



# SAE PYTHON ET MONGODB

Migration de données vers un environnement NoSQL

**Victor Chassé  
Axel Dagnaud  
Noah de Gunst**

**Enseignant :  
François Garnier**



# I. CHOIX DES COLLECTIONS

La base de données orientée documents, conçue via MongoDB, est structurée autour de trois collections principales :

- **Info\_pays**

Cette collection contient des informations générales et statiques sur chaque pays, telles que le nom, le code ISO, le continent, etc. Ces données sont figées dans le temps et ne sont pas destinées à être modifiées fréquemment.

- **Stats\_covid**

Cette collection regroupe des données temporelles liées à la pandémie de COVID-19, telles que les statistiques sur les cas, les vaccinations, les hospitalisations, etc. Elle peut être mise à jour quotidiennement selon les besoins.

- **Stats\_demographie**

Cette collection contient des données démographiques pour chaque pays, comme la population, l'âge médian, et d'autres indicateurs similaires. Ces données peuvent être actualisées périodiquement.

Les collections incluent un certain degré de redondance, notamment pour des champs comme le nom du pays et la population. Cette approche vise à minimiser les jointures entre collections, qui peuvent être coûteuses en termes de ressources et de temps, particulièrement dans un contexte de bases de données hébergées sur le cloud où la facturation dépend de l'utilisation des ressources.

En outre, cette redondance facilite l'historisation des données. Par exemple, si la population d'un pays est mise à jour dans la collection Info\_pays, les enregistrements existants dans Stats\_covid conserveront les valeurs de population historiques, maintenant ainsi une chronologie précise des données.



## II. STRUCTURE DU PROJET

À la racine du projet, vous trouverez le fichier **dataprep\_et\_insertion\_mongodb.py**, qu'il convient d'exécuter à l'aide d'un interpréteur Python pour créer la base de données et les collections nécessaires. De plus, dans le même répertoire, vous pourrez accéder aux 12 requêtes MongoDB et SQLite.

Pour visualiser le tableau de bord, rendez-vous dans le dossier Tableau de Bord, où se trouve le fichier Python '**Tableau de Bord.py**', développé avec Dash. Ce script permet d'afficher et d'interagir avec le tableau de bord.