

Les index et les agrégats

Gestion des posts d'un blog

Créez une nouvelle collection nommée posts. Elle contient des documents de cette forme :

```
{
  createdAt: Date,
  author: String,
  title: String,
  content: String,
  comments: [{
    date: Date,
    writer: String,
    role: String,
    msg: String
  }]
}
```

Ajoutez les documents suivants :

```
{
  createdAt: new Date("2016-01-18"),
  author: "Loutre joyeuse",
  title: "D'accord",
  content: "Mais",
  comments: [{
    date: new Date("2016-01-20"),
    writer: "Tortue géniale",
    msg: "Pas d'accord"
  }]
}

{
  createdAt: new Date("2016-01-18"),
  author: "Bison futé",
  title: "d'accord",
  content: "Ou",
  comments: [{
    date: new Date("2016-01-20"),
    writer: "Tortue géniale",
    msg: "Pas d'accord"
  }, {
    date: new Date("2016-01-21"),
    writer: "Loutre joyeuse",
    msg: "D'accord"
  }]
}

{
  createdAt: new Date("2016-03-18"),
  author: "Loutre joyeuse",
  title: "Pas d'accord",
  content: "Et",
  comments: [{
    date: new Date("2016-03-20"),
    writer: "Tortue géniale",
    msg: "Pas d'accord"
  },
  {
```

```
        date: new Date("2016-03-21"),
        writer: "Loutre joyeuse",
        msg: "Pas d'accord"
    }]
}

{
    createdAt: new Date("2017-01-18"),
    author: "Tortue géniale",
    title: "Pas d'accord",
    content: "Donc"
}

{
    createdAt: new Date("2017-01-18"),
    author: "Loutre joyeuse",
    title: "Pas d'accord",
    content: "Or",
    comments: [{
        date: new Date("2017-01-20"),
        writer: "Bison futé",
        msg: "D'accord"
    },
    {
        date: new Date("2017-01-21"),
        writer: "Tortue géniale",
        msg: "Pas d'accord"
    },
    {
        date: new Date("2017-01-22"),
        writer: "Loutre joyeuse",
        msg: "Pas d'accord"
    }
    ]
}
```

Index

Exercice 1

Ajoutez un index sur le champ author.

Exercice 2

Ajoutez un index décroissant sur le champ createdAt.

Exercice 3

Ajoutez un index pour avoir title dans l'ordre alphabétique et le createdAt du plus récent au plus ancien. Faites une recherche sur les posts ayant 'Pas d'accord' comme titre.

Exercice 4

Affichez tous vos index.

Agrégation

Exercice 1

Ecrivez la requête pour connaître le nombre de posts par auteur (ne pas prendre en compte les commentaires).

Exercice 2

Regroupez les posts par auteur et affichez, pour chaque auteur, les dates de ses posts ainsi que les titres.

Exercice 3

Comptez le nombre de posts par mois.

Pour cette requête vous pouvez extraire le mois d'une date de la façon suivante : `$month: "$createdAt"`

Ca fonctionne aussi pour l'année.

Pensez aussi que `_id` peut aussi être un objet js.

Exercice 4

Comptez le nombre de posts pour l'auteur 'Loutre joyeuse'.

Exercice 5

Grâce au pipeline `$unwind`, vous pouvez éclater un tableau en documents.

Testez cette requête : `db.posts.aggregate({$unwind: "$comments"});`

Exercice 6

Compter le nombre de commentaires par post.

Exercice 7

Compter le nombre de commentaires par auteur.

Exercice 8

Cette fois-ci, votre requête va permettre d'avoir tous les articles de 'Loutre joyeuse', trié par date du plus récent au plus ancien, avec les commentaires du plus ancien au plus récent.