

Exercice 1

1: Créer une base de données nommé **university** et afficher la liste des bases de données existantes

```
test> use university
switched to db university
university> show dbs
BD_famille  64.00 KiB
admin       64.00 KiB
bd1         64.00 KiB
config      80.00 KiB
db1         64.00 KiB
local       72.00 KiB
movies      28.55 MiB
```

2: Créer les collections **students** et **courses** puis insérer les documents

```
university> db.createCollection("students")
{ ok: 1 }
university> db.createCollection("courses")
{ ok: 1 }
```

Créer **enrollments**

3: Combien y a-t-il d'étudiants dans la collection students ?

```
db.students.countDocuments();
// -> 5
```

4: Combien y a-t-il de cours dans la collection courses ?

```
db.courses.countDocuments();
// -> 5
```

5: Combien d'étudiants ont un gpa supérieur à 3.5 ?

```
db.students.find({ gpa: { $gt: 3.5 } }).count();
// -> 3
```

6: Combien de cours ont plus de 3 crédits ?

```
db.courses.find({ credits: { $gt: 3 } }).count();
// -> 2
```

7: Combien d'étudiants sont inscrits dans chaque filière (major) ?

```
university >
  db.students.aggregate({ $group: { _id: "$major", number: { $count: {} } } })[
    ({ _id: "Computer Science", number: 2 },
    { _id: "Physics", number: 1 },
    { _id: "Mathematics", number: 1 },
    { _id: "Chemistry", number: 1 })
  ];
```

8: Combien de cours sont offerts pour chaque nombre de crédits (credits) ?

```
university >
  db.courses.aggregate({ $group: { _id: "$credits", number: { $count: {} } } })[
    ({ _id: 3, number: 3 }, { _id: 4, number: 2 })
  ];
```

9: Combien de cours ont un titre contenant le mot "Computer" ?

```
db.courses.find({ title: { $regex: "Computer" } }).count();
// -> 1
```

10: Combien d'étudiants ont un gpa supérieur à 3.6 et un âge inférieur à 22 ans ?

```
db.students.find({ gpa: { $gt: 3.6 }, age: { $lt: 22 } }).count();  
// -> 2
```

11: Combien de cours ont un titre qui commence par "Introduction" ?

```
// Le symbole ^ en regex symbolise le début de la ligne  
university > db.courses.find({ title: { $regex: "^Introduction" } }).count();  
// -> 1
```

12: Combien d'étudiants ont un gpa inférieur à 3.2 ?

```
db.students.find({ gpa: { $lt: 3.2 } }).count();  
// -> 1
```

13: Combien de cours ont le même nombre de crédits que "CS102" ?

```
// On récupère le nombre de crédits de CS102 dans une première requête et on l'utilise comme condition de la deuxième requêtes  
university >  
  db.courses  
    .find({ credits: db.courses.findOne({ courseId: "CS102" }).credits })  
    .count();  
// -> 3
```

14: Combien d'étudiants ont un name qui contient la lettre "e" ?

```
db.students.find({ name: { $regex: "e" } }).count();  
// -> 5
```

15: Combien de cours ont un nombre de crédits égal à 3 ?

```
db.courses.find({ credits: 3 }).count();  
// -> 3
```

16: Combien d'étudiants sont inscrits dans les cours dont le titre contient "Data" ?

```
// Ne peut pas répondre pour le moment
```

17: Combien de cours sont offerts pour chaque major ?

```
// Ne peut pas répondre pour le moment
```

18: Combien d'étudiants sont inscrits dans les cours dont le nombre de crédits est supérieur à 3 ?

```
// Ne peut pas répondre pour le moment
```

19: Combien d'étudiants ont une gpa entre 3.0 et 3.5 ?

```
db.students.find({ gpa: { $gte: 3, $lte: 3.5 } }).count();  
// -> 2
```

20: Combien de cours ont un nombre de crédits supérieur ou égal à 4 ?

```
db.courses.find({ credits: { $gte: 4 } }).count();  
// -> 2
```

21: Combien d'étudiants ont un age supérieur à 21 et sont inscrits dans "Computer Science" ?

```
db.students.find({ age: { $gt: 21 }, major: "Computer Science" }).count();  
// -> 0
```

22: Combien d'étudiants sont inscrits dans les cours dont le titre contient "Physics" et ont un gpa supérieur à 3.5 ?

```
// Ne peut pas répondre pour le moment
```

23: Combien d'étudiants sont inscrits dans des cours ayant un nombre de crédits égal à 3 ?

```
// Ne peut pas répondre pour le moment
```

24: Combien de cours ont un nombre de crédits inférieur à la moyenne des crédits de tous les cours ?

```
// On fait une première requête pour trouver le nombre moyen de crédits
// Puis on l'utilise pour un find simple sur le nombre de crédits
db.courses
  .find({
    credits: {
      $lt: db.courses
        .aggregate({ $group: { _id: null, avg: { $avg: "$credits" } } })
        .toArray()[0].avg,
    },
  })
  .count();
// -> 3
```

Exercice 2

1: mongorestore

```
$ mongorestore --db gym ./gym/
2024-09-06T10:43:05.619+0200 The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead,
i.e. with --nsInclude=${DATABASE}.${COLLECTION}
2024-09-06T10:43:05.620+0200 building a list of collections to restore from gym dir
2024-09-06T10:43:05.622+0200 reading metadata for gym.Gymnases from gym/Gymnases.metadata.json
2024-09-06T10:43:05.622+0200 reading metadata for gym.Sportifs from gym/Sportifs.metadata.json
2024-09-06T10:43:05.659+0200 restoring gym.Sportifs from gym/Sportifs.bson
2024-09-06T10:43:05.671+0200 restoring gym.Gymnases from gym/Gymnases.bson
2024-09-06T10:43:05.671+0200 finished restoring gym.Sportifs (150 documents, 0 failures)
2024-09-06T10:43:05.682+0200 finished restoring gym.Gymnases (28 documents, 0 failures)
2024-09-06T10:43:05.682+0200 no indexes to restore for collection gym.Gymnases
2024-09-06T10:43:05.682+0200 no indexes to restore for collection gym.Sportifs
2024-09-06T10:43:05.682+0200 178 document(s) restored successfully. 0 document(s) failed to restore.
```

2: Vérifier les données

```
test> use gym
switched to db gym
gym> db.Sportifs.countDocuments()
150
gym> db.Gymnases.countDocuments()
28
```

3: Afficher les collections

```
gym> show collections
Gymnases
Sportifs
```

4: Donner le nombre de documents dans chaque collection

```
gym> db.Sportifs.countDocuments()
150
gym> db.Gymnases.countDocuments()
28
```

5: Quels sont les sportifs (identifiant, nom et prénom) qui ont entre 20 et 30 ans ?

```
db.Sportifs.find({ Age: { $gte: 20, $lte: 30 } }, { idSportif: 1, Nom: 1, Prenom: 1 });
```

6: Quels sont les gymnases de "Villetaneuse" ou de "Sarcelles" qui ont une surface de plus de 400 m2 ?

```
db.Gymnases.find({
  $or: [{ Ville: "VILLETANEUSE" }, { Ville: "SARCELLES" }],
```

```
    Surface: { $gt: 400 },
  });
```

7: Quels sont les sportifs (identifiant et nom) qui pratiquent du hand ball ?

```
db.Sportifs.find({ "Sports.Jouer": "Hand ball" }, { IdSportif: 1, Nom: 1 });
```

8: Dans quels gymnases et quels jours y a-t-il des séances de hand ball ?

```
db.Gymnases.aggregate([
  { $unwind: "$Seances" },
  // On garde seulement les séances de handball
  { $match: { "Seances.Libelle": "Hand ball" } },
  // On groupe par gymnase et on garde chaque jour une seule fois
  {
    $group: {
      _id: "$IdGymnase",
      IdGymnase: { $first: "$IdGymnase" },
      NomGymnase: { $first: "$NomGymnase" },
      Adresse: { $first: "$Adresse" },
      Ville: { $first: "$Ville" },
      jours: { $addToSet: "$Seances.Jour" },
    },
  },
]);
```

9: Dans quels gymnases peut-on jouer au hockey le mercredi après 15H ?

```
db.Gymnases.find(
  {
    Seances: {
      $elemMatch: {
        Libelle: "Hockey",
        Horaire: { $gt: 15 },
        Jour: "mercredi",
      },
    },
  },
  {
    IdGymnase: 1,
    NomGymnase: 1,
    Adresse: 1,
    Ville: 1,
  }
);
```

10: Quels sportifs (identifiant et nom) ne pratiquent aucun sport ?

```
db.Sportifs.find({ "Sports.Jouer": { $exists: false } });
```

11: Quels gymnases n'ont pas de séances le dimanche ?

```
db.Gymnases.find({ Seances: { $not: { $elemMatch: { Jour: "dimanche" } } } });
```

12: Quels gymnases ne proposent que des séances de basket ball ou de volley ball ?

```
// On cherche les gymnases avec aucune séance qui n'est ni du basket ni du volley
db.Gymnases.find({
  Seances: {
    $not: {
      $elemMatch: {
        $and: [
          { Libelle: { $ne: "Basket Ball" } },
          { Libelle: { $ne: "Volley Ball" } },
        ],
      },
    },
  },
});
```

13: Quels sont les entraîneurs qui sont aussi joueurs ?

```
// On cherche les sportifs avec une correspondance entre un sport joué et un sport entraîné
db.Sportifs.aggregate([
  { $unwind: "$Sports.Entraîner" },
```

```

    { $unwind: "$Sports.Jouer" },
    {
      $match: {
        $expr: {
          $eq: ["$Sports.Entraîner", "$Sports.Jouer"],
        },
      },
    },
  ],
  {
    $group: {
      _id: "$_id",
      IdSportif: { $first: "$IdSportif" },
      Nom: { $first: "$Nom" },
      Prenom: { $first: "$Prenom" },
    },
  },
];

```

14: Quels sont les sportifs qui sont des conseillers ?

```

// Une première requête pour trouver les id des conseillers
// Puis une deuxième pour avoir leurs informations
db.Sportifs.find( { IdSportif: { $in: db.Sportifs.distinct("IdSportifConseiller") } } );

```

15: Pour le sportif "Kervadec" quel est le nom de son conseiller ?

```

db.Sportifs.findOne(
  {
    IdSportif: db.Sportifs.findOne( { Nom: "KERVADEC" }).IdSportifConseiller,
  },
  { Nom: 1 }
);

```

16: Quels entraîneurs entraînent du hand ball et du basket ball ?

```

db.Sportifs.find( {
  $and: [ { "Sports.Entraîner": "Hand ball" }, { "Sports.Entraîner": "Basket ball" } ],
} );

```

17: Quels sont les couples de sportifs (identifiant et nom et prénom de chaque) de même âge ?

```

db.Sportifs.aggregate([
  {
    // Faire des groupes de sportifs du même âge
    $group: {
      _id: "$Age",
      sportifs: {
        $push: {
          IdSportif: "$IdSportif",
          Nom: "$Nom",
          Prenom: "$Prenom",
        },
      },
    },
  },
]);

```

18: Quelle est la moyenne d'âge des sportives qui pratiquent du basket ball ?

```

db.Sportifs.aggregate([
  { $match: { Sexe: "F", "Sports.Jouer": "Basket ball" } },
  {
    $group: {
      _id: null,
      age: { $avg: "$Age" },
    },
  },
]);

```

19: Quels sont les sportifs les plus jeunes ?

```

// Un aggregate pour trouver l'âge minimum puis une find pour récupérer tous les sportifs ayant cet âge
db.Sportifs.find( {
  Age: db.Sportifs.aggregate([
    {
      $group: { _id: null, age_min: { $min: "$Age" } },
    },
  ] ),
} );

```

```
)).toArray()[0].age_min,
});
```

20: Quels sont les gymnases de “Stains” ou de “Montmorency” qui ont la plus grande surface ?

```
db.Gymnases.find({
  $or: [{ Ville: "STAINS" }, { Ville: "MONTMORENCY" }],
  Surface: db.Gymnases.aggregate([
    {
      $match: {
        $or: [{ Ville: "STAINS" }, { Ville: "MONTMORENCY" }],
      },
    },
    {
      $group: {
        _id: null,
        surface_max: { $max: "$Surface" },
      },
    },
  ]).toArray()[0].surface_max,
});
```

21: Quels entraîneurs n’entraînent que du hand ball ou du basket ball ?

```
db.Sportifs.find({
  "Sports.Entraîner.0": { $exists: false }, // exclure les listes
  $or: [{ "Sports.Entraîner": "Hand ball" }, { "Sports.Entraîner": "Basket ball" }],
});
```

22: Quels sont les couples de sportifs (identifiant et nom et prénom de chaque) de même âge avec le même conseiller ?

```
db.Sportifs.aggregate([
  {
    $group: {
      _id: {
        age: "$Age",
        conseiller: "$IdSportifConseiller",
      },
      sportifs: {
        $push: {
          IdSportif: "$IdSportif",
          Nom: "$Nom",
          Prenom: "$Prenom",
        },
      },
    },
  },
]);
```

23: Quels sportifs n’ont pas de conseillers ?

```
db.Sportifs.find({ IdSportifConseiller: { $exists: false } });
```

24: Pour chaque sportif donner le nombre de sports qu’il arbitre.

```
db.Sportifs.aggregate([
  {
    $unwind: {
      path: "$Sports.Arbitrer",
      // On garde les enregistrements avec le champ manquant
      preserveNullAndEmptyArrays: true,
    },
  },
  {
    $group: {
      _id: "$_id",
      IdSportif: { $first: "$IdSportif" },
      Nom: { $first: "$Nom" },
      Prenom: { $first: "$Prenom" },
      // On ajouter un attribut qui sera null si le champ n'existe pas
      arbitrer: { $first: "$Sports.Arbitrer" },
      sports_arbitre: { $count: {} },
    },
  },
  {
    $project: {
      IdSportif: 1,
      Nom: 1,
      Prenom: 1,
      sports_arbitre: {
```

```

        $cond: {
          // On met 0 si le champ qu'on a créé est null
          if: { $eq: ["$arbirter", null] },
          then: { $literal: 0 },
          else: "$sports_arbitre",
        },
      },
    },
  },
});

```

25: Pour chaque gymnase de Stains donner par jour d'ouverture les horaires des premières et dernières séances.

```

db.Gymnases.aggregate([
  { $match: { Ville: "STAINS" } },
  { $unwind: "$Seances" },
  {
    $group: {
      _id: {
        _id: "$_id",
        jour: "$Seances.Jour",
      },
      IdGymnase: { $first: "$IdGymnase" },
      NomGymnase: { $first: "$NomGymnase" },
      Adresse: { $first: "$Adresse" },
      jour: { $first: "$Seances.Jour" },
      premiere_seance: { $min: "$Seances.Horaire" },
      derniere_seance: { $max: "$Seances.Horaire" },
    },
  },
]);

```

26: Pour chaque entraîneur de hand ball quel est le nombre de séances journalières qu'il assure ?

```

db.Gymnases.aggregate([
  { $unwind: "$Seances" },
  { $match: { "Seances.Libelle": "Hand ball" } },
  // On commence par compter les séances par semaine
  {
    $group: {
      _id: "$Seances.IdSportifEntraîneur",
      seances_par_semaine: { $count: {} },
    },
  },
  // Ensuite une projection pour diviser par 7 et avoir le nombre par jour
  {
    $project: {
      _id: 1,
      seances_par_jour: {
        $divide: ["$seances_par_semaine", 7],
      },
    },
  },
  // On ramène les infos des entraîneurs
  {
    $lookup: {
      from: "Sportifs",
      localField: "_id",
      foreignField: "IdSportif",
      as: "infos",
    },
  },
  { $unwind: "$infos" },
  {
    $project: {
      _id: "$infos._id",
      seances_par_jour: 1,
      IdSportif: "$infos.IdSportif",
      Nom: "$infos.Nom",
      Prenom: "$infos.Prenom",
    },
  },
]);

```

27: Quels sont les gymnases ayant plus de 15 séances le mercredi ?

```

db.Gymnases.aggregate([
  { $unwind: "$Seances" },
  { $match: { "Seances.Jour": "mercredi" } },
  {
    $group: {
      _id: "$_id",
      IdGymnase: { $first: "$IdGymnase" },
      NomGymnase: { $first: "$NomGymnase" },
      Adresse: { $first: "$Adresse" },
    },
  },
]);

```

```
      Ville: { $first: "$Ville" },
      seances_mercredi: { $count: {} },
    },
  },
  { $match: { seances_mercredi: { $gt: 15 } } } },
]);
```