

Structures de données pour le Cloud

Rapport 4

Auteurs:

Roland DENIZOT

Sébastien YUNG

Hugo RAVÉ

Amine MOUSSA



Contents

1	Introduction	2
2	Importation des données	2
2.1	Schémas	2
2.2	Génération de la dénormalisation	3
2.3	Importation de la base de données en SSH	4
2.4	Statistiques en sharding	6
3	Mesure de performances	11
3.1	Requête 1	11
3.2	Requête 2	12
3.3	Requête 3	12
3.4	Requête 4	13
3.5	Requête 5	13
3.6	Requête 6	14
3.7	Requête 7	14
3.8	Requête 8	15
3.9	Conclusion sur les performances	15
4	Vues	16
4.1	Vues utilisateur standard	16
4.2	Vues Analyste/Décisionnaire	17
4.3	Vue Administrateur	18

1 Introduction

Dans le cadre de la dernière partie du projet du cours "Structures de données pour le Cloud", nous allons appliquer la dénormalisation de la base données puis nous allons importer cette dernière dans un MongoDB sur machines virtuelles. Nous allons tester plusieurs configurations de sharding afin de constater l'évolution du temps de réponse des requêtes ainsi que proposer une application cloud pour des utilisateurs. Le projet entier est disponible sur GitHub en suivant [ce lien](#).

2 Importation des données

2.1 Schémas

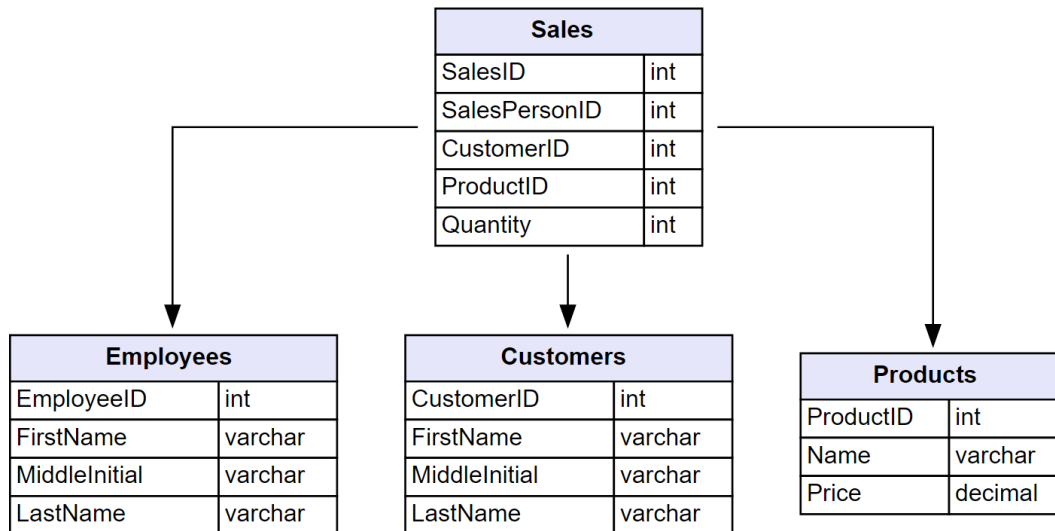


Figure 1: Schéma de notre base de données initiale



Figure 2: Schéma de notre base de données dénormalisée

2.2 Génération de la dénormalisation

Afin d'appliquer la dénormalisation sur la base de données initiale, nous avons exécuté le programme suivant en Python (disponible sur GitHub en suivant [ce lien](#)):

```
import pandas as pd

customers = pd.read_csv("SalesDB/customers.csv")
sales = pd.read_csv("SalesDB/sales.csv")
employees = pd.read_csv("SalesDB/employees.csv")
products = pd.read_csv("SalesDB/products.csv")

final_table = (
    sales
    .merge(products, on="ProductID", how="inner")
    .merge(customers, on="CustomerID", how="inner")
    .merge(
        employees,
        left_on="SalesPersonID",
        right_on="EmployeeID",
        how="left",
        suffixes=["_customer", "_employee"]
    )
    .sort_values(by="SalesID")
    .reset_index(drop=True)
)

final_table = final_table.drop("SalesPersonID", axis=1)

final_table.to_json("Unique_table.json", orient="records")
```

Il s'agit de charger toutes les tables et d'effectuer un left join grâce aux clés primaires de chaque table qui sont contenues dans la table Sales. Il s'agit aussi d'ajouter un suffixe aux clés ayant le même nom pour savoir de quelle table elle provient. Enfin, nous sauvegardons le résultat final en JSON en supprimant la clé SalesPersonID car c'est la même que EmployeeID avec un nom différent. Le résultat est un fichier JSON de 2Go importable dans MongoDB grâce à l'IDE MongoCompass. La première entrée de la nouvelle et, dorénavant unique, collection dans l'IDE MongoCompass est :

```
_id: ObjectId('63727561f06a759b9fdd5af6')
SalesID: 1
CustomerID: 10482
ProductID: 500
Quantity: 500
Name: "ML Bottom Bracket"
Price: 101.24
FirstName_customer: "Joseph"
MiddleInitial_customer: null
LastName_customer: "White"
EmployeeID: 17
FirstName_employee: "Albert"
MiddleInitial_employee: "i"
LastName_employee: "Ringer"
```

Figure 3: Affichage de la première entrée de la base de données dans Mongo Compass

2.3 Importation de la base de données en SSH

Dans un premier temps, il s'agit de créer un MS Mongos et un Serveur de configuration. Nous avons 8 serveurs à notre disposition, 2 par membre du groupe. Nous avons choisi la répartition des serveurs comme suit:

Propriétaire	Nom Serveur	Role	Shard	ReplicaSet
Roland	MESIIN592022-0022	MS Mongos et Shard	S8	RS8
Roland	MESIIN592022-0023	Shard	S3	RS3
Sébastien	MESIIN592022-0090	CS1 ReplicaSet et Shard	S7	RS7
Sébastien	MESIIN592022-0091	Shard	S4	RS4
Hugo	MESIIN592022-0068	Shard	S6	RS6
Hugo	MESIIN592022-0069	Shard	S5	RS5
Amine	MESIIN592022-0052	Shard	S1	RS1
Amine	MESIIN592022-0053	Shard	S2	RS2

Pour ce faire, nous nous connectons en ssh sur les serveurs et effectuons les commandes suivantes :

- Pour le config server: **mongod -f conf/mongo_configSvr.conf** puis **mongo --host MESIIN592022-0090 --port 27018** puis **rs.initiate()** pour initialiser le serveur de configuration.
- Pour le MS Mongos : **mongos -f conf/mongos.conf** puis **mongo --host MESIIN592022-0022 --port 30000** pour pouvoir effectuer les requêtes sur MongoDB en ayant préalablement inséré la ligne suivante dans le fichier mongos.conf : **configdb=configSvr/ MESIIN592022-0090:27018**
- Pour les shards, il s'agissait d'exécuter la commande **mongod -f conf/mongo_RSx.conf** avec x le numéro du shard puis **mongo --host XXXX** et enfin **rs.iniate()** pour initialiser le shard.

Une fois tous les serveurs démarrés, nous avons ajouté un premier shard grâce à la commande **sh.addShard("RS3/MESIIN592022-0023:27017")**, puis nous avons créé notre base de données nommée SalesDB, nous avons autorisé le sharding, nous avons créé notre collection et nous avons défini la clé de sharding sur LastName_employee car comme démontré dans le modèle de coût, cette clé de sharding est la plus optimisée parmi les deux calculées :

- **use SalesDB;**
- **sh.enableSharding("SalesDB");**
- **db.createCollection(Unique_table);**
- **sh.shardCollection('SalesDb.Unique_table', "Nom_shard": 1);**

"Nom_shard" devant être remplacé par "LastName_employee" ou "FirstName_customer", les deux configurations de sharding différentes pour cette dénormalisation.

Toutes ces manipulations étant faites, nous pouvions nous connecter en SSH sur MongoCompass et importer notre unique collection en JSON:

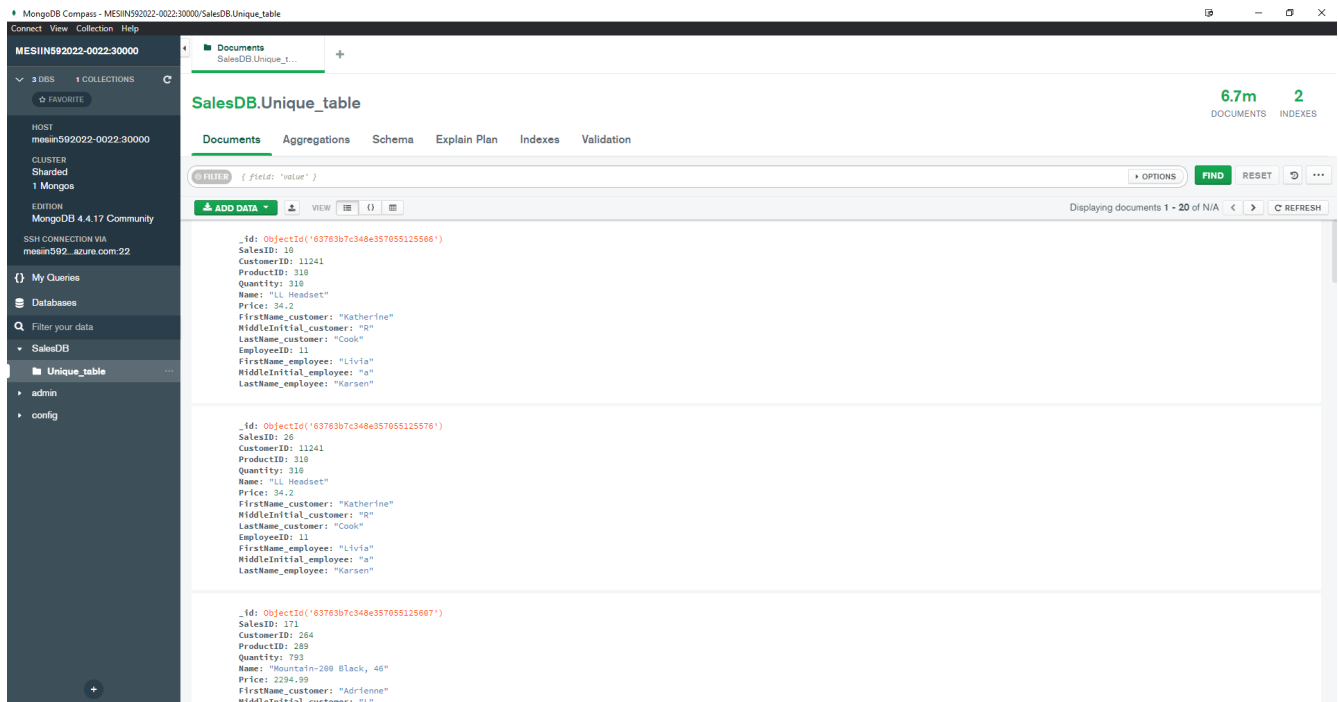


Figure 4: Page MongoDB de notre base de données

Il est de plus précisé que la collection possède 2 index ainsi que 6.7M de documents.

Par la suite, nous avons effectué nos requêtes en ajoutant un sharding à la fois avec la commande **sh.addShard("RSx/MESIIN592022-00xx:27017")**. Nous avons pu être confrontés à certains problèmes lors de l'ajout ou la suppression de shards telles que des erreurs de transfert ou des erreurs de manipulation qui nous ont amené à recommencer l'ajout d'un shard de zéro ou recommencer le processus de suppression de shard.

2.4 Statistiques en sharding

Voici un exemple de status de données shardées en console dans notre cas pour 4 shards:

```
mongos> db.printShardingStatus()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("6373ce7bf850754b74b576f8")
  }
  shards:
    { "_id" : "RS3", "host" : "RS3/MESIIN592022-0023:27017", "state" : 1 }
    { "_id" : "RS4", "host" : "RS4/MESIIN592022-0091:27017", "state" : 1 }
    { "_id" : "RS5", "host" : "RS5/MESIIN592022-0069:27017", "state" : 1 }
    { "_id" : "RS7", "host" : "RS7/MESIIN592022-0090:27017", "state" : 1 }
  active mongoses:
    "4.4.17" : 2
  autosplit:
    Currently enabled: yes
  balancer:
    Currently enabled: yes
    Currently running: no
    Failed balancer rounds in last 5 attempts: 0
    Migration Results for the last 24 hours:
      3955 : Success
      24 : Failed with error 'aborted', from RS7 to RS3
      1 : Failed with error 'aborted', from RS3 to RS4
      1 : Failed with error 'aborted', from RS3 to RS5
      1 : Failed with error 'aborted', from RS3 to RS7
  databases:
    { "_id" : "SalesDB", "primary" : "RS3", "partitioned" : true, "version" : { "uuid" : UUID("50128f44-7d11-4a08-a667-4be5f44e043b"), "lastMod" : 1 } }
      SalesDB.Unique_table
        shard key: { "LastName_employee" : 1 }
        unique: false
        balancing: true
        chunks:
          RS3      6
          RS4      5
          RS5      6
          RS7      6
          too many chunks to print, use verbose if you want to force print
    { "_id" : "config", "primary" : "config", "partitioned" : true }
      config.system.sessions
        shard key: { "_id" : 1 }
        unique: false
        balancing: true
        chunks:
          RS3      266
          RS4      225
          RS5      266
          RS7      267
          too many chunks to print, use verbose if you want to force print
    { "_id" : "test", "primary" : "RS3", "partitioned" : false, "version" : { "uuid" : UUID("b1c04b36-d8c2-45b2-bba3-fd6390f64343"), "lastMod" : 1 } }
```

Figure 5: Exemple de résultat en console de la commande db.printShardingStatus()

```
mongos> db.Unique_table.getShardDistribution()

Shard RS5 at RS5/MESIIN592022-0069:27017
data : 632.9MiB docs : 2071655 chunks : 6
estimated data per chunk : 105.48MiB
estimated docs per chunk : 345275

Shard RS4 at RS4/MESIIN592022-0091:27017
data : 354MiB docs : 1160156 chunks : 5
estimated data per chunk : 70.8MiB
estimated docs per chunk : 232031

Shard RS3 at RS3/MESIIN592022-0023:27017
data : 801.5MiB docs : 2631855 chunks : 6
estimated data per chunk : 133.58MiB
estimated docs per chunk : 438642

Shard RS7 at RS7/MESIIN592022-0090:27017
data : 615.68MiB docs : 2011711 chunks : 6
estimated data per chunk : 102.61MiB
estimated docs per chunk : 335285

Totals
data : 2.34GiB docs : 7875377 chunks : 23
Shard RS5 contains 26.32% data, 26.3% docs in cluster, avg obj size on shard : 3208
Shard RS4 contains 14.72% data, 14.73% docs in cluster, avg obj size on shard : 3198
Shard RS3 contains 33.33% data, 33.41% docs in cluster, avg obj size on shard : 3198
Shard RS7 contains 25.6% data, 25.54% docs in cluster, avg obj size on shard : 3208
```

Figure 6: Exemple de détail de la répartition du sharding

Informations shard par shard:

- 1 Shard:
 - RS3:
 - * Nombre de chunks: 23
 - * Nombre de documents: 6,715,221
- 2 Shards:
 - RS3:
 - * Nombre de chunks: 12
 - * Nombre de documents: 3,720,919
 - RS7:
 - * Nombre de chunks: 11
 - * Nombre de documents: 3,141,212
 - Total:
 - * Nombre de chunks: 23
 - * Nombre de documents: 6,862,131
- 3 Shards:
 - RS3:
 - * Nombre de chunks: 8
 - * Nombre de documents: 2,631,855
 - RS5:
 - * Nombre de chunks: 8
 - * Nombre de documents: 2,071,655
 - RS7:
 - * Nombre de chunks: 7
 - * Nombre de documents: 2,011,711
 - Total:
 - * Nombre de chunks: 23
 - * Nombre de documents: 6,862,131
- 4 Shards:
 - RS3:
 - * Nombre de chunks: 6
 - * Nombre de documents: 2,631,855
 - RS4:
 - * Nombre de chunks: 5
 - * Nombre de documents: 1,160,156
 - RS5:
 - * Nombre de chunks: 6
 - * Nombre de documents: 2,071,655
 - RS7:
 - * Nombre de chunks: 6
 - * Nombre de documents: 2,011,711
 - Total:
 - * Nombre de chunks: 23

- * Nombre de documents: 7,875,377
- 5 Shards:
 - RS3:
 - * Nombre de chunks: 5
 - * Nombre de documents: 2,086,215
 - RS4:
 - * Nombre de chunks: 5
 - * Nombre de documents: 1,160,156
 - RS5:
 - * Nombre de chunks: 5
 - * Nombre de documents: 1,759,608
 - RS6:
 - * Nombre de chunks: 3
 - * Nombre de documents: 854,657
 - RS7:
 - * Nombre de chunks: 5
 - * Nombre de documents: 1,709,242
 - Total:
 - * Nombre de chunks: 23
 - * Nombre de documents: 7,569,878
- 6 Shards:
 - RS1:
 - * Nombre de chunks: 3
 - * Nombre de documents: 879,441
 - RS3:
 - * Nombre de chunks: 4
 - * Nombre de documents: 1,784,004
 - RS4:
 - * Nombre de chunks: 4
 - * Nombre de documents: 1,160,156
 - RS5:
 - * Nombre de chunks: 4
 - * Nombre de documents: 1,482,475
 - RS6:
 - * Nombre de chunks: 4
 - * Nombre de documents: 854,657
 - RS7:
 - * Nombre de chunks: 4
 - * Nombre de documents: 1,434,329
 - Total:
 - * Nombre de chunks: 23
 - * Nombre de documents: 7,594,662

- 7 Shards:
 - RS1:
 - * Nombre de chunks: 3
 - * Nombre de documents: 879,441
 - RS2:
 - * Nombre de chunks: 3
 - * Nombre de documents: 755,162
 - RS3:
 - * Nombre de chunks: 3
 - * Nombre de documents: 1,491,028
 - RS4:
 - * Nombre de chunks: 4
 - * Nombre de documents: 1,160,156
 - RS5:
 - * Nombre de chunks: 3
 - * Nombre de documents: 1,187,881
 - RS6:
 - * Nombre de chunks: 4
 - * Nombre de documents: 854,657
 - RS7:
 - * Nombre de chunks: 3
 - * Nombre de documents: 1,142,058
 - Total:
 - * Nombre de chunks: 23
 - * Nombre de documents: 7,470,383

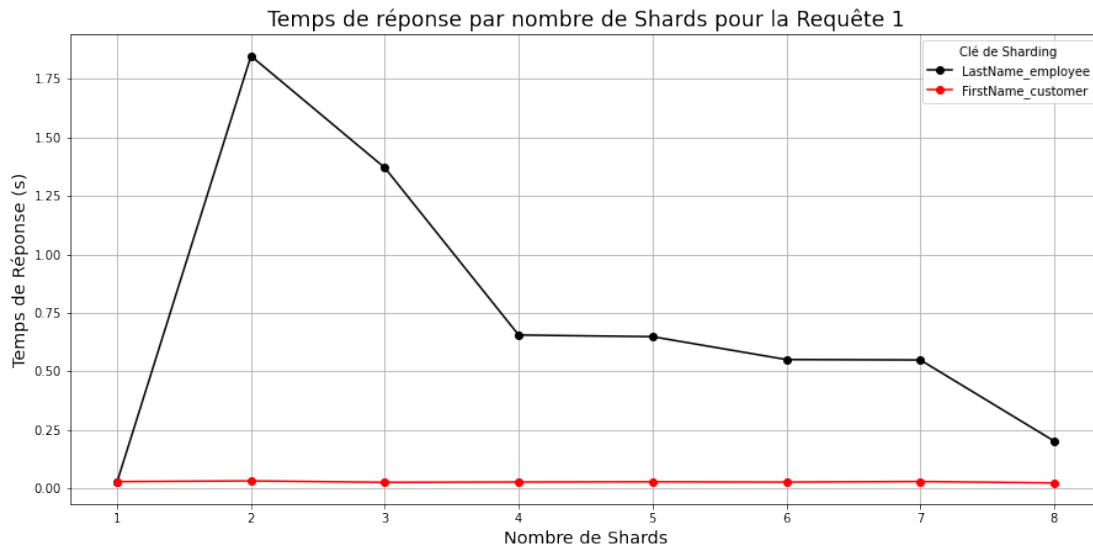
- 8 Shards:
 - RS1:
 - * Nombre de chunks: 3
 - * Nombre de documents: 879,441
 - RS2:
 - * Nombre de chunks: 3
 - * Nombre de documents: 755,162
 - RS3:
 - * Nombre de chunks: 3
 - * Nombre de documents: 1,281,614
 - RS4:
 - * Nombre de chunks: 3
 - * Nombre de documents: 848,109
 - RS5:
 - * Nombre de chunks: 3
 - * Nombre de documents: 900,065
 - RS6:
 - * Nombre de chunks: 3
 - * Nombre de documents: 854,657

- RS7:
 - * Nombre de chunks: 3
 - * Nombre de documents: 884,126
- RS8:
 - * Nombre de chunks: 2
 - * Nombre de documents: 312,047
- Total:
 - * Nombre de chunks: 23
 - * Nombre de documents: 6,715,221

3 Mesure de performances

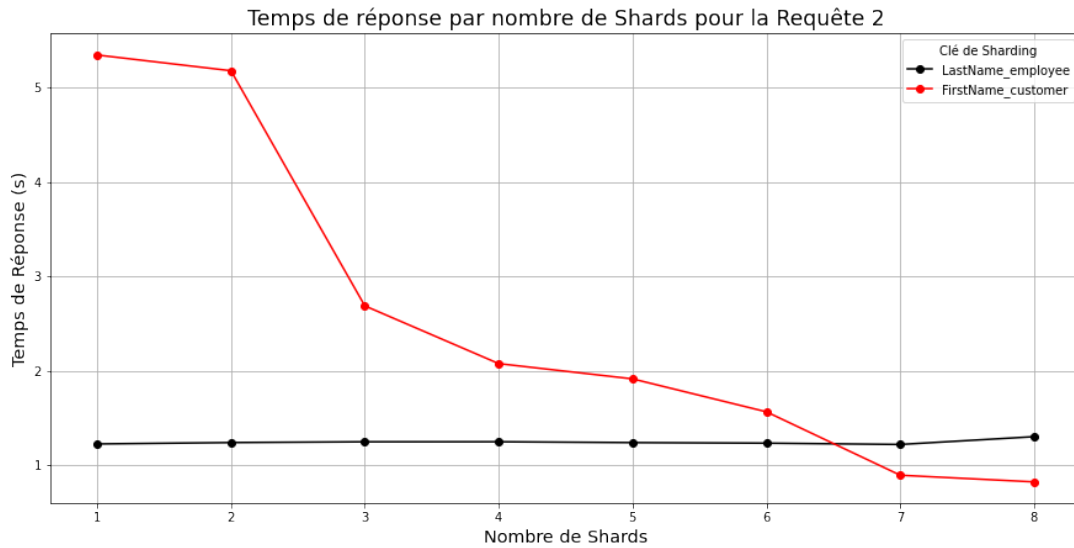
Afin de pouvoir mesurer les performances shard par shard, un programme python a été écrit (disponible sur GitHub en suivant [ce lien](#)), se connectant au MS Mongos en SSH et mesurant le temps d'exécution de chaque requête pour une configuration de sharding en particulier. L'opération a été répétée 8 fois pour pouvoir avoir les mesures sur 1 à 8 shards.

3.1 Requête 1



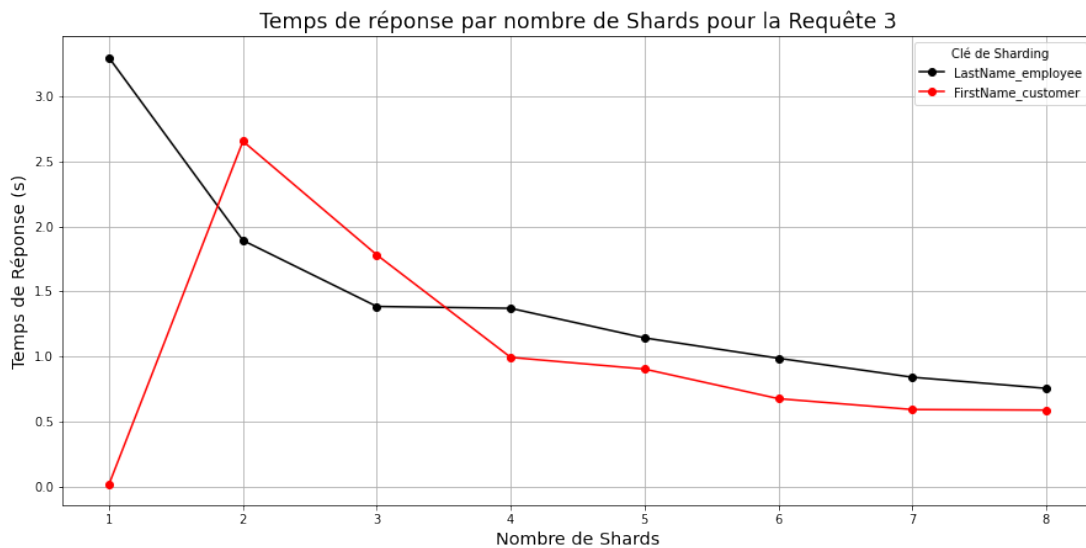
Pour le clé de sharding "LastName_employee", nous constatons qu'entre 1 et 2 shards, le temps de réponse augmente drastiquement. Cela est probablement dû au fait que la requête étant très simple, les communications ont une grande influence sur le temps de réponse. Il est donc plus optimisé de ne conserver qu'un seul shard pour cette requête, ou alors grandement augmenter le nombre de shards car la courbe semble tendre vers le temps de réponse pour un shard quand le nombre de shards augmente. Néanmoins, dans l'objectif d'avoir une base de données optimisée, il faut considérer le temps de réponse pour toutes les requêtes avant de choisir une implémentation de sharding finale. En revanche, pour la clé de sharding "FirstName_customer", le temps de réponse est nettement plus faible et relativement stable selon le nombre de shards.

3.2 Requête 2



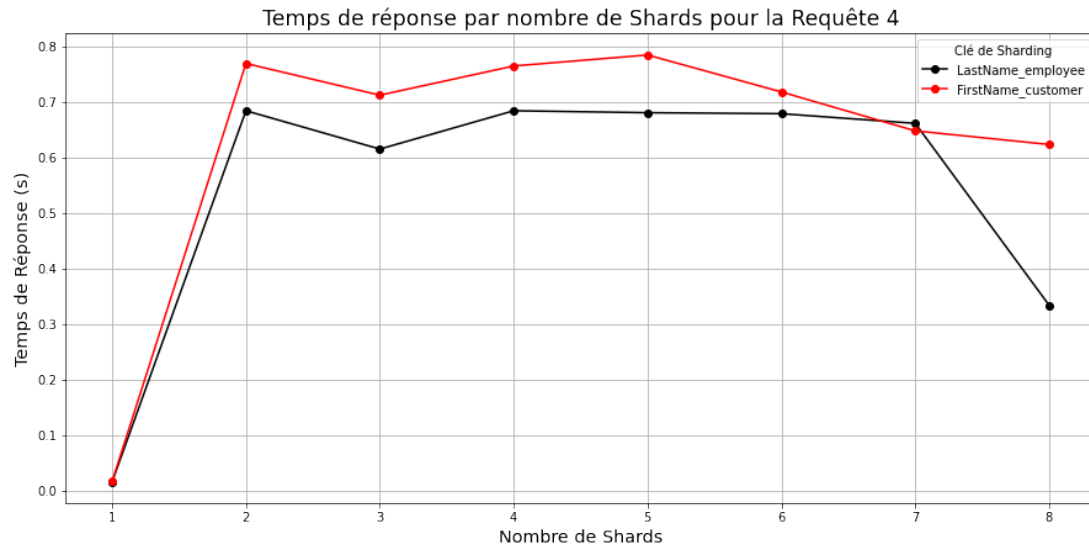
Nous pouvons constater une nette différence entre les deux configurations de sharding: en effet, l'une semble nettement impactée par le nombre de shards avec une baisse nette du temps de réponse avec l'augmentation du nombre de shards alors que pour l'autre configuration, le temps de réponse semble relativement stable et satisfaisant de 1 à 7 shards. Nous constatons une augmentation du temps de réponse de la requête pour 8 shards. Sachant que le 8ème shard testé est situé sur la même machine virtuelle que le MS Mongos, nous pouvons en conclure que sharder sur le MS Mongos n'est pas profitable, car il semble prioriser les recherches sur son propre serveur avant de regarder les autres shards. Dans cette situation, la configuration de sharding a une forte influence sur le temps de réponse.

3.3 Requête 3



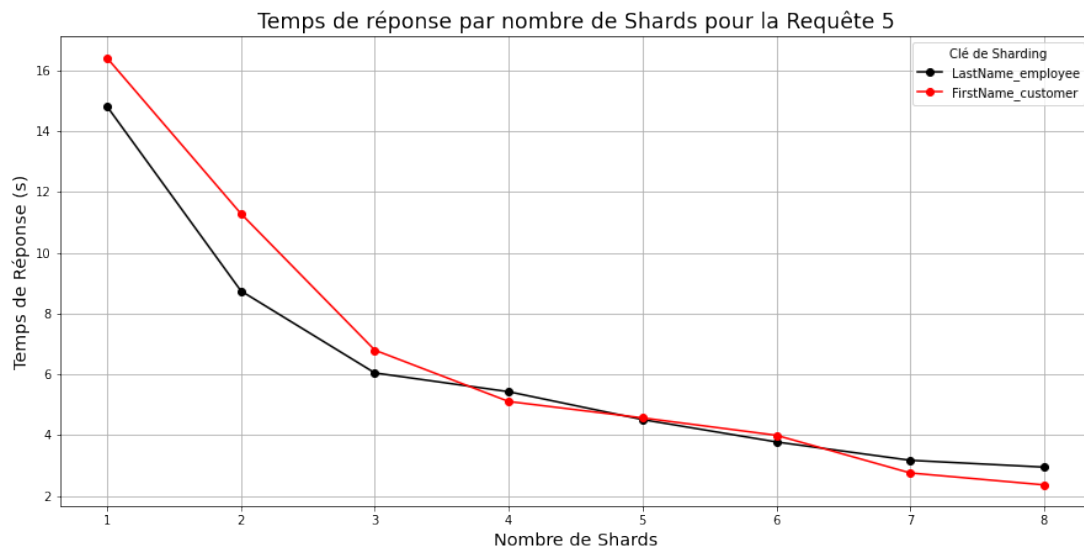
Le temps de réponse de la requête décroît en fonction du nombre de shards, avec un petit sursaut au 8ème shard pour la même raison que la requête précédente. Les choix de sharding semblent donc optimisés pour cette requête, ou du moins de bonne qualité. Comme pour la requête 1 mais sur l'autre configuration de sharding, le temps de réponse entre 1 et 2 shards augmente en raison de la simplicité de la requête qui fait qu'inclure des communications fait augmenter le temps de réponse.

3.4 Requête 4

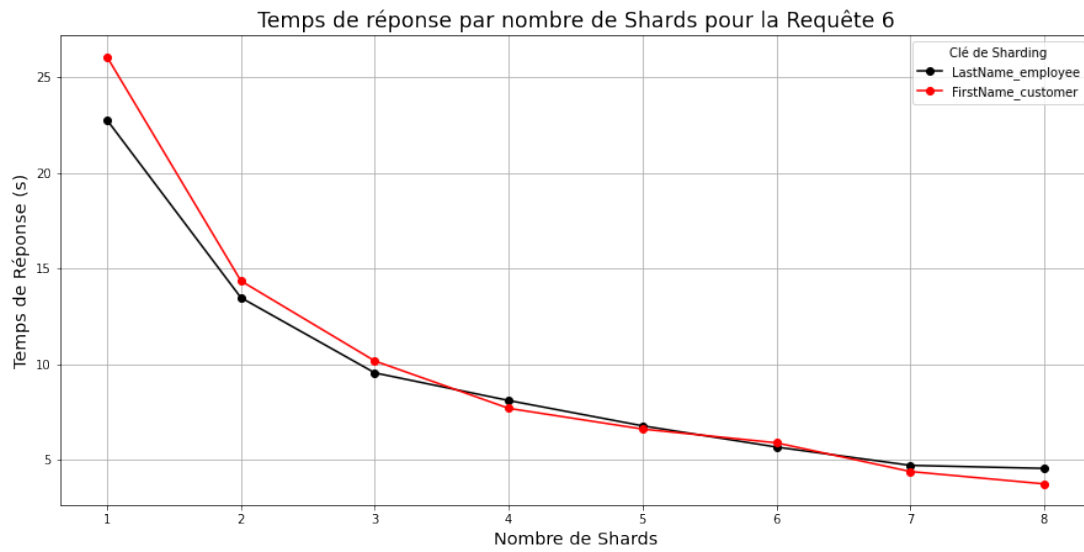


Comme pour les requête 1 et 3, le temps augmente entre 1 sharding et 2 shardings à cause des communications qui allongent le temps de réponse comparé à la complexité de la requête. La chute entre 7 et 8 shards est probablement due au fait que les données concernées par la requête sont situées sur le shard S8.

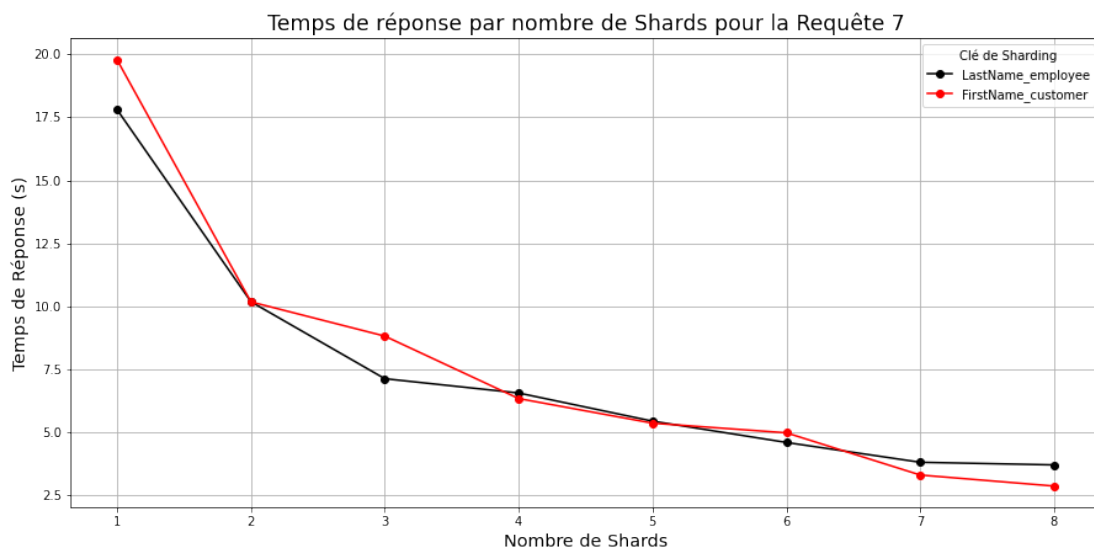
3.5 Requête 5



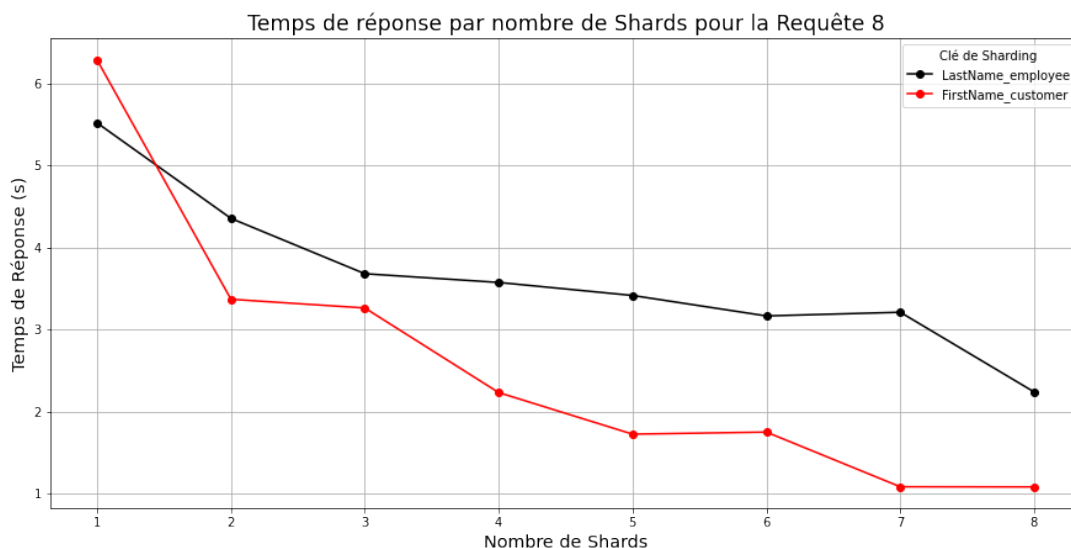
3.6 Requête 6



3.7 Requête 7



3.8 Requête 8



Pour les 4 requêtes complexes ci-dessus. La courbe est parfaitement comme on pouvait s'y attendre, c'est à dire convexe décroissante. Cela montre que la configuration de sharding est optimisée pour ces requêtes. Nous constatons néanmoins que la configuration de sharding peut avoir un impact positif, comme le montrent les courbes de la requête 8.

3.9 Conclusion sur les performances

La majorité des courbes de temps de réponse semblent montrer que la configuration de sharding est bénéfique. Nous pouvons donc la conserver. De plus, dans un cas concret, nous augmenterions le nombre de RéplikatSet ConfigServ et de MS Mongos afin de s'assurer que la base de données soit tolérante aux pannes.

4 Vues

Afin de créer les vues demandées, nous avons utilisé la librairie Flask sur Python afin de créer une application interactive sur un serveur local, le code est disponible sur [GitHub](#) en suivant [ce lien](#). Il est important de savoir que les serveurs doivent être lancés et opérationnels en accord avec la [tableau de description de la base de données](#) présent précédemment dans ce rapport (2.3). Cette application offre un menu principal permettant de choisir la vue que nous souhaitons lancer:

App Cloud - Structures de données pour le cloud - Groupe 1

Menu Principal

Vues Utilisateur Standard

Vues Analyste/Décisionnaire

Vue Administrateur

Figure 7: Menu principal de l'application cloud

La première case interactive sert à customiser le nombre de lignes de résultat à afficher car il a été constaté que lorsque de très nombreux résultats sont à afficher, cela peut mettre beaucoup de temps. Dans l'objectif d'une démonstration, le nombre de résultats sera par défaut limité à 5. De plus, lorsque chaque bouton est cliqué, il est possible de revenir au menu principal via un bouton en haut de la page.

4.1 Vues utilisateur standard

Les 4 premiers boutons cliquables concernent les vues d'un utilisateur standard, pour les quatre requêtes simples. Il n'y a pas d'affichage de la requête mais seulement de son résultat.

App Cloud - Structures de données pour le Cloud - Groupe 1

Résultats de la Requete 2

	_id	Sum of quantity
Ringer	293741490	

Figure 8: Exemple de résultat pour la requête 2

App Cloud - Structures de données pour le Cloud - Groupe 1

Résultats de la Requete 3

Menu Principal

	SalesID	Price
2344616	130.4	
2373455	130.4	
2385516	130.4	
2435609	130.4	
2505626	130.4	

Figure 9: Exemple de résultat pour la requête 3

4.2 Vues Analyste/Décisionnaire

Les 4 boutons cliquables suivants concernent les vues d’un analyste/décisionnaire, pour les quatre requêtes lourdes. La requête est affichée ainsi que son résultat et certains paramètres sont modifiables via un formulaire HTML.

App Cloud - Structures de données pour le Cloud - Groupe 1

Résultats de la Requete 5

Menu Principal

Intitulé de la requête:

[[{"\$group": [{"_id": "\$CustomerID", "FirstName_customer": {"\$first": "\$FirstName_customer"}, "LastName_customer": {"\$first": "\$LastName_customer"}, "MiddleInitial_employee": {"\$first": "\$MiddleInitial_employee"}, "Somme du nombre d'achats": {"\$sum": "\$Quantity"}}], [{"\$match": {"MiddleInitial_employee": "e", "Somme du nombre d'achats": {"\$gte": 1000}}}], [{"project": {"FirstName_customer": 1, "LastName_customer": 1, "_id": 0}], {"\$sort": {"LastName_customer": 1}}]]

Customisation de MiddleName_employee

Customisation de Somme du nombre d'achats

Confirmer

FirstName_customer	LastName_customer
Mackenzie	Adams
Aaron	Alexander
Bruce	Alonso
Shaun	Andersen
Kate	Becker

Figure 10: Exemple de résultat pour la requête 5

App Cloud - Structures de données pour le Cloud - Groupe 1

Résultats de la Requete 8

[Menu Principal](#)

Intitulé de la requête:
[{"\$match": {"MiddleInitial_Employee": "e"}}, {"\$group": {"_id": {"EmployeeID": "\$EmployeeID", "ProductID": "\$ProductID", "Name": {"\$first": "\$Name", "Nb": {"\$sum": "\$Quantity"}}}, {"\$sort": {"Nb": -1}}, {"\$limit": 1}, {"\$project": {"Name": 1, "Nb": 1, "_id": 0}}]

Customisation de MiddleName_employee

Customisation de la limite

Confirmer

	Name	Nb
HL Mountain Tire		23740666

Figure 11: Exemple de résultat pour la requête 8

4.3 Vue Administrateur

La vue administrateur, accessible via le dernier bouton cliquable, propose des statistiques et informations à propos de la base de données. En autres, sont disponibles informations de sharding, d'indexation et de distribution des données.

App Cloud - Structures de données pour le Cloud - Groupe 1

Statistiques Administrateur

[Menu Principal](#)

Indexes existants: ['LastName_employee_1', '_id_']
Nombre de shards: 8
Nombre de chunks: 23

Statistiques des shards:

Nom du Shard	Nombre de documents	Taille des données stockées	Pourcentage des données stockées
RS1	879441	282.53 Mo	13.14%
RS2	755162	241.1 Mo	11.22%
RS3	1281614	410.55 Mo	19.1%
RS4	848109	269.25 Mo	12.53%
RS5	900065	288.51 Mo	13.42%
RS6	854657	271.14 Mo	12.61%
RS7	884126	284.65 Mo	13.24%
RS8	312047	101.95 Mo	4.74%

Figure 12: Exemple de résultat pour la vue administrateur