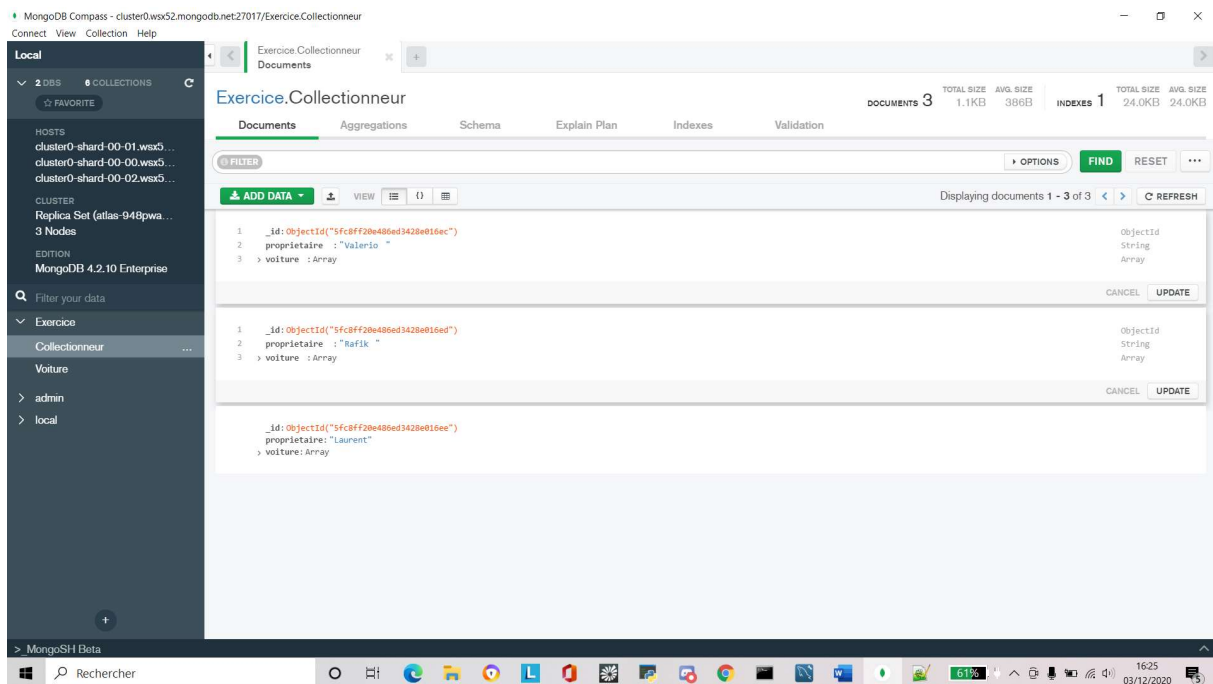
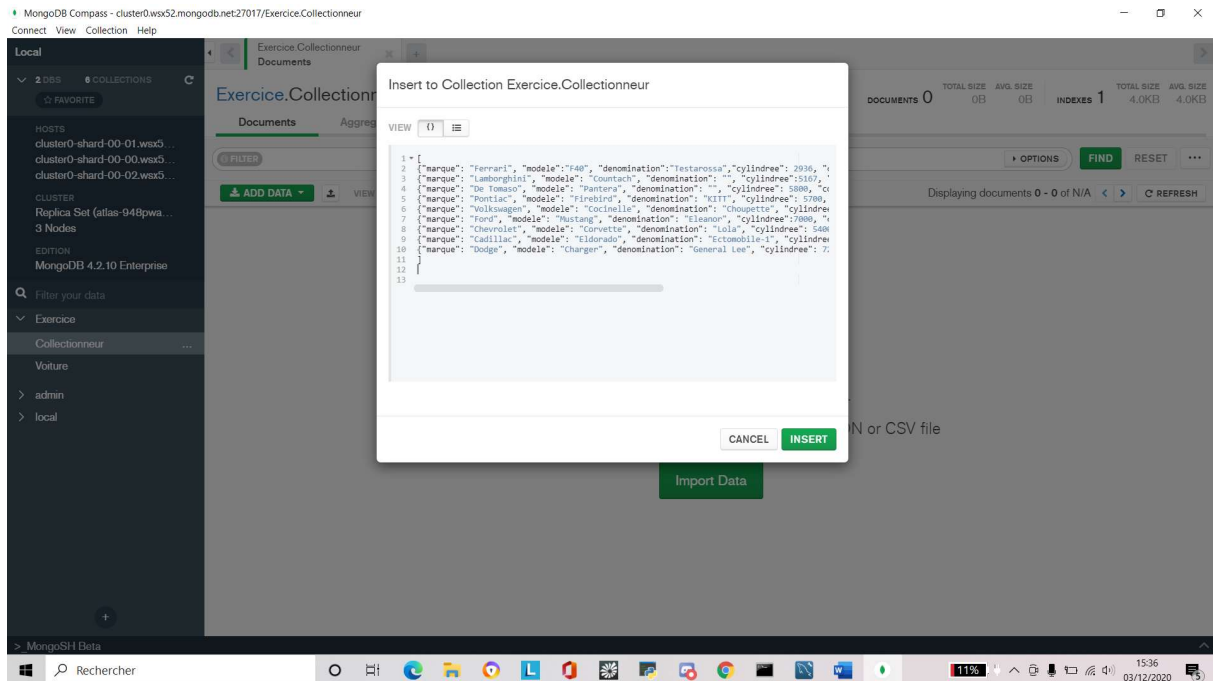


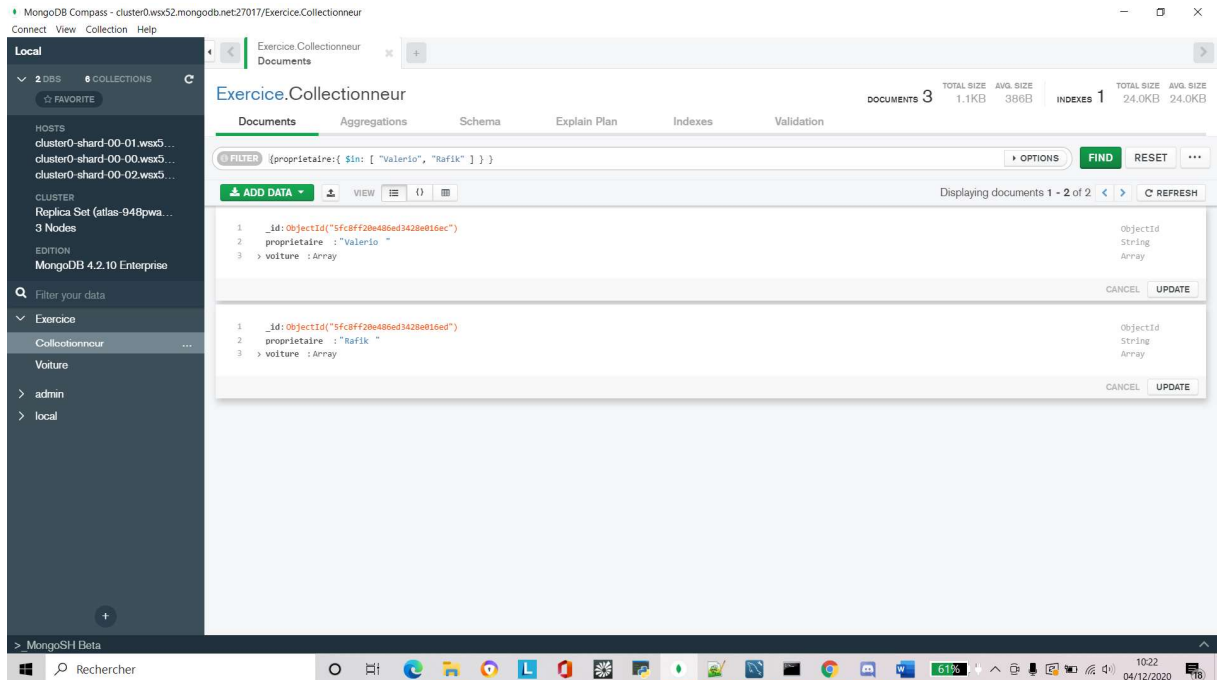
MongoDB Partie 2

1. Insérer les documents dans une collection collectionneur :



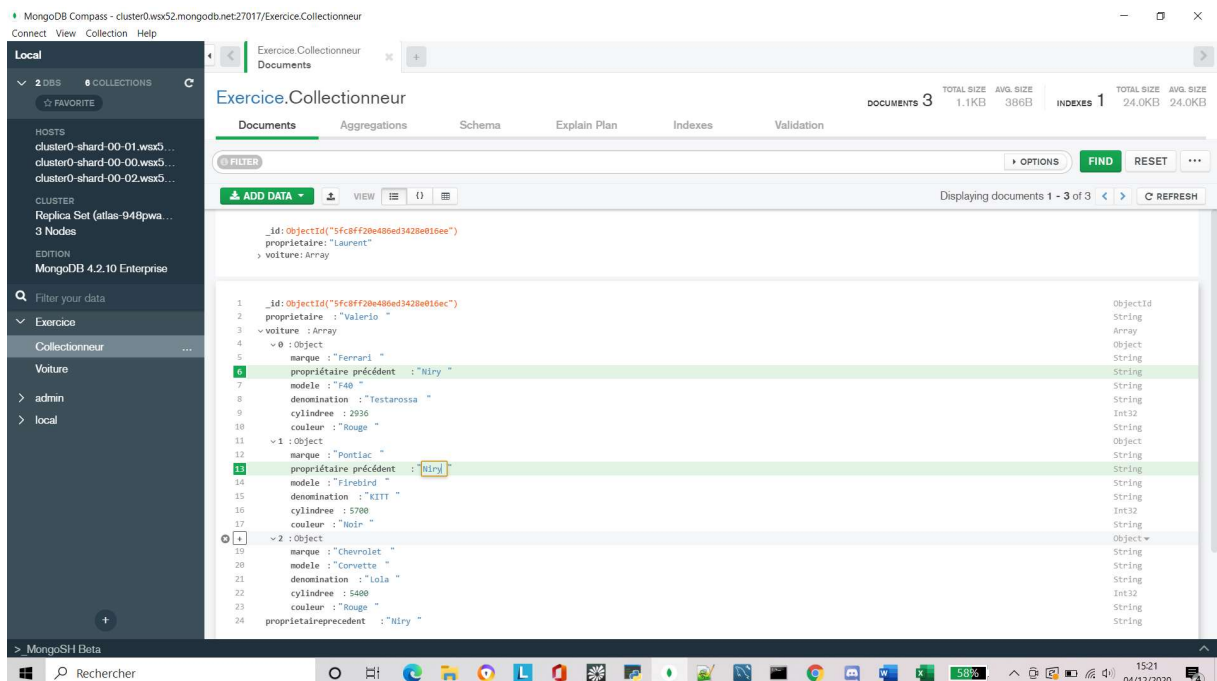
2. Sélectionner tous les documents avec Valerio ou Rafik en propriétaire :

```
{proprietaire:{ $in: [ "Valerio", "Rafik" ] }}
```



3. Ajouter un champ PropriétairePrécédent auquel vous donnerez la valeur Valerio
Mettre à jour la valeur du champ Propriétaire avec votre prénom

```
{proprietaire:"Valerio"}
```



4. Supprimer tous les documents ayant Couleur:Blanc

```
{ "voiture.couleur": "Blanc" }
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Local' sidebar displays the database structure: 2 DBS, 6 COLLECTIONS, and a list of hosts and clusters. The main panel shows the 'Exercice.Collectionneur' collection with a filter of `{ "voiture.couleur": "Blanc" }`. The document list shows two documents, both with 'couleur: "Blanc"'. The first document is expanded, showing details like '_id', 'proprietaire', and 'voiture' (an array of two car objects).

This screenshot is identical to the one above, but with an additional orange banner at the bottom of the document list area that reads 'Document Flagged For Deletion.' with 'CANCEL' and 'DELETE' buttons. This indicates that the document has been successfully marked for deletion by the filter operation.

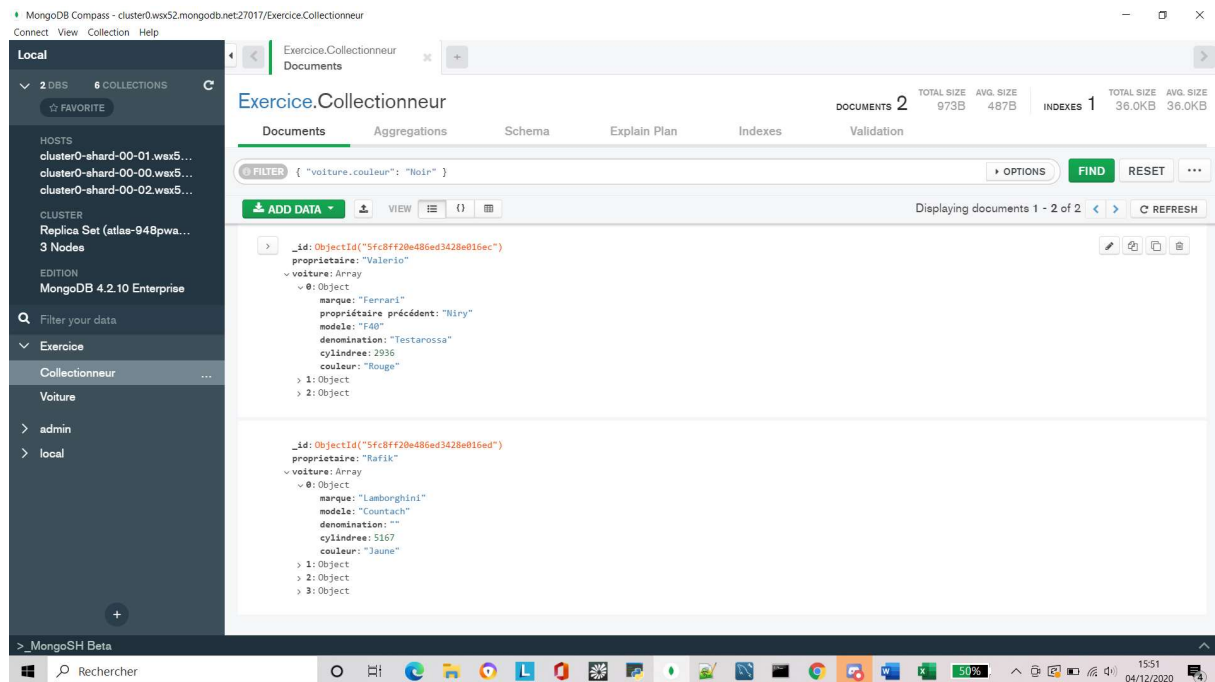
5. Sélectionner les voitures de couleur noire { "voiture.couleur": "Noir" }

The screenshot displays the MongoDB Compass interface. On the left sidebar, the 'Local' database is selected, showing a collection named 'Exercice'. The main panel shows the 'Documents' tab for the 'Exercice.Collectionneur' collection. A filter is applied: `{ "voiture.couleur": "Noir" }`. The results show two documents:

```
{ "_id": ObjectId("5fc8ff20e486ed3428e016ec"),  
  "proprietaire": "Valerio",  
  "voiture": Array  
}
```

```
{ "_id": ObjectId("5fc8ff20e486ed3428e016ed"),  
  "proprietaire": "Rafik",  
  "voiture": Array  
}
```

The interface also shows document statistics: 2 documents, 973B total size, 487B avg size, and 1 index (36.0KB total size, 36.0KB avg size). The bottom status bar indicates the date and time: 15:47, 04/12/2020.



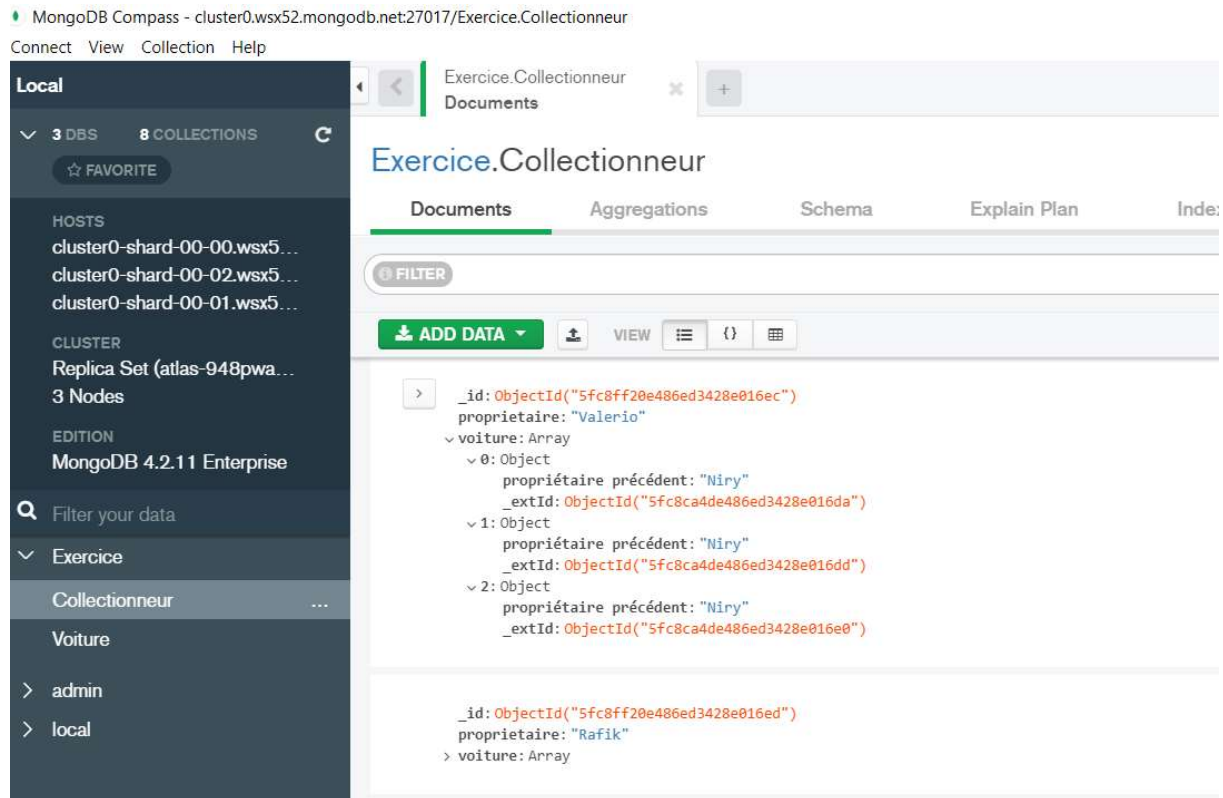
Remarque :

On ne peut pas accéder directement aux voitures, propriétaire et voiture apparaissent en premier, après il faut dérouler voiture afin d'avoir la liste des objets et ouvrir chaque objet pour enfin accéder aux caractéristiques des voitures (même souci d'ailleurs quand on a sélectionné les voitures blanches, Laurent apparaissait d'abord, etc..).

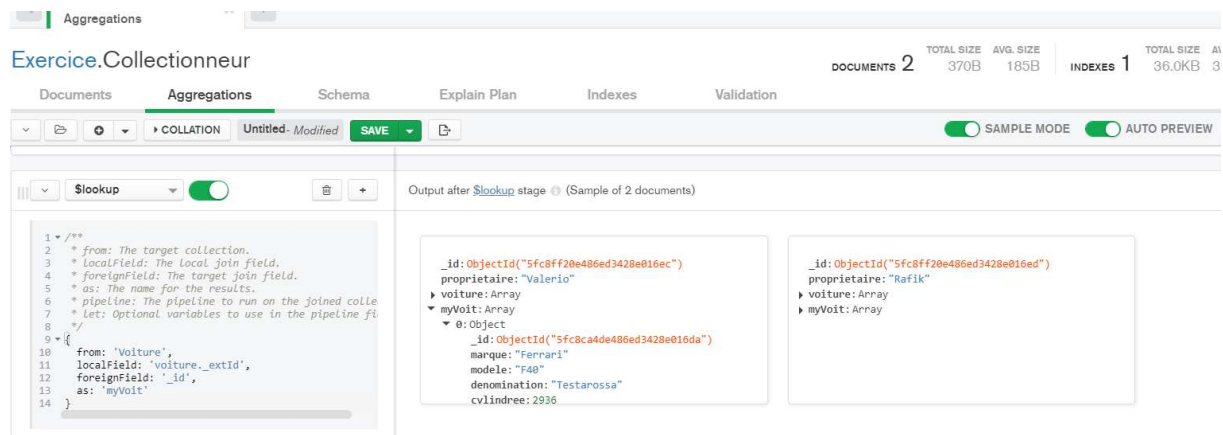
La requête fait ressortir toutes les voitures d'un collectionneur qui a une ou plusieurs voitures noires, même si elles ne sont pas noires.

Proposer une autre structure de la base de données :

On peut faire une jointure /agrégation entre les deux collections collectionneur et voiture.
Il faudra modifier collectionneur de telle façon qu'elle contienne une liste de clés étrangères (les object id) vers voiture.



De cette manière, on peut sélectionner les voitures répondant à des critères spécifiques.
On procède en 4 étapes.
Etape 1 : lookup



Etape 2 : unwind

The screenshot displays the Mocha test runner interface. On the left, the configuration for the 'Sunwind' stage is shown, including a path to the array field, an optional name for the array index, and a toggle for preserving null and empty arrays. On the right, the output after the 'Sunwind' stage is shown, displaying three sample documents with their respective IDs and properties.

Etape 3 : match



The screenshot shows the MongoDB Compass interface. On the left, the 'Query' tab is active, displaying a query in the 'Smatch' stage. The query is a JSON document with a 'myVoit.couleur' field set to 'Noir'. On the right, the 'Output after Smatch stage' tab is active, showing a sample of 1 document. The output is a JSON document with fields: '_id' (ObjectId), 'proprietaire' (Valerio), 'voiture' (Array), and 'myVoit' (Object). The 'myVoit' object contains fields: '_id' (ObjectId), 'marque' (Pontiac), 'modele' (Firebird), 'denomination' (KIT), and 'cylindres' (5700).

```

1 /**
2  * query: The query in MQL.
3  */
4 {
5   "myVoit.couleur": "Noir"
6 }

```

Output after **\$match** stage (Sample of 1 document)

```

  _id: ObjectId("5fc8ff20e486ed3428e016ec")
  proprietaire: "Valerio"
  voiture: Array
  myVoit: Object
    _id: ObjectId("5fc8ca4de486ed3428e016dd")
    marque: "Pontiac"
    modele: "Firebird"
    denomination: "KIT"
    cylindres: 5700

```

Etape 4 : project

The screenshot shows the VS Code editor with a file named `$project` open. The editor displays a JSON schema for a car, with comments in French. The output of the `$project` stage is shown in the right-hand pane, displaying a sample document with the following structure:

```
{
  "_id": "5fc8ff20e486ed3428e016ec",
  "proprietaire": "Valerio",
  "myVoit": {
    "_id": "5fc8ca4de486ed3428e016dd",
    "marque": "Pontiac",
    "modele": "Firebird",
    "denomination": "KIT",
    "cylindree": 5700,
    "couleur": "Noir",
    "proprietaire": "Nirv"
  }
}
```