

UE L315 - EF - Partie 2

Hugo Lignères

06/02/2025



Exercice 1

Question 1

L'équivalent de cette table serait, au format Mongo, la collection suivante :

```
[
  {
    "utilisateur_id": ObjectId(1),
    "nom": "Ligneres",
    "prenom": "Hugo",
    "email": "hugo.ligneres@orange.fr"
  }
]
```

J'ai repris le même nom des champs dans la base au format SQL. Par contre, au format Mongo, pas besoin de préciser au préalable le type de données de chaque champs ainsi que la longueur maximale. Donc je n'ai pas indiqué `INT VARCHAR(255)` ou encore `NOT NULL`. Par contre, j'ai ajouté `ObjectId` pour préciser qu'il s'agissait de l'ID de la collection.

Question 2

Au format Mongo, l'équivalent de cette requête serait :

```
db.utilisateurs.find({"prenom"}, {"nom": "John"})
```

Question 3

Au format Mongo, l'équivalent de cette requête serait :

```
db.utilisateurs.countDocuments()
```

Exercice 2

Question 1

Collection 1 : utilisateurs

```
[
  {
    "utilisateur_id": ObjectId(1),
    "nom": "Ligneres",
    "prenom": "Hugo",
    "email": "hugo.ligneres@orange.fr"
  }
]
```

Collection 2 : commandes

```
[
  {
    "commande_id": ObjectId(1),
```

```

    "utilisateur_id": 1,
    "date_commande": "2025-05-15",
    "montant": 10
  }
]

```

Collection 3 : lignes_commandes

```

[
  {
    "ligne_commande_id": ObjectId(1),
    "commande_id": 1,
    "produit_id": 1,
    "quantite": 10,
    "prix_unitaire": 20
  }
]

```

Collection 4 : produits

```

[
  {
    "produit_id": ObjectId(1),
    "nom": "Pantalon",
    "prix": "10",
  }
]

```

Question 2

Collection

```

[
  {
    "utilisateur_id": 1,
    "nom": "Ligneres",
    "prenom": "Hugo",
    "email": "hugo.ligneres@orange.fr"
  }
  {
    "produit_id": 1,
    "quantite": 10,
    "prix_unitaire": 20
  }
  {
    "commande_id": 1,
    "date_commande": "2025-05-15",
    "montant": 10
  }
]

```

J'ai voulu rangé ces données dans 3 sous-collections :

- Une pour les informations sur la personne : son id, son nom, son prénom et son email ;
- Une pour les informations sur le produit : son id, la quantité disponible et son prix unitaire ;
- Enfin, une pour les informations sur la commande : son id, la date et le montant de la commande.

Question 3

En utilisant la collection de la question 2 :

```
db.collection.find({"produit_id"}, {"utilisateur_id": 1})
```

Exercice 3

Proposition 1

La structure proposée me paraît plus cohérente que de tout mettre dans une seule collection. Les informations sont clairement séparées en fonction de l'information qu'elles apportent (informations sur le livre dans la collection `books`, informations sur l'auteur dans la collection `author`), et le lien entre les 2 collections est gérée par le champs `author` dans la collection `books`, relié au champs `_id` de la collection `authors`.

Proposition 2

La même chose que pour la proposition 1. De plus, avec le champs `biography` de la collection `authors` qui s'est ajouté, il serait encore moins pertinent de mélanger les 2 collections ensemble.

Proposition 3

Pour cette proposition, je pense qu'il serait préférable de tout ranger dans une seule collection. En effet, il y a de fortes chances que les id des livres dans le champs `books` de la table `authors` soient mal renseignés, ce qui peut poser des problèmes. À mon avis, il serait préférable de tout mettre dans la même collection, comme ça le champs `books` sera renseigné de titres de livres et non d'id, réduisant les erreurs possibles.