

## CH3 MongoDB

### EXERCICE 1 page 40 (Daniel)

```
// The current database to use.  
const database = 'bdBateaux';  
const collection = 'skippers';  
  
// Create a new database.  
use(database);  
  
// Create a new collection.  
db.createCollection(collection);  
  
db.skippers.deleteMany({});  
  
db.skippers.insertMany([  
  { SKNUM: 1, SKNOM: "JEAN", SKPORT: "AJACCIO", SALAIRE: 3000 },  
  { SKNUM: 2, SKNOM: "PAUL", SKPORT: "AJACCIO", SALAIRE: 2000 },  
  { SKNUM: 3, SKNOM: "PIERRE", SKPORT: "ANTIBES"},  
  { SKNUM: 4, SKNOM: "MARIE", SKPORT: "BASTIA", SALAIRE: 1500 }  
])  
  
// Afficher tous les skippers  
db.skippers.find();  
  
// Afficher uniquement les noms et ports  
db.skippers.find({}, { SKNOM: 1, SKPORT: 1, _id: 0 });
```

### EXERCICE 2 Page 48 (Dumenicu)

```
db.getCollection('skippers').find(  
  {  
    $and: [  
      { "skport": "AJACCIO" },  
      { "salaire": { $gt: 2000 } }  
    ]  
  },  
  { _id: 0, sknom: 1 }  
);  
  
db.skippers.find(  
  {  
    "salaire" : { $exists: false }  
  },
```

```
{ sknom: 1 }  
);
```

### EXERCICE 3 Page 58

- 3.1. ajouter un champ salaire = 1500 aux skippers qui n'ont pas de salaire  
(M'hand AMRANE)

```
db.skippers.updateMany(  
  { SALAIRE: { $exists: false } },  
  { $set: { SALAIRE: 1500 } }  
);
```

- 3.2 . Augmenter de 500 euros le salaire de tous les skippers

(M'hand AMRANE)

```
db.skippers.updateMany(  
  {},  
  { $inc: { SALAIRE: 500 } }  
);
```

- 3.3. avec liste (SANNA Thomas)

```
db.skippers.updateMany(  
  { 'sknom': 'JEAN'},  
  {$set: { 'specialites': ["barreur", "transatlantique"]}}  
)
```

### EXERCICE 4 Page 89

- 4.1 Calcul de la moyenne (Yanis BELLAHCENE)

Base avec une seule collection skippers

```
// 1 - Calcul de la moyenne  
print("Salaire moyen des skippers");  
db.skippers.aggregate([  
  { $match: { SALAIRE: { $exists: true, $ne: "aucun" } } },  
  { $group: { _id: null, SalaireMoyen: { $avg: "$SALAIRE" } } },  
  { $project: { _id: 0, SalaireMoyen: 1 } }  
])
```

- 4.2 Compter le nombre de skippers par port (FABBRI Yohann)

```
db.skipper.aggregate([
```

```

{ $group: {
  _id: "$skport",
  skipperparport: { $count: {} }
}
]);

```

4.3 Trouver pour chaque port le salaire le plus élevé et le plus bas (FABBRI Yohann)

```

db.skipper.aggregate([
  { $group: {
    _id: "$skport",
    maxsalaire: { $max: "$salaire" },
    minsalire: { $min: "$salaire" }
  }}
]);

```

4.4 Donner pour chaque skipper, son nom et le nombre de croisières au départ de Bastia qu'il effectue (SANNA Thomas) Version avec deux collections : *skipppers* et *croisières*.

```

db.getCollection('skipppers').insertMany([
  { 'sknum': 1, 'sknom': "JEAN", 'skport': "AJACCIO", 'salaire': 3000 },
  { 'sknum': 2, 'sknom': "PAUL", 'skport': "AJACCIO", 'salaire': 2000 },
  { 'sknum': 3, 'sknom': "PIERRE", 'skport': "ANTIBES", 'salaire': 1200 },
  { 'sknum': 4, 'sknom': "MARIE", 'skport': "BASTIA", 'salaire': 1500 },
  { 'sknum': 5, 'sknom': "BAPTISTE", 'skport': "AJACCIO", 'salaire': 2010 },
]);

db.getCollection('croisières').insertMany([
  { "CROISNUM": "C001", "SKNUM": 1, "BATNUM": "B002", "DEPPORT": "BASTIA",
  "ARRPORT": "CALVI", "DEPDATE": ISODate("2024-07-10"), "ARRDATE":
  ISODate("2024-07-11") },
  { "CROISNUM": "C002", "SKNUM": 2, "BATNUM": "B001", "DEPPORT": "CALVI",
  "ARRPORT": "AJACCIO", "DEPDATE": ISODate("2024-07-15"), "ARRDATE":
  ISODate("2024-07-16") },
  { "CROISNUM": "C003", "SKNUM": 1, "BATNUM": "B003", "DEPPORT": "AJACCIO",
  "ARRPORT": "BASTIA", "DEPDATE": ISODate("2024-08-21"), "ARRDATE":
  ISODate("2024-08-22") },
  { "CROISNUM": "C004", "SKNUM": 3, "BATNUM": "B002", "DEPPORT": "BASTIA",
  "ARRPORT": "PORTO-VECCHIO", "DEPDATE": ISODate("2024-09-01"), "ARRDATE":
  ISODate("2024-09-03") },
  { "CROISNUM": "C005", "SKNUM": 1, "BATNUM": "B002", "DEPPORT": "AJACCIO",
  "ARRPORT": "CALVI", "DEPDATE": ISODate("2024-10-10"), "ARRDATE":
  ISODate("2024-10-11") },
  { "CROISNUM": "C006", "SKNUM": 4, "BATNUM": "B001", "DEPPORT": "PORTO-
  VECCHIO", "ARRPORT": "AJACCIO", "DEPDATE": ISODate("2024-11-05"), "ARRDATE":
  ISODate("2024-11-06") }
]
)

```

```

]);
// 4 Donner pour chaque skipper, son nom et le nombre de croisières au départ de Bastia qu'il effectue

// Version avec unwind et group

db.skippers.aggregate([
{
  $lookup: {
    from: "croisières",
    localField: "sknum",
    foreignField: "SKNUM",
    as: "croisières_du_skipper"
  }
},
{
  $unwind: "$croisières_du_skipper"
},
{
  $match: {
    "croisières_du_skipper.DEPPORT": "BASTIA"
  }
},
{
  $group: {
    _id: "$sknom",
    nombreCroisièresBastia: { $sum: 1 }
  }
},
{
  $project: {
    _id: 0,
    sknom: "$_id",
    nombreCroisièresBastia: 1
  }
}
])

```

4.4 Donner pour chaque skipper, son nom et le nombre de croisière au départ de Bastia qu'il effectue (avec croisière dans la même collection) (Yanis BELLAHCENE)

Version avec une seule collection skippers

```
// 4 - Nom et nb de croisières au dpart de bastia qu'ffectue chaque skipper
print("Nom et nombre de croisières au départ de Bastia effectuées par chaque skipper");
```

```

db.skippers.aggregate([
  { $unwind: "$CROISIERES" },
  {
    $group: {
      _id: "$SKNOM",
      nbCroisieresBastia: {
        $sum: {
          $cond: [{ $eq: ["$CROISIERES.DEPPORT", "BASTIA"] }, 1, 0]
        }
      }
    },
    { $project: { _id: 0, SKNOM: "$_id", nbCroisieresBastia: 1 } }
])

```

4.5 Donner la liste des ports d'arrivée pour chaque nom de skipper § (utiliser \$push ou \$addToSet dans le \$group)

Version avec une seule collection skippers (Korentin Georget)

```

use("bdBateaux");
db.skippers.aggregate([
{
  $unwind: "$CROISIERES"
},
{
  $group: {
    _id: "$SKNOM",
    SKNOM: { $first: "$SKNOM" },
    portsDarrivee: { $addToSet: "$CROISIERES.ARRPORT" }
  }
},
{
  $project: {
    _id: 0,
    SKNOM: 1,
    portsDarrivee: 1
  }
}
]);

```

Versions avec deux collections (SANNA Thomas)

```

db.skippers.aggregate([
  {
    $lookup: {
      from: "croisières",
      localField: "sknum", // sknum de skippers
      foreignField: "SKNUM", // sknum de croisières
      as: "croisières_du_skipper"
    },
  },
  {
    $unwind: "$croisières_du_skipper"
  },
  {
    $group: {
      _id: "$sknom",
      tous_les_ports_arrives: { $addToSet:
        "$croisières_du_skipper.ARRPORT" }
    }
  }
])

```

#### 4.6 Calculer le total des jours en mer pour chaque skipper

Version avec deux collections (SANNA Thomas)

```

db.skippers.aggregate([
  {
    $lookup: {
      from: "croisières",
      localField: "sknum",
      foreignField: "SKNUM",
      as: "croisières_du_skipper"
    }
  },
  {
    $unwind: "$croisières_du_skipper"
  },
  {
    $group: {
      _id: "$sknom",
      totalJoursEnMer: {
        $sum: {
          $dateDiff: {
            startDate: "$croisières_du_skipper.DEPDATE",

```

```

        endDate: "$croisieres_du_skipper.ARRDATE",
        unit: "day"
    }
}
}
}
])
)
```

Version avec une seule collection skippers (Korentin Georget)

```

use("bdBateaux");

db.skippers.aggregate([
{
    $unwind: "$CROISIERES"
},
{
    $group: {
        _id: "$SKNUM",
        SKNAME: { $first: "$SKNOM" },
        totalJoursMer: {
            $sum: {
                $dateDiff: {
                    startDate: "$CROISIERES.DEPDATE",
                    endDate: "$CROISIERES.ARRDATE",
                    unit: "day"
                }
            }
        }
    },
    {

```

```
$project: {  
    _id: 0,  
    SKNAME: 1,  
    totalUoursMer: 1  
}  
}  
]);
```