
 FONDATION TUNISIE POUR LE DÉVELOPPEMENT	Centre ELIFE Djerba Formation : Devenir développeur FullStack JS	
Formateurs :	Anis ASSAS & Yessine ZEKRI	

Atelier : MongoDB

Création de la collection "produits" et insertion de quelques documents :

Pour démarrer, créons la collection "produits" et insérons quelques documents à titre d'exemple.

```
use ma_base_de_donnees

db.produits.insertMany([
  {
    _id: 1,
    designation: "Ordinateur portable",
    quantite_stock: 10,
    tva: 20,
    prix: 800,
    fournisseur: "Fournisseur A"
  },
  {
    _id: 2,
    designation: "Smartphone",
    quantite_stock: 15,
    tva: 19,
    prix: 600,
    fournisseur: "Fournisseur B"
  },
  {
    _id: 3,
    designation: "Imprimante",
    quantite_stock: 5,
    tva: 20,
    prix: 200,
    fournisseur: "Fournisseur A"
  }
])
```

1. Question : Combien de produits ont une quantité en stock supérieure à 10 ?

Réponse : Utilisez la méthode find avec un opérateur de comparaison (\$gt pour "greater than").

Résultat attendu : Vous devriez obtenir le produit avec id 2, car ils ont une quantité en stock supérieure à 10.

```
db.produits.find({ quantite_stock: { $gt: 10 } }).pretty()
```

2. Question : Quels sont les produits fournis par "Fournisseur A" ?

Réponse : Utilisez la méthode find avec le critère de recherche sur le champ "fournisseur".

Résultat attendu : Vous devriez obtenir les produits avec les id 1 et 3, car ils sont fournis par "Fournisseur A".

```
db.produits.find({ fournisseur: "Fournisseur A" }).pretty()
```

3. Question : Quels sont les produits avec un prix supérieur à 500 et une TVA de 20 % ?

Réponse : Utilisez la méthode find avec des opérateurs de comparaison (\$gt et \$eq).

Résultat attendu : Vous devriez obtenir le produit avec l'id 1, car il a un prix supérieur à 500 et une TVA de 20 %.

```
db.produits.find({ prix: { $gt: 500 }, tva: { $eq: 20 } }).pretty()
```

4. Question : Comment mettre à jour la quantité en stock du produit avec l'id 2 pour la réduire de 3 unités ?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$inc.

Résultat attendu : La quantité en stock du produit avec l'id 2 devrait être réduite de 3 unités.

```
db.produits.updateOne({ _id: 2 }, { $inc: { quantite_stock: -3 } })  
db.produits.find({_id:2}).pretty()
```

5. Question : Comment supprimer le produit avec l'id 3 de la collection ?

Réponse : Utilisez la méthode deleteOne avec le critère de recherche sur l'id.

Résultat attendu : Le produit avec l'id 3 devrait être supprimé de la collection.

```
db.produits.deleteOne({ _id: 3 })  
db.produits.find().pretty()
```

6. Question : Quel est le produit le moins cher dans la collection?

Réponse : Utilisez la méthode find avec l'option de tri sort pour trier par ordre croissant de prix et ajoutez la méthode limit pour obtenir le premier document.

Résultat attendu : Vous devriez obtenir le produit avec le prix le plus bas dans la collection.

```
db.produits.find().sort({ prix: 1 }).limit(1).pretty()
```

7. Question : Comment obtenir la somme totale de la quantité en stock pour tous les produits?

Réponse : Utilisez la méthode aggregate avec le stade \$group pour regrouper et la fonction d'agrégation \$sum.

Résultat attendu : Vous devriez obtenir la somme totale de la quantité en stock pour tous les produits.

```
db.produits.aggregate([
  {
    $group: {
      _id: null,
      quantite_totale: { $sum: "$quantite_stock" }
    }
  }
])
```

8. Question : Comment mettre à jour le prix du produit avec l'id 1 pour l'augmenter de 10 %?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$mul.

Résultat attendu : Le prix du produit avec l'ID 1 devrait être augmenté de 10 %.

```
db.produits.updateOne({ _id: 1 }, { $mul: { prix: 1.1 } })
db.produits.find({_id:1}).pretty()
```

9. Question : Comment obtenir les produits dont le prix est supérieur à la moyenne des prix de tous les produits ?

Réponse : Utilisez la méthode aggregate pour calculer la moyenne des prix, puis \$match pour filtrer.

Résultat attendu : Vous devriez obtenir les produits avec un prix supérieur à la moyenne des prix.

```
var moyenne_prix = db.produits.aggregate([
  {
    $group: {
      _id: null,
      moyenne_prix: { $avg: "$prix" }
    }
  }
]).next().moyenne_prix;

db.produits.find({ prix: { $gt: moyenne_prix } }).pretty()
```

10. Question : Comment obtenir la liste des fournisseurs uniques dans la collection ?

Réponse : Utilisez la méthode distinct pour obtenir les valeurs uniques du champ "fournisseur".

Résultat attendu : Vous devriez obtenir la liste des fournisseurs uniques dans la collection.

```
db.produits.distinct("fournisseur")
```

11. Question : Comment obtenir les produits qui ont une TVA de 19 % et dont le prix est compris entre 300 et 700 ?

Réponse : Utilisez la méthode find avec des opérateurs de comparaison \$eq, \$gte et \$lte :

```
db.produits.find({ tva: 19, prix: { $gte: 300, $lte: 700 } }).pretty()
```

12. Question : Comment ajouter un nouveau produit à la collection "produits"?

Réponse : Utilisez la méthode insertOne pour insérer un document dans la collection.

```
db.produits.insertOne({
  _id: 4,
  designation: "Tablette",
  quantite_stock: 8,
  tva: 19,
  prix: 400,
  fournisseur: "Fournisseur C"
})
```

13. Question : Comment mettre à jour le fournisseur du produit avec l'id 2 en changeant de "Fournisseur B" à "Fournisseur D" ?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$set.

```
db.produits.updateOne({ _id: 2 }, { $set: {fournisseur: "Fournisseur D"}})
db.produits.find({ _id: 2 }).pretty()
```

14. Question : Comment obtenir les produits triés par ordre alphabétique de désignation?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ designation: 1 }).pretty()
```

15. Question : Comment obtenir la liste des produits avec leurs désignations et leurs prix mais exclure l'id ?

Réponse : Utilisez la méthode find avec l'option de projection pour inclure uniquement les champs désirés.

```
db.produits.find({}, { _id: 0, designation: 1, prix: 1 }).pretty()
```

16. Question : Comment mettre à jour le prix de tous les produits ayant une TVA de 20 % en augmentant leur prix de 5 % ?

Réponse : Utilisez la méthode `updateMany` avec l'opérateur `$mul`.

```
db.produits.updateMany({ tva: 20 }, { $mul: { prix: 1.05 } })
db.produits.find().pretty()
```

17. Question : Comment supprimer tous les produits fournis par "Fournisseur A"?

Réponse : Utilisez la méthode `deleteMany` avec le critère de recherche sur le champ "fournisseur".

```
db.produits.deleteMany({ fournisseur: "Fournisseur A" })
db.produits.find().pretty()
```

18. Question : Comment obtenir le produit avec le prix le plus élevé dans la collection ?

Réponse : Utilisez la méthode `find` avec l'option de tri sort en ordre décroissant :

```
db.produits.find().sort({ prix: -1 }).limit(1).pretty()
```

19. Question : Comment ajouter un champ "remise" à tous les produits avec une remise de 10 % ?

Réponse : Utilisez la méthode `updateMany` avec l'opérateur `$set`.

```
db.produits.updateMany({}, { $set: { remise: 0.1 } })
db.produits.find().pretty()
```

20. Question : Comment obtenir les produits triés par ordre de quantité en stock décroissante, mais exclure les produits avec une TVA supérieure à 20 %?

Réponse : Utilisez la méthode `find` avec l'option de tri sort et l'option de projection pour exclure les champs indésirables.

```
db.produits.find({ tva: { $lte: 20 } }).sort({ quantite_stock: -1 }).pretty()
```

21. Question : Comment obtenir la liste des fournisseurs et le nombre de produits qu'ils fournissent?

Réponse : Utilisez la méthode aggregate avec le stade \$group pour regrouper par fournisseur et la fonction d'agrégation \$sum pour compter les produits.

```
db.produits.aggregate([
  {
    $group: {
      _id: "$fournisseur",
      nombre_de_produits: { $sum: 1 }
    }
  }
])
```

22. Question : Comment obtenir les produits avec un prix supérieur à la moyenne des prix des produits fournis par "Fournisseur C" ?

Réponse : Utilisez la méthode aggregate pour calculer la moyenne des prix des produits de "Fournisseur C" puis \$match pour filtrer.

```
var moyenne_prix_fournisseur_c = db.produits.aggregate([
  {
    $match: { fournisseur: "Fournisseur C" }
  },
  {
    $group: {
      _id: null,
      moyenne_prix: { $avg: "$prix" }
    }
  }
]).next().moyenne_prix;

db.produits.find({prix: { $gt: moyenne_prix_fournisseur_c } }).pretty()
```

23. Question : Comment obtenir la liste des produits triés par ordre décroissant de prix, mais exclure les produits avec une quantité en stock inférieure à 8 ?

Réponse : Utilisez la méthode find avec l'option de tri sort et l'option de filtrage \$expr pour les quantités en stock.

```
db.produits.find({ $expr: { $gt: ["$quantite_stock", 8] } }).sort({ prix: -1 }).pretty()
```

{ \$expr: { \$gt: ["\$quantite_stock", 8] } } : L'opérateur \$expr permet d'utiliser des opérateurs d'agrégation dans les requêtes de recherche standard. Ici, l'opérateur \$gt (greater than, signifiant "supérieur à") compare la valeur de la propriété "quantite_stock" de chaque document avec la valeur 8. Cela signifie que seuls les documents ayant une quantité en stock supérieure à 8 seront sélectionnés.

24. Question : Comment mettre à jour le prix des produits de "Fournisseur C" en ajoutant une remise de 15 % ?

Réponse : Utilisez la méthode updateMany avec l'opérateur \$mul et un filtre sur le fournisseur.

```
db.produits.updateMany({ fournisseur: "Fournisseur C" }, { $mul: { prix: 0.85 } })
db.produits.find().pretty()
```

25. Question : Comment supprimer tous les produits avec une remise de 20 % ?

Réponse : Utilisez la méthode deleteMany avec le critère de recherche sur le champ "remise".

```
db.produits.deleteMany({ remise: 0.2 })
db.produits.find().pretty()
```


26. Question : Comment obtenir les produits dont la quantité en stock est supérieure à la moyenne de toutes les quantités en stock ?

Réponse : Utilisez la méthode aggregate pour calculer la moyenne des quantités en stock, puis \$match pour filtrer.

```
var moyenne_quantite = db.produits.aggregate([
  {
    $group: {
      _id: null,
      moyenne_quantite: { $avg: "$quantite_stock" }
    }
  }
]).next().moyenne_quantite;

db.produits.find({ quantite_stock: { $gt: moyenne_quantite } }).pretty()
```

27. Question : Comment obtenir les produits dont la désignation contient le mot "Smartphone" (recherche insensible à la casse) ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex pour une recherche régulière insensible à la casse.

```
db.produits.find({ designation: { $regex: "Smartphone", $options: "i" } }).pretty()
```

28. Question : Comment obtenir le produit avec le prix le plus élevé parmi ceux fournis par "Fournisseur C"?

Réponse : Utilisez la méthode find avec l'option de tri sort et le filtre sur le fournisseur.

```
db.produits.find({ fournisseur: "Fournisseur C" }).sort({ prix: -1 }).limit(1).pretty()
```

29. Question : Comment mettre à jour le champ "prix" du produit avec l'id 2 pour le ramener à 750 ?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$set.

```
db.produits.updateOne({ _id: 2 }, { $set: { prix: 750 } })
db.produits.find({ _id: 2 }).pretty()
```

30. Question : Comment obtenir les produits dont la TVA est de 19 % et le prix est supérieur à 400, triés par ordre croissant de prix ?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et l'option de tri sort.

```
db.produits.find({ tva: 19, prix: { $gt: 400 } }).sort({ prix: 1 }).pretty()
```

31. Question : Comment obtenir les produits dont la désignation commence par "Tablette"?

Réponse : Utilisez la méthode find avec l'opérateur \$regex pour une recherche régulière.

```
db.produits.find({ designation: { $regex: "^Tablette" } }).pretty()
```

32. Question : Comment obtenir les produits avec un prix supérieur à la moyenne des prix des produits fournis par "Fournisseur C" ?

Réponse : Utilisez la méthode aggregate pour calculer la moyenne des prix des produits de "Fournisseur C", puis \$match pour filtrer :

```
var moyenne_prix_fournisseur_c = db.produits.aggregate([
  {
    $match: { fournisseur: "Fournisseur C" }
  },
  {
    $group: {
      _id: null,
      moyenne_prix: { $avg: "$prix" }
    }
  }
]).next().moyenne_prix;

db.produits.find({prix: { $gt: moyenne_prix_fournisseur_c } }).pretty()
```

33. Question : Comment obtenir les produits triés par ordre décroissant de quantité en stock et prix ?

Réponse : Utilisez la méthode find avec l'option de tri sort en utilisant un tableau de champs et de directions.

```
db.produits.find().sort({ quantite_stock: -1, prix: -1 }).pretty()
```

34. Question : Comment ajouter un champ "categorie" avec la valeur "Électronique" à tous les produits ?

Réponse : Utilisez la méthode updateMany avec l'opérateur \$set.

```
db.produits.updateMany({}, { $set: { categorie: "Électronique" } })
db.produits.find().pretty()
```

35. Question : Comment obtenir les produits dont la quantité en stock est supérieure à 10 et la TVA est de 19 %?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et \$and :

```
db.produits.find({ $and: [{ quantite_stock: { $gt: 10 } }, { tva: 19 } ] }).pretty()
```

36. Question : Comment obtenir la liste des fournisseurs avec le nombre de produits qu'ils fournissent, mais exclure les fournisseurs qui ont moins de 3 produits ?

Réponse : Utilisez la méthode aggregate avec les stades \$group et \$match.

```
db.produits.aggregate([
  {
    $group: {
      _id: "$fournisseur",
      nombre_de_produits: { $sum: 1 }
    }
  },
  {
    $match: {
      nombre_de_produits: { $gte: 3 }
    }
  }
])
```

37. Question : Comment obtenir les produits triés par ordre alphabétique décroissant de désignation ?

Réponse : Utilisez la méthode find avec l'option de tri sort et la direction -1.

```
db.produits.find().sort({ designation: -1 }).pretty()
```

38. Question : Comment obtenir le nombre total de produits dans la collection ?

Réponse : Utilisez la méthode count pour compter le nombre de documents dans la collection.

```
db.produits.count()
```

39. Question : Comment supprimer tous les produits dont le prix est inférieur à 300 et la quantité en stock est inférieure à 5 ?

Réponse : Utilisez la méthode deleteMany avec les opérateurs de comparaison

```
db.produits.deleteMany({ prix: { $lt: 300 }, quantite_stock: { $lt: 5 } })
```

40. Question : Comment obtenir les produits dont le fournisseur est "Fournisseur C" ou le prix est supérieur à 500 ?

Réponse : Utilisez la méthode find avec l'opérateur \$or.

```
db.produits.find({ $or: [{ fournisseur: "Fournisseur C" }, { prix: { $gt: 500 } } ] }).pretty()
```

41. Question : Comment obtenir les produits avec un prix supérieur à 400 et une quantité en stock supérieure à 7, triés par ordre croissant de prix ?

Réponse : Utilisez la méthode find avec des opérateurs de comparaison et l'option de tri sort.

```
db.produits.find({ prix: { $gt: 400 }, quantite_stock: { $gt: 7 } }).sort({ prix: 1 }).pretty()
```

42. Question : Comment mettre à jour la TVA de tous les produits ayant une TVA de 19 % pour la changer à 18 % ?

Réponse : Utilisez la méthode updateMany avec l'opérateur \$set.

```
db.produits.updateMany({ tva: 19 }, { $set: { tva: 18 } })
db.produits.find({}).pretty()
```

43. Question : Comment obtenir les produits dont le nom du fournisseur commence par "Fournisseur" ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex pour une recherche régulière.

```
db.produits.find({ fournisseur: { $regex: "^Fournisseur" } }).pretty()
```

44. Question : Comment obtenir la somme totale des prix des produits fournis par "Fournisseur C" ?

Réponse : Utilisez la méthode aggregate avec le stade \$match pour filtrer et le stade \$group pour agréger.

```
db.produits.aggregate([
  {
    $match: { fournisseur: "Fournisseur C" }
  },
  {
    $group: {
      _id: null,
      somme_prix: { $sum: "$prix" }
    }
  }
])
```

45. Question : Comment obtenir les produits triés par ordre décroissant de prix, mais en excluant les produits avec une remise de 10 % ?

Réponse : Utilisez la méthode find avec l'option de tri sort et l'option de filtrage \$ne.

```
db.produits.find({ remise: { $ne: 0.1 } }).sort({ prix: -1 }).pretty()
```

46. Question : Comment obtenir les produits dont la désignation contient le mot "Imprimante" ou "Tablette" ?

Réponse : Utilisez la méthode find avec l'opérateur \$in.

```
db.produits.find({ designation: { $in: ["Imprimante", "Tablette"] } }).pretty()
```

47. Question : Comment obtenir les produits triés par ordre croissant du fournisseur, puis par ordre décroissant de prix ?

Réponse : Utilisez la méthode find avec l'option de tri sort en spécifiant chaque champs de tri et sa direction

```
db.produits.find().sort({ fournisseur: 1, prix: -1 }).pretty()
```

48. Question : Comment obtenir le produit avec le prix le plus bas parmi ceux fournis par "Fournisseur D" ?

Réponse : Utilisez la méthode find avec l'option de tri sort et le filtre sur le fournisseur :

```
db.produits.find({ fournisseur: "Fournisseur D" }).sort({ prix: 1 }).limit(1).pretty()
```

49. Question : Comment mettre à jour la quantité en stock du produit avec l'id 2 pour la doubler ?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$mul.

```
db.produits.updateOne({ _id: 2 }, { $mul: { quantite_stock: 2 } })  
db.produits.find({ _id: 2 }).pretty()
```

50. Question : Comment obtenir les produits triés par ordre alphabétique inversé de fournisseur et par ordre croissant de prix ?

Réponse : Utilisez la méthode find avec l'option de tri sort en spécifiant chaque champs de tri et sa direction.

```
db.produits.find().sort({ fournisseur: -1, prix: 1 }).pretty()
```

51. Question : Comment obtenir les produits avec un prix supérieur à 500 et une TVA de 18 % ou 19 % ?

Réponse : Utilisez la méthode find avec les opérateurs \$and et \$in.

```
db.produits.find({
  $and: [
    { prix: { $gt: 500 } },
    { tva: { $in: [18, 19] } }
  ]
}).pretty()
```

52. Question : Comment obtenir le nombre de produits fournis par chaque fournisseur, trié par ordre décroissant du nombre de produits ?

Réponse : Utilisez la méthode aggregate avec les stades \$group et \$sort.

```
db.produits.aggregate([
  {
    $group: {
      _id: "$fournisseur",
      nombre_de_produits: { $sum: 1 }
    }
  },
  {
    $sort: { nombre_de_produits: -1 }
  }
])
```

53. Question : Comment obtenir les produits triés par ordre croissant de TVA, puis par ordre alphabétique de fournisseur ?

Réponse : Utilisez la méthode find avec l'option de tri sort en spécifiant chaque champs de tri et sa direction.

```
db.produits.find().sort({ tva: 1, fournisseur: 1 }).pretty()
```

54. Question : Comment obtenir le produit avec le prix le plus élevé parmi ceux dont la TVA est de 18 % ?

Réponse : Utilisez la méthode find avec l'option de tri sort et le filtre sur la TVA.

```
db.produits.find({ tva: 18 }).sort({ prix: -1 }).limit(1).pretty()
```

55. Question : Comment mettre à jour la désignation du produit avec l'id 2 pour la changer en "Scanner" ?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$set :

```
db.produits.updateOne({ _id: 2 }, { $set: { designation: "Scanner" } })  
db.produits.find({ _id: 2 }).pretty()
```

56. Question : Comment obtenir les produits avec une quantité en stock supérieure à 5 et une remise inférieure à 0.2 ?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison.

```
db.produits.find({ quantite_stock: { $gt: 5 }, remise: { $lt: 0.2 } }).pretty()
```

57. Question : Comment obtenir le nombre total de produits fournis par "Fournisseur B" ?

Réponse : Utilisez la méthode count avec le critère de recherche sur le champ "fournisseur" :

```
db.produits.count({ fournisseur: "Fournisseur B" })
```

58. Question : Comment obtenir les produits triés par ordre décroissant de prix, mais en mettant la TVA en premier ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ tva: 1, prix: -1 }).pretty()
```


59. Question : Comment supprimer le champ "remise" de tous les produits ?

Réponse : Utilisez la méthode updateMany avec l'opérateur \$unset.

```
db.produits.updateMany({}, { $unset: { remise: "" } })
db.produits.find({}).pretty()
```

60. Question : Comment obtenir les produits avec une TVA de 19 % et un prix supérieur à 600, triés par ordre alphabétique de désignation ?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et l'option de tri sort.

```
db.produits.find({ tva: 18, prix: { $gt: 600 } }).sort({ designation: 1 }).pretty()
```

61. Question : Comment obtenir les produits dont la désignation contient le mot "Tablette" ou "PC" ?

Réponse : Utilisez la méthode find avec l'opérateur \$in :

```
db.produits.find({ designation: { $in: ["Tablette", "PC"] } }).pretty()
```

62. Question : Comment obtenir les produits triés par ordre décroissant de quantité en stock, puis par ordre croissant de prix ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ quantite_stock: -1, prix: 1 }).pretty()
```

63. Question : Comment obtenir le fournisseur qui fournit le plus grand nombre de produits ?

Réponse : Utilisez la méthode aggregate avec les stades \$group, \$sort et \$limit :

```
db.produits.aggregate([
  {
    $group: {
      _id: "$fournisseur",
      nombre_de_produits: { $sum: 1 }
    }
  },
  {
    $sort: { nombre_de_produits: -1 }
  },
  {
    $limit: 1
  }
])
```

64. Question : Comment obtenir les produits triés par ordre alphabétique de fournisseur, puis par ordre croissant de TVA ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ fournisseur: 1, tva: 1 }).pretty()
```

65. Question : Comment mettre à jour la quantité en stock de tous les produits dont le prix est supérieur à 300 pour la diminuer de 2 unités ?

Réponse : Utilisez la méthode updateMany avec l'opérateur \$inc :

```
db.produits.updateMany({ prix: { $gt: 300 } }, { $inc: { quantite_stock: -2 } })
db.produits.find().pretty()
```

66. Question : Comment obtenir les produits dont la désignation contient le mot "Scanner" et la quantité en stock est inférieure à 30 ?

Réponse : Utilisez la méthode find avec l'opérateur \$and et les opérateurs de comparaison

```
db.produits.find({
  $and: [
    { designation: { $regex: "Scanner" } },
    { quantite_stock: { $lt: 30 } }
  ]
}).pretty()
```

67. Question : Comment obtenir les produits triés par ordre alphabétique de fournisseur, puis par ordre décroissant de quantité en stock ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ fournisseur: 1, quantite_stock: -1 }).pretty()
```

68. Question : Comment obtenir la moyenne de prix des produits fournis par "Fournisseur C"?

Réponse : Utilisez la méthode aggregate avec le stade \$match pour filtrer et le stade \$group pour calculer la moyenne.

```
db.produits.aggregate([
  {
    $match: { fournisseur: "Fournisseur C" }
  },
  {
    $group: {
      _id: null,
      moyenne_prix: { $avg: "$prix" }
    }
  }
])
```

69. Question : Comment obtenir les produits triés par ordre décroissant de TVA, puis par ordre alphabétique inversé de fournisseur ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ tva: -1, fournisseur: -1 }).pretty()
```

70. Question : Comment supprimer tous les produits avec une TVA de 20 % ?

Réponse : Utilisez la méthode deleteMany avec le critère de recherche sur le champ "tva".

```
db.produits.deleteMany({ tva: 20 })
db.produits.find().pretty()
```

71. Question : Comment obtenir les produits avec une TVA de 18 % et un prix inférieur à 400, triés par ordre décroissant de quantité en stock ?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et l'option de tri sort.

```
db.produits.find({ tva: 18, prix: { $lt: 400 } }).sort({ quantite_stock: -1 }).pretty()
```

72. Question : Comment obtenir le nombre total de produits avec une TVA de 18 % ?

Réponse : Utilisez la méthode count avec le critère de recherche sur le champ "tva".

```
db.produits.count({ tva: 18 })
```

73. Question : Comment obtenir les produits triés par ordre croissant de fournisseur, puis par ordre alphabétique de désignation ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ fournisseur: 1, designation: 1 }).pretty()
```

74. Question : Comment obtenir la somme totale de la quantité en stock de tous les produits?

Réponse : Utilisez la méthode aggregate avec le stade \$group pour calculer la somme.

```
db.produits.aggregate([
  {
    $group: {
      _id: null,
      somme_quantite: { $sum: "$quantite_stock" }
    }
  }
])
```

75. Question : Comment mettre à jour la désignation du produit avec l'id 2 pour la changer en "Clavier" et la quantité en stock à 50 ?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$set :

```
db.produits.updateOne({ _id: 2 }, { $set: { designation: "Clavier", quantite_stock: 50 } })
db.produits.find({ _id: 2 }).pretty()
```

76. Question : Comment obtenir les produits dont la désignation contient le mot "Clavier" et le prix est compris entre 700 et 800 ?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et l'opérateur \$regex.

```
db.produits.find({
  designation: { $regex: "Clavier" },
  prix: { $gte: 700, $lte: 800 }
}).pretty()
```

77. Question : Comment obtenir les produits triés par ordre alphabétique inversé de fournisseur, puis par ordre croissant de prix ?

Réponse : Utilisez la méthode find avec l'option de tri.

```
db.produits.find().sort({ fournisseur: -1, prix: 1 }).pretty()
```

78. Question : Comment obtenir la somme totale des prix des produits fournis par "Fournisseur C" ?

Réponse : Utilisez la méthode aggregate avec le stade \$match pour filtrer et le stade \$group pour calculer la somme.

```
db.produits.aggregate([
  {
    $match: { fournisseur: "Fournisseur C" }
  },
  {
    $group: {
      _id: null,
      somme_prix: { $sum: "$prix" }
    }
  }
])
```

79. Question : Comment obtenir les produits triés par ordre croissant de TVA, puis par ordre décroissant de fournisseur ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ tva: 1, fournisseur: -1 }).pretty()
```

80. Question : Comment supprimer tous les produits dont le prix est inférieur à 100 et la quantité en stock est inférieure à 3 ?

Réponse : Utilisez la méthode deleteMany avec les opérateurs de comparaison.

```
db.produits.deleteMany({ prix: { $lt: 100 }, quantite_stock: { $lt: 3 } })  
db.produits.find().pretty()
```

81. Question : Comment obtenir les produits dont la désignation commence par "Clav" et la TVA est de 18 % ?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et l'opérateur \$regex.

```
db.produits.find({  
  designation: { $regex: "^Clav" },  
  tva: 18  
}).pretty()
```

82. Question : Comment obtenir les produits triés par ordre croissant de prix, puis par ordre alphabétique de désignation ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ prix: -1, designation: 1 }).pretty()
```

83. Question : Comment obtenir le nombre total de produits avec une TVA de 18 % et fournis par "Fournisseur C" ?

Réponse : Utilisez la méthode `countDocuments` avec le critère de recherche sur les champs "tva" et "fournisseur".

```
db.produits.countDocuments({ tva: 18, fournisseur: "Fournisseur C" })
```

84. Question : Comment obtenir les produits triés par ordre alphabétique de fournisseur, puis par ordre décroissant de quantité en stock ?

Réponse : Utilisez la méthode `find` avec l'option de tri `sort`.

```
db.produits.find().sort({ fournisseur: 1, quantite_stock: -1 }).pretty()
```

85. Question : Comment mettre à jour la TVA de tous les produits ayant une TVA de 18 % pour la changer à 21 % ?

Réponse : Utilisez la méthode `updateMany` avec l'opérateur `$set`.

```
db.produits.updateMany({ tva: 18 }, { $set: { tva: 21 } })
db.produits.find().pretty()
```

86. Question : Comment obtenir les produits dont la désignation contient le mot "Clav" et la quantité en stock est supérieure à 20 ?

Réponse : Utilisez la méthode `find` avec les opérateurs de comparaison et l'opérateur `$regex`.

```
db.produits.find({
  designation: { $regex: "Clav" },
  quantite_stock: { $gt: 20 }
}).pretty()
```

87. Question : Comment obtenir les produits triés par ordre alphabétique inversé de fournisseur, puis par ordre décroissant de prix ?

Réponse : Utilisez la méthode `find` avec l'option de tri `sort`.

```
db.produits.find().sort({ fournisseur: -1, prix: -1 }).pretty()
```

88. Question : Comment obtenir le fournisseur qui fournit le moins de produits ?

Réponse : Utilisez la méthode aggregate avec les stades \$group, \$sort et \$limit.

```
db.produits.aggregate([
  {
    $group: {
      _id: "$fournisseur",
      nombre_de_produits: { $sum: 1 }
    }
  },
  {
    $sort: { nombre_de_produits: 1 }
  },
  {
    $limit: 1
  }
])
```

89. Question : Comment obtenir les produits triés par ordre décroissant du fournisseur, puis par ordre croissant de TVA ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ fournisseur: -1, tva: 1 }).pretty()
```

90. Question : Comment supprimer tous les produits dont la désignation contient le mot "Câble" ?

Réponse : Utilisez la méthode deleteMany avec l'opérateur \$regex pour la recherche dans la désignation :

```
db.produits.deleteMany({ designation: { $regex: "Câble" } })
```

91. Question : Comment obtenir les produits dont la désignation contient le mot "Clav" et le prix est inférieur à 800?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et l'opérateur \$regex.

```
db.produits.find({
  designation: { $regex: "Clav" },
  prix: { $lt: 800 }
}).pretty()
```

92. Question : Comment obtenir les produits triés par ordre croissant de fournisseur, puis par ordre croissant de TVA ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ fournisseur: 1, tva: 1 }).pretty()
```

93. Question : Comment obtenir le fournisseur qui fournit le plus grand nombre de produits ayant une TVA de 21 % ?

Réponse : Utilisez la méthode aggregate avec les stades \$match, \$group, \$sort et \$limit.

```
db.produits.aggregate([
  {
    $match: { tva: 21 }
  },
  {
    $group: {
      _id: "$fournisseur",
      nombre_de_produits: { $sum: 1 }
    }
  },
  {
    $sort: { nombre_de_produits: -1 }
  },
  {
    $limit: 1
  }
])
```

94. Question : Comment obtenir les produits triés par ordre décroissant de fournisseur, puis par ordre croissant de quantité en stock ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ fournisseur: -1, quantite_stock: 1 }).pretty()
```

95. Question : Comment mettre à jour la désignation et le prix du produit avec l'id 2 pour changer respectivement en "Écouteurs" et 25 ?

Réponse : Utilisez la méthode updateOne avec l'opérateur \$set.

```
db.produits.updateOne({ _id: 2 }, { $set: { designation: "Écouteurs", prix: 25 } })  
db.produits.find({ _id: 2 }).pretty()
```

96. Question : Comment obtenir les produits dont la désignation contient le mot "Écouteurs" et la TVA est de 21% ?

Réponse : Utilisez la méthode find avec les opérateurs de comparaison et l'opérateur \$regex.

```
db.produits.find({  
  designation: { $regex: "Écouteurs" },  
  tva: 21  
}).pretty()
```

97. Question : Comment obtenir les produits triés par ordre alphabétique inversé de fournisseur, puis par ordre alphabétique de désignation ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ fournisseur: -1, designation: 1 }).pretty()
```

98. Question : Comment obtenir le nombre total de produits fournis par "Fournisseur D" avec une TVA de 21% ?

Réponse : Utilisez la méthode countDocuments avec le critère de recherche sur les champs "fournisseur" et "tva".

```
db.produits.countDocuments({ fournisseur: "Fournisseur D", tva: 21 })
```

99. Question : Comment obtenir les produits triés par ordre croissant de prix, puis par ordre alphabétique de fournisseur ?

Réponse : Utilisez la méthode find avec l'option de tri sort.

```
db.produits.find().sort({ prix: 1, fournisseur: 1 }).pretty()
```

100. Question : Comment supprimer tous les produits dont la quantité en stock est égale à zero?

Réponse : Utilisez la méthode deleteMany avec l'opérateur de comparaison.

```
db.produits.deleteMany({ quantite_stock: 0 })  
db.produits.find().pretty()
```

101. Question : Comment obtenir les produits dont la désignation commence par "Tab" et contient ensuite n'importe quel caractère ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^Tab.* / } }).pretty()
```

102. Question : Comment obtenir les produits dont le fournisseur contient le mot "sreur" en n'importe quelle position ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ fournisseur: { $regex: /sreur/ } }).pretty()
```

103. Question : Comment obtenir les produits dont la désignation contient au moins un chiffre?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /\d/ } }).pretty()
```

104. Question : Comment obtenir les produits dont la désignation commence par "Tablette" ou "Souris"?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et l'opérateur \$in.

```
db.produits.find({ designation: { $regex: /^(Tablette|Souris)/ } }).pretty()
```

105. Question : Comment obtenir les produits dont la désignation contient exactement huit caractères ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^.{8}$/ } }).pretty()
```

106. Question : Comment obtenir les produits dont la désignation ne contient pas le mot "Tablette" ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et l'opérateur \$not.

```
db.produits.find({ designation: { $not: /Tablette/ } }).pretty()
```

107. Question : Comment obtenir les produits dont la désignation se termine par "lette" ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /lette$/ } }).pretty()
```

108. Question : Comment obtenir les produits dont le fournisseur commence par "Fournisseur" suivi d'un espace et une lettre ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex :

```
db.produits.find({ fournisseur: { $regex: /^Fournisseur [A-Za-z]/ } }).pretty()
```

109. Question : Comment obtenir les produits dont la désignation contient uniquement des lettres ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^[A-Za-z]+$/ } }).pretty()
```

110. Question : Comment obtenir les produits dont le prix est une valeur décimale (ex: 99.99)?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ prix: { $regex: /^\d+\.\d{2}$/ } }).pretty()
```

111. Question : Comment obtenir les produits dont la désignation contient le mot "Carte" suivi de n'importe quel chiffre?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /Carte\d/ } }).pretty()
```

112. Question : Comment obtenir les produits dont la désignation contient uniquement des lettres majuscules ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^[A-Z]+$ / } }).pretty()
```

113. Question : Comment obtenir les produits dont la désignation commence par une lettre majuscule, suivie de lettres minuscules ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^[A-Z][a-z]+ / } }).pretty()
```

114. Question : Comment obtenir les produits dont le fournisseur contient le mot "foUR" en ignorant la casse ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ fournisseur: { $regex: /foUR/i } }).pretty()
```

115. Question : Comment obtenir les produits dont le prix est supérieur à 100 et contient une décimale (ex: 150.99)?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({
  $and: [
    { prix: { $gt: 100 } },
    { prix: { $regex: /^[1-9]\d*\.\d{2}$/ } }
  ]
}).pretty()
```

116. Question : Comment obtenir les produits dont la désignation contient uniquement des chiffres ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et l'option \$not.

```
db.produits.find({ designation: { $not: /\D/ } }).pretty()
```

117. Question : Comment obtenir les produits dont la désignation contient au moins trois voyelles (a, e, i, o, u) ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et une classe de caractères.

```
db.produits.find({ designation: { $regex: /[aeiouAEIOU].*[aeiouAEIOU].*[aeiouAEIOU]/ }
}).pretty()
```

118. Question : Comment obtenir les produits dont le fournisseur contient au moins une lettre ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et une classe de caractères.

```
db.produits.find({ fournisseur: { $regex: /[A-Za-z]/ } }).pretty()
```

119. Question : Comment obtenir les produits dont la désignation contient le mot "Tablette" ou "Tablet" avec une variation de casse ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /tablette|tablet/i } }).pretty()
```

120. Question : Comment obtenir les produits dont le prix est une valeur avec une seule décimale (ex: 49.5) ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ prix: { $regex: /^\d+\.\d$/ } }).pretty()
```

121. Question : Comment obtenir les produits dont la désignation commence par une lettre majuscule suivie de lettres minuscules et se termine par "ette" ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^[A-Z][a-z]+ette$/ } }).pretty()
```

122. Question : Comment obtenir les produits dont le fournisseur est composé de deux mots séparés par un espace ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ fournisseur: { $regex: /^[a-zA-Z]+[a-zA-Z]+$/ } }).pretty()
```

123. Question : Comment obtenir les produits dont la désignation contient au moins deux chiffres consécutifs ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /\d{2}/ } }).pretty()
```

124. Question : Comment obtenir les produits dont le prix est un nombre entier ?

Réponse : Utilisez la méthode find avec les opérateurs \$expr et \$mod.

```
db.produits.updateOne( { _id: 4 }, { $set: { prix: 340.25 } } )  
db.produits.find( { $expr: { $eq: [ { $mod: [ "$prix", 1 ] }, 0 ] } }).pretty()
```

125. Question : Comment obtenir les produits dont la désignation contient au moins une lettre majuscule et une lettre minuscule ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find( { designation: { $regex: /[A-Z]/, $regex: /[a-z]/ } }).pretty()
```

126. Question : Comment obtenir les produits dont le fournisseur commence par "Fournisseur" suivi de deux chiffres ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find( { fournisseur: { $regex: /^Fournisseur\d{2}/ } }).pretty()
```

127. Question : Comment obtenir les produits dont la désignation contient au moins trois chiffres consécutifs ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find( { designation: { $regex: /\d{3}/ } }).pretty()
```

128. Question : Comment obtenir les produits dont la désignation contient au moins un caractère spécial ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et une classe de caractères spéciaux.

```
db.produits.find( { designation: { $regex: /[!@#%&*(),.?:{}|<>]/ } }).pretty()
```

129. Question : Comment obtenir les produits dont le fournisseur contient uniquement des lettres majuscules ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ fournisseur: { $regex: /^[A-Z]+$/ } }).pretty()
```

130. Question : Comment obtenir les produits dont la désignation contient uniquement des chiffres?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et l'option \$not.

```
db.produits.find({ designation: { $regex: /\d+$/ } }).pretty()
```

131. Question : Comment obtenir les produits dont la désignation commence par "Chaise" ou "Table" suivi de n'importe quel caractère ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^(Chaise|Table).*/ } }).pretty()
```

132. Question : Comment obtenir les produits dont la désignation contient au moins une lettre majuscule suivie d'au moins deux lettres minuscules ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find({ designation: { $regex: /^[A-Z][a-z]{2,}/ } }).pretty()
```

133. Question : Comment obtenir les produits dont le prix est supérieur à 25 mais ne contient pas la décimale ?

Réponse : Utilisez la méthode find avec l'opérateur \$expr.

```
db.produits.updateOne( { _id: 4 }, { $set: { prix: 340 } } )
db.produits.find({
  $and: [
    { prix: { $gt: 25 } },
    { $expr: { $eq: [{ $mod: ["$prix", 1] }, 0] } }
  ]
}).pretty()
```

134. Question : Comment obtenir les produits dont la désignation contient au moins une lettre majuscule, une lettre minuscule et un chiffre ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et des classes de caractères.

```
db.produits.updateOne( { _id: 4 }, { $set: { designation: "taBlette44" } } )  
db.produits.find( { designation: { $regex: /[A-Z]/, $regex: /[a-z]/, $regex: /\d/ } } ).pretty()
```

135. Question : Comment obtenir les produits dont la désignation contient exactement trois caractères, où les deux premiers sont des lettres et le dernier est un chiffre ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.updateOne( { _id: 4 }, { $set: { designation: "Hp5" } } )  
db.produits.find( { designation: { $regex: /^[A-Za-z]{2}\d$/ } } ).pretty()
```

136. Question : Comment obtenir les produits dont la désignation contient au moins un caractère spécial (excluant les lettres et les chiffres) ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex et une classe de caractères spéciaux.

```
db.produits.find( { designation: { $regex: /^[^A-Za-z0-9]/ } } ).pretty()
```

137. Question : Comment obtenir les produits dont le fournisseur contient au moins trois lettres minuscules consécutives ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.find( { fournisseur: { $regex: /[a-z]{3}/ } } ).pretty()
```

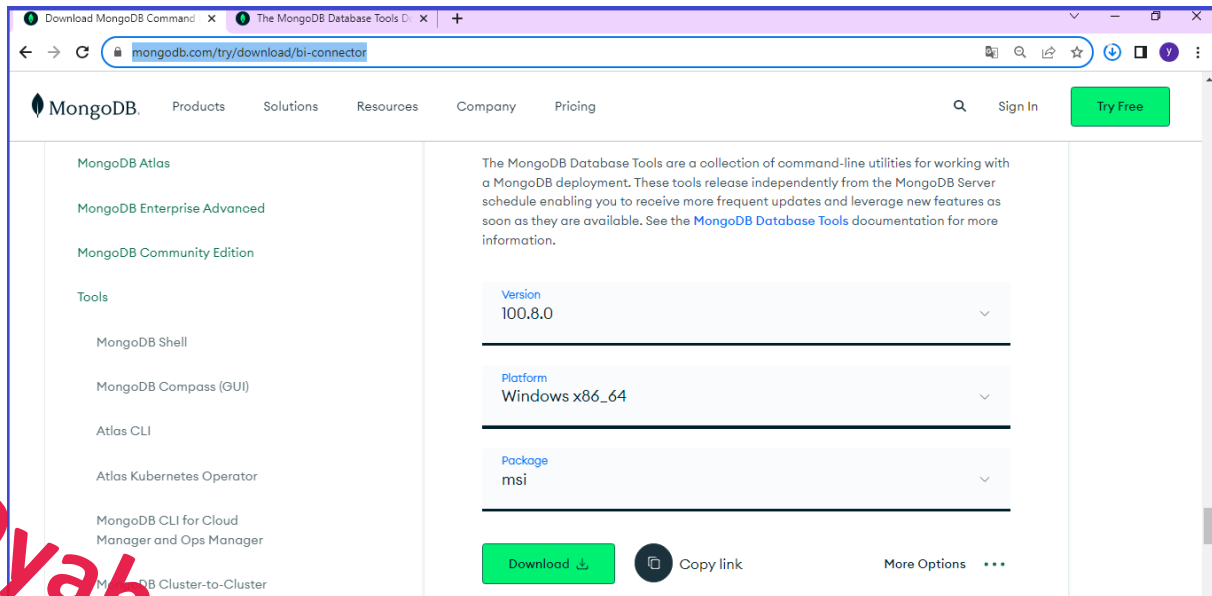
138. Question : Comment obtenir les produits dont la désignation contient au moins deux chiffres consécutifs ?

Réponse : Utilisez la méthode find avec l'opérateur \$regex.

```
db.produits.updateOne( { _id: 4 }, { $set: { designation: "taBlette44" } } )  
db.produits.find( { designation: { $regex: /\d{2}/ } } ).pretty()
```

139. Question : Comment exporter la collection "produits" au format JSON ?

Télécharger et installer MongoDB Command Line Database Tools en se connectant à MongoDB Download Center : <https://www.mongodb.com/try/download/bi-connector>



Ajouter le chemin du fichier **mongoexport** : **C:\Program Files\MongoDB\Tools\100\bin** au path de la variable d'environnement.

Réponse : Utilisez la commande **mongoexport** au terminal (en dehors du mongo shell) pour exporter la collection au format JSON.

```
mongoexport --db ma_base_de_donnees --collection produits --out c:\exportfolder\produits.json
```

140. Question : Comment importer à nouveau la collection "produits" à partir du fichier JSON?

Réponse : Utilisez la commande **mongoimport** pour importer la collection depuis le fichier JSON.

```
mongoimport --db ma_base_de_donnees --collection produits2 --file c:\exportfolder\produits.json
```

Exécuter l'instruction mongo shell suivante pour vérifier la création d'une nouvelle collection **produits2**.

```
show collections
```

141. Question : Comment exporter la collection "produits" au format CSV ?

Réponse : Utilisez la commande mongoexport pour exporter la collection au format CSV.

```
mongoexport --db ma_base_de_donnees --collection produits --type=csv --fields id,designation,quantite_stock,tva,prix,fournisseur --out c:\exportfolder\produits.csv
```

142. Question : Comment importer à nouveau la collection "produits" à partir du fichier CSV?

Réponse : Utilisez la commande mongoimport pour importer la collection depuis le fichier CSV.

```
mongoimport --db ma_base_de_donnees --collection produits3 --type csv --headerline --file c:\exportfolder\produits.csv
```

143. Question : Comment exporter la collection "produits" au format BSON ?

Réponse : Utilisez la commande mongodump pour exporter la base de données au format BSON.

```
mongodump --db ma_base_de_donnees --collection produits --out c:\exportfolder
```

144. Question : Comment importer à nouveau la collection "produits" à partir du fichier BSON?

Réponse : Utilisez la commande mongorestore pour restaurer la base de données depuis le fichier BSON.

```
mongorestore --db ma_base_de_donnees --collection produits4 c:\exportfolder\ma_base_de_donnees\produits.bson
```

145. Question : Comment exporter la collection "produits" au format JSON avec des données limitées à celles dont la TVA est de 20 %?

Réponse : Utilisez la commande mongoexport avec l'option --query pour spécifier la requête de filtrage.

Exécuter la commande mongo shell suivante :

```
db.produits.updateOne( { _id: 4 }, { $set: { tva: 20 } } )
```

Puis la commande suivante en dehors de mongo shell :

```
mongoexport --db ma_base_de_donnees --collection produits --query "{ \"tva\": 20 }" --out c:\exportfolder\produits_tva20.json
```

146. Question : Comment importer à nouveau la collection "produits" avec des données filtrées depuis le fichier JSON ?

Réponse : Utilisez la commande mongoimport pour importer la collection filtrée depuis le fichier JSON

```
mongoimport --db ma_base_de_donnees --collection produits6 --file c:\exportfolder\produits_tva20.json
```

147. Question : Comment créer un index sur la collection "produits" pour accélérer les recherches de désignations ?

Réponse : Utilisez la méthode createIndex pour créer un index textuel sur le champ "designation".

```
db.produits.createIndex({ designation: "text" })
```

148. Question : Comment rechercher rapidement les produits dont la désignation contient le mot "Écouteurs" ?

Réponse : Utilisez l'opérateur \$text dans une requête de recherche pour profiter de l'index textuel.

```
db.produits.find({ $text: { $search: "Écouteurs" } }).pretty()
```

149. Question : Comment créer un index sur la collection "produits" pour accélérer les requêtes de recherche par prix?

Réponse : Utilisez la méthode createIndex pour créer un index ascendant sur le champ "prix".

```
db.produits.createIndex({ prix: 1 })
```

150. Question : Comment créer un utilisateur avec un nom d'utilisateur et un mot de passe dans MongoDB?

Réponse : Utilisez la méthode createUser pour créer un utilisateur avec des informations d'identification.

```
db.createUser({  
  user: "utilisateur",  
  pwd: "motdepasse",  
  roles: ["readWrite"]  
})
```

Le rôle "readWrite". Le rôle "readWrite" accorde à cet utilisateur le privilège de lire et d'écrire des données dans la base de données spécifiée, mais il est important de noter que ce rôle ne permet pas à l'utilisateur de créer de nouvelles collections.

151. Question : Comment accorder des privilèges d'écriture et de lecture à un utilisateur déjà créé sur la collection "produits" ?

Réponse : Utilisez la méthode `grantRolesToUser` pour attribuer des rôles spécifiques à un utilisateur.

```
db.grantRolesToUser("utilisateur", [
  { role: "readWrite", db: "ma_base_de_donnees", collection: "produits" }
])
```

152. Question : Comment retirer les privilèges d'écriture à un utilisateur sur la collection "produits"?

Réponse : Utilisez la méthode `revokeRolesFromUser` pour supprimer des rôles d'un utilisateur.

```
db.revokeRolesFromUser("utilisateur", [
  { role: "readWrite", db: "ma_base_de_donnees", collection: "produits" }
])
```

153. Question : Comment supprimer un utilisateur de MongoDB ?

Réponse : Utilisez la méthode `dropUser` pour supprimer un utilisateur existant, MongoDB supprimera automatiquement les privilèges associés.

```
db.dropUser("utilisateur")
```

154. Question : Comment créer un rôle personnalisé dans MongoDB avec des privilèges spécifiques ?

Réponse : Utilisez la méthode `createRole` pour créer un rôle personnalisé avec des privilèges spécifiques.

```
db.createRole({
  role: "lecteurProduits",
  privileges: [
    { resource: { db: "ma_base_de_donnees", collection: "produits" }, actions: ["find"] }
  ],
  roles: []
})
```

155. Question : Comment accorder un rôle personnalisé à un utilisateur dans MongoDB?

Réponse : Utilisez la méthode `grantRolesToUser` pour attribuer un rôle personnalisé à un utilisateur.

```
db.createUser({
  user: "utilisateur",
  pwd: "motdepasse",
  roles: []
})
db.grantRolesToUser("utilisateur", ["lecteurProduits"])
```

156. Question : Comment afficher les rôles attribués à un utilisateur spécifique ?

Réponse : Utilisez la méthode `getUser` pour afficher les rôles d'un utilisateur.

```
db.getUser("utilisateur")
```

157. Question : Comment supprimer un rôle personnalisé dans MongoDB ?

Réponse : Utilisez la méthode `dropRole` pour supprimer un rôle personnalisé existant.

```
db.dropRole("lecteurProduits")
```

158. Question : Comment lister tous les utilisateurs et leurs rôles dans MongoDB ?

Réponse : Utilisez la méthode `getUsers` pour afficher la liste des utilisateurs et de leurs rôles.

```
db.getUsers()
```

159. Question : Comment lister tous les rôles disponibles dans MongoDB ?

Réponse : Utilisez la méthode `getRoles` pour afficher la liste de tous les rôles disponibles.

```
db.getRoles({ showBuiltinRoles: true })
```

160. Question : Comment un utilisateur se connecte à une base de donnée MongoDB ?

Réponse : Utilisez la méthode `auth`.

```
use ma_base_de_donnees
db.auth("utilisateur", "motdepasse")
```

161. Question : Comment changer le mot de passe pour un utilisateur existant dans MongoDB ?

Réponse : Utilisez la méthode `changeUserPassword` pour changer le mot de passe d'un utilisateur :

```
db.changeUserPassword("utilisateur", "nouveauMotdepasse")
db.auth("utilisateur", "nouveauMotdepasse")
```

162. Question : Comment limiter l'accès en lecture seule à la collection `produits` pour l'utilisateur `"utilisateur"`?

Réponse : Créez un rôle personnalisé en donnant le privilège de lecture à la collection `produits` et attribuez ce rôle à l'utilisateur `"utilisateur"`.

```
db.createRole({
  role: "lectureProduits",
  privileges: [
    { resource: { db: "ma_base_de_donnees", collection: "produits" }, actions: ["find"] }
  ],
  roles: []
})

db.grantRolesToUser("utilisateur", ["lectureProduits"])
db.getUser("utilisateur")
```

163. Question : Comment empêcher l'utilisateur `"utilisateur"` d'accéder à la collection `"produits"` dans MongoDB?

Réponse : Créez un rôle personnalisé sans aucun privilège sur la collection "produits" et attribuez ce rôle à l'utilisateur "utilisateur".

```
db.revokeRolesFromUser("utilisateur", [
  { role: "lectureProduits", db: "ma_base_de_donnees", collection: "produits" }
])

db.createRole({
  role: "aucunAccesProduits",
  privileges: [],
  roles: []
})

db.grantRolesToUser("utilisateur", ["aucunAccesProduits"])
db.getUser("utilisateur")
```

164. Question : Comment afficher les privilèges du rôle "aucunAccesProduits" dans MongoDB?

Réponse : Utilisez la méthode `getRole` pour afficher les privilèges du rôle "aucunAccesProduits".

```
db.getRole("aucunAccesProduits")
```

165. Question : Créer la collection clients avec les propriétés `_id`, `nom`, `prénom`, `ville` et `email`.

Réponse : Utilisez la méthode insertMany pour ajouter plusieurs documents à la collection clients.

```
db.clients.insertMany([
  {
    _id: 1,
    nom: 'Nom1',
    prenom: 'Prenom1',
    ville: 'Ville1',
    email: 'email1@example.com'
  },
  {
    _id: 2,
    nom: 'Nom2',
    prenom: 'Prenom2',
    ville: 'Ville2',
    email: 'email2@example.com'
  },
  {
    _id: 3,
    nom: 'Nom3',
    prenom: 'Prenom3',
    ville: 'Ville3',
    email: 'email3@example.com'
  }
])
```

166. Question : Comment créer un rôle personnalisé avec des privilèges sur les deux collections produits et clients dans MongoDB ?

Réponse : Créez un rôle personnalisé en spécifiant les privilèges pour chaque collection :

```
db.createRole({
  role: "lecteurProduitsCommandes",
  privileges: [
    { resource: { db: "ma_base_de_donnees", collection: "produits" }, actions: ["find"] },
    { resource: { db: "ma_base_de_donnees", collection: "clients" }, actions: ["find"] }
  ],
  roles: []
})
```