

Projet NoSQL

Base de données de statistiques d'une ligue de basket

I. Introduction

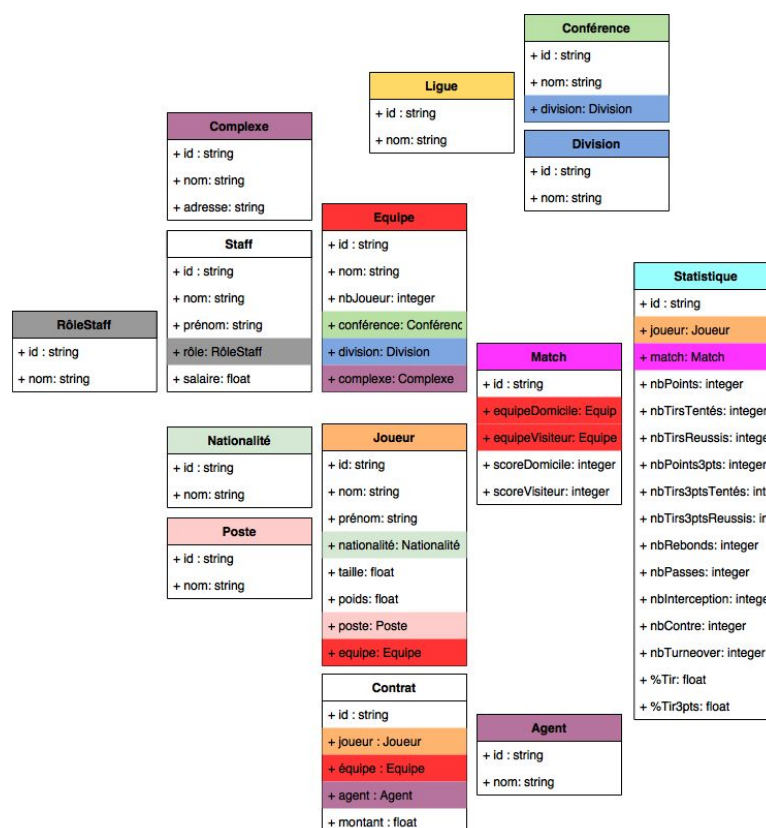
Le contexte pour l'utilisation de notre base de données est le suivant, une application mobile permettant de retrouver toutes les statistiques d'une ligue de basket depuis plusieurs années comprenant statistiques des matchs, des joueurs... .

Notre volonté première a donc été de créer une base de données pouvant stocker toutes ces données statistiques de la ligue de basket, ceci passant par les équipes, les joueurs, mais aussi et surtout les matchs induisant les statistiques du match et des joueurs.

Dans un soucis d'évolution de cette dernière nous avons voulu la rendre scalable, ainsi nous pouvons ajouter autant de ligues que nous le souhaitons, les autres type de données découlant des nouvelles ligues ajoutées.

Ce souhait de scalabilité nous avons pu y répondre grâce à l'utilisation d'une base de données NoSQL.

II. Schéma



III. Choix technique

Parmis les différentes catégories de base de données à notre disposition telle que orientées document, clé/valeur, en colonnes, orientées graphes, nous avons choisit de partir sur une base de données orientées documents.

Cette solution a pour habitude d'être utilisée pour des systèmes de gestion de contenus, permettant de collecter et traiter des documents notamment pour des applications mobiles ou Web ayant un fort trafic ce qui correspond totalement à notre cas.

Ainsi on peut retrouver des collections au format JSON de ce type référençant des documents :

en haut à gauche League collection,
en haut à droite Team collection,
en bas Player collection

```
{
  "Leagues": [
    {
      "Id": ObjectID(5sfsf546sd21),
      "Name": "NBA",
    },
    {
      "Id": ObjectID(5sfsf546sd22),
      "Name": "Pro A",
    },
    {
      "Id": ObjectID(5sfsf546sd23),
      "Name": "Pro B",
    },
    {
      "Id": ObjectID(5sfsf546sd23),
      "Name": "EuroLeague",
    },
  ],
}

{
  "Teams": [
    {
      "Id": ObjectID(5sfsf785azl2),
      "Name": "Celtics",
      "Players": [
        ObjectID(5sfsf77878456), <-- tableau de Player
        ObjectID(fg65d4g597sf), <-- référence à un objet Player
        ObjectID(5sfsf7845sff2),
        ObjectID(fg546df897dsg4),
        ObjectID(5sfsf77865d5gl2),
        ObjectID(5sfsf7845sff2),
        ObjectID(5sfsf7qsf78429),
        ObjectID(5d6fg465f4g),
        ObjectID(5sfsf7845sff2),
        ObjectID(5sfsf7qdgdsf78429),
        ObjectID(5sfsf77845l2),
        ObjectID(5sfsf7845sff2),
        ObjectID(56445d6fg4rfqdg),
      ],
      "Conference": ObjectID(5sfsf7784784),
      "Division": ObjectID(5sfsf778456),
    },
  ],
}

{
  "Player": [
    {
      "Id": ObjectID(5sfsf7854d23),
      "Lastname": "James",
      "Firstname": "LeBron",
      "Nationality": ObjectID(5sfsf7878gh89), <-- référence à un objet Nationality
      "height": 205,
      "weight": 95,
      "position": ObjectID(5sfsf7856df1), <-- référence à un objet Position
      "team": ObjectID(5sfsf78jhf12), <-- référence à un objet Team
    },
    {
      "Id": ObjectID(5sfsf7854d23),
      "Lastname": "Irving",
      "Firstname": "Kyrie",
      "Nationality": ObjectID(5sfsf7878gh89), <-- référence à un objet Nationality
      "height": 195,
      "weight": 88,
      "position": ObjectID(5sfsf7856df5), <-- référence à un objet Position
      "team": ObjectID(5sfsf78jhf45), <-- référence à un objet Team
    },
  ],
}
```

On peut voir très nettement l'avantage de la base de données orientées documents, notamment sur la collection "Team" en effet une équipe est composée de joueur ceux ci stockés sous forme de tableaux "Players" et donc stockants des références à des objets Player.

Concernant le choix de la technologie NoSQL, nous avons préféré MongoDB à d'autre, pour sa souplesse et son accessibilité, sa documentation plutôt riche et dense ainsi que sa réponse à la plupart des besoins.

IV. Conclusion

Pour conclure notre choix d'une base de données orientées documents nous permet de garder une cohérence au niveau structurel des données ainsi qu'une persistance de ces dernières, elle permet aussi une scalabilité plus aisée notamment si l'on souhaite l'ajout d'une nouvelle ligue dans notre cas.