Projet DB\_165



**Rapport de Projet**

Zeqiri Amir – CID2A

ETML, Vennes – A01

32 Périodes

Maître : M. Mveng

Table des matières

[1 Description du projet dans son ensemble 3](#_Toc161385757)

[1.1 Sujet 3](#_Toc161385758)

[1.2 Spécificités 3](#_Toc161385759)

[1.2.1 Chapitre A – Importer les données et le schéma de la base de données 3](#_Toc161385760)

[1.2.2 Chapitre F – Création des index 3](#_Toc161385761)

[1.2.3 Chapitre G – Backup/Restore 3](#_Toc161385762)

[2 Pratique 4](#_Toc161385763)

[2.1 A – Importer les données et le schéma de la base de données 4](#_Toc161385764)

[2.2 F – Création des index 4](#_Toc161385765)

[2.3 G – Backup/Restore 4](#_Toc161385766)

# Description du projet dans son ensemble

## Sujet

Implémenter et utiliser une base de données MongoDB.

## Spécificités

### Chapitre A – Importer les données et le schéma de la base de données

* Restaurer les collections dans une base de données nommé « db\_mflix » dans une instance de MongoDB.
* Expliquer en détail la commande utilisée.

### Chapitre F – Création des index

* Dans votre base de données « db\_mflix », identifiez les champs qui bénéficieraient le plus de la création d'index.

### Chapitre G – Backup/Restore

* Expliquez chaque étape de ces processus et les raisons pour lesquelles ces étapes sont importantes.

# Pratique

## A – Importer les données et le schéma de la base de données

docker exec -i mongo mongorestore --uri=mongodb://root:admin@localhost:27017 --gzip --archive=backupdb/db\_mflix.gz

Explications de la commande utilisée :

## docker exec: Cela signifie que nous utilisons Docker pour exécuter une commande à l'intérieur d'un conteneur Docker.

## -i: Cela dit à Docker d'exécuter la commande de manière interactive, en prenant l'entrée standard de l'utilisateur pour cette commande.

## mongo: C'est le nom du conteneur Docker où se trouve MongoDB, le système de gestion de base de données.

## mongorestore: C'est la commande spécifique à MongoDB que nous utilisons pour restaurer une base de données à partir d'une sauvegarde.

## --uri=mongodb://root:admin@localhost:27017: C'est l'adresse pour se connecter à MongoDB. "root" est le nom d'utilisateur et "admin" est le mot de passe pour se connecter à MongoDB. "localhost:27017" spécifie l'emplacement et le port de MongoDB.

## --gzip: Cela signifie que l'archive de sauvegarde est compressée avec Gzip. Cette option indique à la commande qu'elle doit décompresser l'archive avant de restaurer la base de données.

## --archive=backupdb/db\_mflix.gz: C'est l'emplacement de l'archive de sauvegarde que nous voulons restaurer. Elle est située dans le répertoire backupdb à l'intérieur du conteneur Docker.

## F – Création des index

## G – Backup/Restore

### Backup (sauvegarde)

Pour sauvegarder la base de données, j’ai utilisé la commande mondodump. Voici la commande :

docker exec -i mongo mongodump --uri=mongodb://root:admin@localhost:27017 --authenticationDatabase=admin --db="db\_mflix" --gzip --archive=./backupdb/mongodump-db\_mflix.gz

Explications de la commande utilisée :

docker exec: Cette commande est utilisée pour exécuter une commande à l'intérieur d'un conteneur Docker.

-i: Cette option signifie que nous voulons exécuter la commande de manière interactive, en utilisant l'entrée standard.

mongo: C'est le nom du conteneur Docker dans lequel nous voulons exécuter la commande. Ce conteneur doit héberger une instance de MongoDB.

mongodump: C'est la commande spécifique à MongoDB pour sauvegarder une base de données.

--uri=mongodb://root:admin@localhost:27017: C'est l'adresse pour se connecter à MongoDB. "root" est le nom d'utilisateur et "admin" est le mot de passe pour se connecter à MongoDB. "localhost:27017" spécifie l'emplacement et le port de MongoDB.

--authenticationDatabase=admin: Cette option spécifie la base de données utilisée pour l'authentification, qui est dans ce cas admin.

--db="db\_mflix": Cela spécifie la base de données à sauvegarder, qui est db\_mflix.

--gzip: Cette option indique que nous voulons compresser la sauvegarde.

--archive=./backupdb/mongodump-db\_mflix.gz: C'est l'emplacement et le nom du fichier dans lequel nous voulons stocker la sauvegarde. L'option --archive spécifie le chemin de l'archive de sauvegarde, ici ./backupdb/mongodump-db\_mflix.gz

Voici le fichier « gzipé » qui prend peu de place :



Maintenant, voici sans le –gzip :



L’option --gzip permet de de compresser le fichier et donc de prendre moi moins de place.

### Restore (restauration)

Pour restaurer la base de données, j’ai utilisé la commande mongorestore. Voici la commande :

docker exec -i mongo mongorestore --uri=mongodb://root:admin@localhost:27017 --gzip --authenticationDatabase=admin --drop --archive=./backupdb/mongodump-db\_mflix.gz

docker exec: Cette commande est utilisée pour exécuter une commande à l'intérieur d'un conteneur Docker.

-i: Cette option indique que la commande sera exécutée de manière interactive, utilisant l'entrée standard.

mongo: C'est le nom du conteneur Docker dans lequel nous voulons exécuter la commande. Ce conteneur doit héberger une instance de MongoDB.

mongorestore: C'est la commande spécifique à MongoDB pour restaurer une base de données à partir d'une sauvegarde.

--uri=mongodb://root:admin@localhost:27017: C'est l'URI de connexion à MongoDB. Il spécifie l'hôte (localhost) et le port (27017) ainsi que le nom d'utilisateur (root) et le mot de passe (admin) pour se connecter à la base de données.

--authenticationDatabase=admin: Cette option spécifie la base de données utilisée pour l'authentification.

--drop: Cette option indique à MongoDB de supprimer toutes les collections existantes dans la base de données cible avant de restaurer à partir de la sauvegarde.

--gzip: Cette option indique que l'archive de sauvegarde est compressée avec Gzip.

--archive=./backupdb/mongodump-db\_mflix.gz: C'est l'emplacement et le nom du fichier dans lequel nous voulons stocker la sauvegarde. L'option --archive spécifie le chemin de l'archive de sauvegarde, qui est ./backupdb/mongodump-db\_mflix.gz.