Base de données de documents

Au format Json sans structure prédéfinie

Schema

Document est un ensemble de donnés cle-valeur = row

En mongo on peut stocker ilimite dans la Collection est un grp de doc = table

Mongo est un système de bdd crossplateform

Possibilite de tableau mais bcp d’info dupliqué

Field = colonne

Jointure = documents imbriqués

Avantage : structure simple et claire en objet

Fait pour la monter en charge

Plus besoin d’orm , stockage sur disque

L’indexation simple :

‘’’

Db.collection.createIndex(<champ\_et\_type>, <options>)

db.personnes.createIndex({"age":-1})

‘’’

Pour consulter les index d’une collection :

‘’’’

Db.collection.getIndexes()

‘’’

Supprimer index :

‘’’

Db.collection.dropIndex(‘age\_-1 ‘ )

Db.collection.createIndex({age:-1},{name: “unsupernom”})

‘’’

Tableaux

‘’’

{$push: {<champs>: <Valeur>, ….)

‘’’

L’opérateur push permet d’ajouter une ou plusieurs valeurs au sein d’un tableau

Ajout d’une passion :



Il est possible d’ajouter plusieurs valeur égales /il n’y a pas de verif de doublons.

‘’’

Db.hoobies.updateOne({« \_id » :2}, $push :{passions :{$each :[“Minecraft “, “rise”]}})

‘’’

Remplacer $push par $addToSet pour éviter les doublons

Exercices :

Créez une base de données sample nommée "sample\_db" et une collection appelée "employees". Insérez les documents suivants dans la collection "employees":

{ name: "John Doe", age: 35, job: "Manager", salary: 80000 }

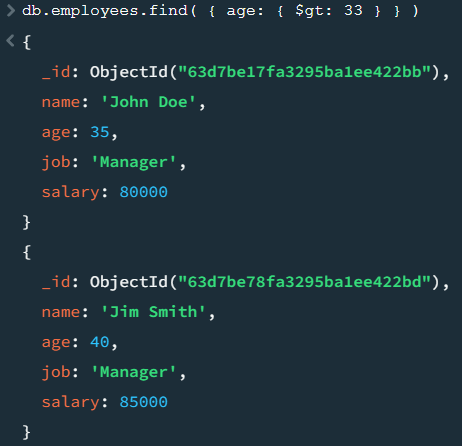
{ name: "Jane Doe", age: 32, job: "Developer", salary: 75000 }

{ name: "Jim Smith", age: 40, job: "Manager", salary: 85000 }

Écrivez une requête MongoDB pour trouver tous les documents dans la collection "employees".

Une image contenant texte

Description générée automatiquement

Écrivez une requête pour trouver tous les documents où l'âge est supérieur à 33.

Écrivez une requête pour trier les documents dans la collection "employees" par

salaire décroissant.

Une image contenant texte

Description générée automatiquement

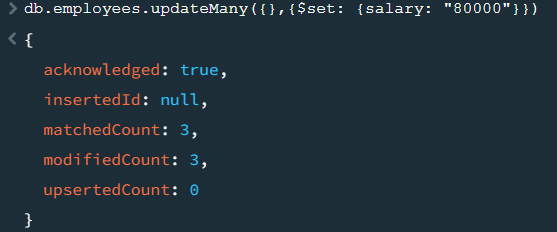
Écrivez une requête pour sélectionner uniquement le nom et le job de chaque document.

Une image contenant texte

Description générée automatiquement

Écrivez une requête pour compter le nombre d'employés par poste.

Écrivez une requête pour mettre à jour le salaire de tous les développeurs à 80000.



Exercice 1

Affichez l’identifiant et le nom des salles qui sont des SMAC.

Une image contenant texte

Description générée automatiquement

Exercice 2

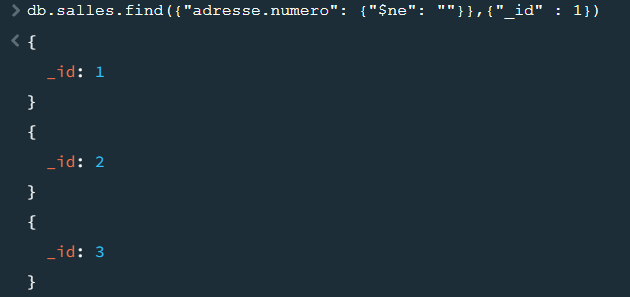
Affichez le nom des salles qui possèdent une capacité d’accueil strictement supérieure à 1000 places.

Une image contenant texte

Description générée automatiquement

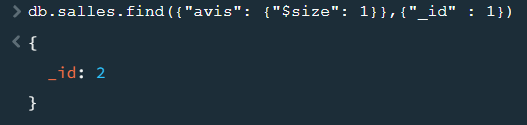
Exercice 3

Affichez l’identifiant des salles pour lesquelles le champ adresse ne comporte pas de numéro.



Exercice 4

Affichez l’identifiant puis le nom des salles qui ont exactement un avis.



Exercice 5

Affichez tous les styles musicaux des salles qui programment notamment du blues.

Une image contenant texte

Description générée automatiquement

Exercice 6

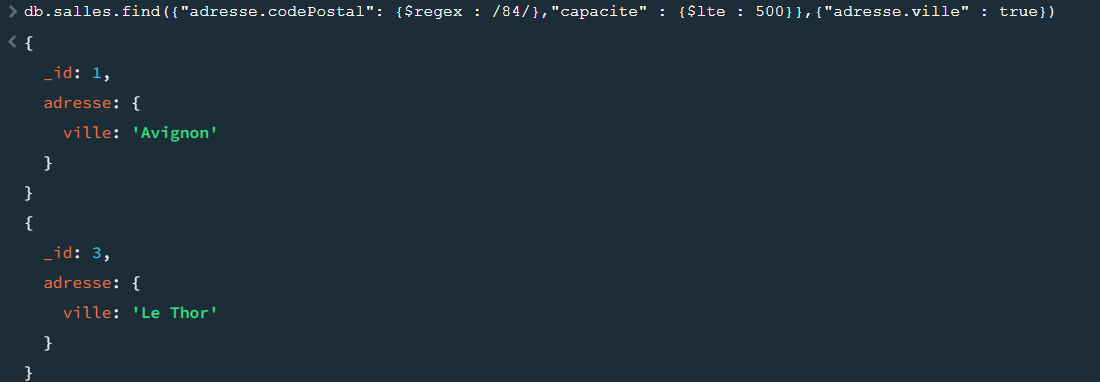
Affichez tous les styles musicaux des salles qui ont le style « blues » en première position dans leur tableau styles.

Une image contenant texte

Description générée automatiquement

Exercice 7

Affichez la ville des salles dont le code postal commence par 84 et qui ont une capacité strictement inférieure à 500 places (pensez à utiliser une expression régulière).



Exercice 8

Affichez l’identifiant pour les salles dont l’identifiant est pair ou le champ avis est absent.



Exercice 9

Affichez le nom des salles dont au moins un des avis comporte une note comprise entre 8 et 10 (tous deux inclus).Une image contenant texte

Description générée automatiquement

Exercice 10

Affichez le nom des salles dont au moins un des avis comporte une date postérieure au 15/11/2019 (pensez à utiliser le type JavaScript Date).

Une image contenant texte

Description générée automatiquement

Exercice 11

Affichez le nom ainsi que la capacité des salles dont le produit de la valeur de l’identifiant par 100 est strictement supérieur à la capacité.



Exercice 12

Affichez le nom des salles de type SMAC programmant plus de deux styles de musiques différents en utilisant l’opérateur $where qui permet de faire usage de JavaScript.

Exercice 13

Affichez les différents codes postaux présents dans les documents de la collection salles.



Exercice 14

Mettez à jour tous les documents de la collection salles en rajoutant 100 personnes à leur capacité actuelle.

Une image contenant texte

Description générée automatiquement

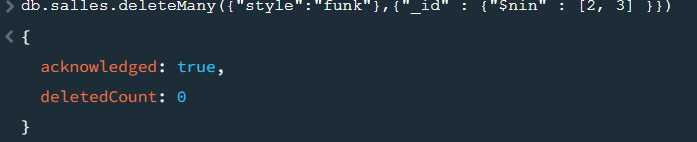
Exercice 15

Ajoutez le style « jazz » à toutes les salles qui n’en programment pas.



Exercice 16

Retirez le style «funk» à toutes les salles dont l’identifiant n’est égal ni à 2, ni à 3.



Exercice 17

Ajoutez un tableau composé des styles «techno» et « reggae » à la salle dont l’identifiant est 3.

Une image contenant texte

Description générée automatiquement

Exercice 18

Pour les salles dont le nom commence par la lettre P (majuscule ou minuscule), augmentez la capacité de 150 places et rajoutez un champ de type tableau nommé contact dans lequel se trouvera un document comportant un champ nommé telephone dont la valeur sera « 04 11 94 00 10 ».

Exercice 19

Pour les salles dont le nom commence par une voyelle (peu importe la casse, là aussi), rajoutez dans le tableau avis un document composé du champ date valant la date courante et du champ note valant 10 (double ou entier). L’expression régulière pour chercher une chaîne de caractères débutant par une voyelle suivie de n’importe quoi d’autre est [^aeiou]+$.

Exercice 20

En mode upsert, vous mettrez à jour tous les documents dont le nom commence par un z ou un Z en leur affectant comme nom « Pub Z », comme valeur du champ capacite 50 personnes (type entier et non décimal) et en positionnant le champ booléen smac à la valeur « false ».

Exercice 21

Affichez le décompte des documents pour lesquels le champ \_id est de type « objectId ».

Exercice 22

Pour les documents dont le champ \_id n’est pas de type « objectId », affichez le nom de la salle ayant la plus grande capacité. Pour y parvenir, vous effectuerez un tri dans l’ordre qui convient tout en limitant le nombre de documents affichés pour ne retourner que celui qui comporte la capacité maximale.

Exercice 23

Remplacez, sur la base de la valeur de son champ \_id, le document créé à l’exercice 20 par un document contenant seulement le nom préexistant et la capacité, que vous monterez à 60 personnes.

Exercice 24

Effectuez la suppression d’un seul document avec les critères suivants : le champ \_id est de type « objectId » et la capacité de la salle est inférieure ou égale à 60 personnes.

Exercice 25

À l’aide de la méthode permettant de trouver un seul document et de le mettre à jour en même temps, réduisez de 15 personnes la capacité de la salle située à Nîmes.

Exercice Index :

db.salles.find({"capacite": {$gt: 500}, "adresse.codePostal": /^30/})

db.salles.find({"adresse.codePostal": /^30/, "capacite": {$lte: 400}})

Je propose deux index pour couvrir ces deux requêtes : sur le codePostal et sur la capacite

Db.salles.createIndex({‘adresse.codePostal’:1})

Db.salles.createIndex({‘capacite’ :1})

Pour supprimer les index :

Db.salles.dropIndex()