matiere" : "JAVA", "note" : 11.75 } ] }
{ "nom" : "Mohamed BOUSSELHAM", "code" : "NAT120", "detail_notes" : [ { "matiere" : "Dev_Mobile", "note" : 18 } ] }
{ "nom" : "Rani KAROUI", "code" : "NAT127", "detail_notes" : [ { "matiere" : "NoSQL", "note" : 19 } ] }
```

2. LES TRAVAUX DES ETUDIANTS EN PROJETS

Il est possible d'évaluer les étudiants par des projets. Un projet port un nom et peut être réalisé par un ou plusieurs étudiants, lorsqu'un projet reçoit une note, tous les étudiants intervenants vont avoir cette note de projet.

- Créez une collection « projet » qui comporte ces informations sur le projet

```
db.projets.insertMany([
  {"codes": ["NAT123", "NAT125"], "matiere": "Robotique", "note": 15},
  {"codes": ["NAT120", "NAT127"], "matiere": "IHM", "note": 12},
  {"codes": ["NAT123", "NAT125", "NAT120"], "matiere": "WEB", "note": 18}
])
```

TP6 : LES JOINTURES AVEC LOOKUP SOUS MONGODB

- Le champ codes est un **tableau** ici, pour que chaque élève puisse récupérer sa note, il va falloir **éclater** ce tableau à l'aide de **\$unwind** avant d'utiliser **lookup** dessus, nous allons partir de la collection **projets** cette fois :

```
db.projets.aggregate([
  {
    $unwind: "$codes"
  },
  {
    $lookup: {
      "from": "eleves",
      "localField": "codes",
      "foreignField": "code",
      "as": "eleve"
    }
  },
  {
    $project: {
      "_id": 0,
      "codes": 0,
      "eleve._id": 0
    }
  }
])
```

Dont voici le résultat

```
{ "matiere" : "Robotique", "note" : 15, "eleve" : [ { "nom" : "Zakaria ABOURIDA", "code" : "NAT123" } ] }
{ "matiere" : "Robotique", "note" : 15, "eleve" : [ { "nom" : "Salma KAMAR", "code" : "NAT125" } ] }
{ "matiere" : "IHM", "note" : 12, "eleve" : [ { "nom" : "Mohamed BOUSSELHAM", "code" : "NAT120" } ] }
{ "matiere" : "IHM", "note" : 12, "eleve" : [ { "nom" : "Rani KAROUI", "code" : "NAT127" } ] }
{ "matiere" : "WEB", "note" : 18, "eleve" : [ { "nom" : "Zakaria ABOURIDA", "code" : "NAT123" } ] }
{ "matiere" : "WEB", "note" : 18, "eleve" : [ { "nom" : "Salma KAMAR", "code" : "NAT125" } ] }
{ "matiere" : "WEB", "note" : 18, "eleve" : [ { "nom" : "Mohamed BOUSSELHAM", "code" : "NAT120" } ] }
```

- Nous allons rajouter une **étape addFields** pour que le tableau contenant un seul document (**eleve**) devienne un document tout simple et pas un une table, voici le pipeline final :

```
db.projets.aggregate([
  {
    $unwind: "$codes"
  },
  {
    $lookup: {
      "from": "eleves",
      "localField": "codes",
      "foreignField": "code",
      "as": "eleve"
    }
  },
  {
    $addFields: {
      "eleve": { $arrayElemAt: ["$eleve", 0] }
    }
  },
  {
    $project: {
      "_id": 0,
      "codes": 0,
```

TP6 : LES JOINTURES AVEC LOOKUP SOUS MONGODB

```
        "eleve._id": 0
    }
}
])
```

Dont voici le résultat

```
{ "matiere" : "Robotique", "note" : 15, "eleve" : { "nom" : "Zakaria ABOURIDA", "code" : "NAT123" } }
{ "matiere" : "Robotique", "note" : 15, "eleve" : { "nom" : "Salma KAMAR", "code" : "NAT125" } }
{ "matiere" : "IHM", "note" : 12, "eleve" : { "nom" : "Mohamed BOUSSELHAM", "code" : "NAT120" } }
{ "matiere" : "IHM", "note" : 12, "eleve" : { "nom" : "Rani KAROUI", "code" : "NAT127" } }
{ "matiere" : "WEB", "note" : 18, "eleve" : { "nom" : "Zakaria ABOURIDA", "code" : "NAT123" } }
{ "matiere" : "WEB", "note" : 18, "eleve" : { "nom" : "Salma KAMAR", "code" : "NAT125" } }
{ "matiere" : "WEB", "note" : 18, "eleve" : { "nom" : "Mohamed BOUSSELHAM", "code" : "NAT120" } }
```

- Complétez le travail en enregistrant la sortie de cette requête dans une collection nommée « **note_projet** »

```
db.projets.aggregate([
{
  $unwind: "$codes"
},
{
  $lookup: {
    "from": "eleves",
    "localField": "codes",
    "foreignField": "code",
    "as": "eleve"
  }
},
{
  $addFields: {
    "eleve": { $arrayElemAt: ["$eleve", 0] }
  }
},
{
  $project: {
    "_id": 0,
    "codes": 0,
    "eleve._id": 0
  }
},
{
  "$out" : "note_projet"
}
])
```