



SAE4-01 : Développement avec une base de données et visualisation

Rapport sur la réalisation du Projet



Professeur : SANDRON Lydie / BLANCHARD Frédéric
Mail : lydie.sandron@univ-reims.fr / frederic.blanchard@univ-reims.fr
Année d'étude : 2022-2023

DANNEAUX Lucas TP1-B

Choix des Données utilisés

Pour la réalisation du projet, j'ai choisi d'utiliser 2 sources de données :

Liste des personnes décédés en France depuis 1970 : Données issues de data.gouv.fr (INSEE) et qui est transformé en API par le site public.opendatasoft.com qui recense les personnes décédés en France avec notamment la ville de leurs décès et leur âge qui sera utiliser dans l'application, l'ajout de nouvelles entrée n'est pas régulière (environ toutes les 2 semaines)

Observations météorologiques historiques France (SYNOP) : Données issues de donneespubliques.meteofrance.fr et qui est transformé en API par le site public.opendatasoft.com qui recense depuis 2010 les données météorologiques collectés toutes les 3h d'une même journée pour une cinquantaines d'EPCI (communautés de communes) en France Métropolitaine et dans certains DROM Français (Mayotte, Guadeloupe, ...) notamment la température moyenne sur 3h ou encore l'humidité de ce secteur sur 3h, l'ajout de nouvelles entrées est quotidienne (entre 21h et 23h)

J'ai choisi ces deux sources de données car je trouvais l'analyse du nombre de décès en fonction de condition météorologique vraiment pertinent en plus de m'offrir de nombreuses données à traiter

Architecture du Projet

Le projet a été réalisé en Python avec l'architecture suivante :

App : Contient les classes qui s'occupe pour chacune d'un concept de l'application

City.py : Classe s'occupant d'ajouter les villes et EPCI dans la base de données en début d'exécution de l'application ainsi qu'insérer les nouvelles entrées et gérer les villes dans lequel insérer les données en fonction des données météorologiques reçu

Connection.py : Classe mettant en place la connexion avec la base de données de l'application à l'aide du module PyMySQL

Database.py : Classe gérant la suppression des tables si ces dernières existent dans la base de données de l'application et la création des tables dans la base de données

People.py : Classe s'occupant de l'insertion des personnes décédés pour une date donnée dans la base de données, seulement si ces dernières appartiennent à une ville présente dans les données météorologiques de ce jour

Weather.py : Classe s'occupant de l'insertion des condition météorologiques de chaque EPCI pour une date donnée dans la base de données, un calcul de la moyenne de chacun de ces éléments sur la journée est au préalable réalisé

Data : Contient les 2 tableurs qui serviront à certaines classes du répertoire App et qui ont été récupérés sur les sites du gouvernement et contenant respectivement la liste des communautés de communes avec les villes les constituant et la liste des départements français

Application.py : Contient la boucle d'exécution de l'application qui fait appel aux différentes classes du répertoire App

Requirements.txt : Contient tous les modules Python qui sont à installer dans l'environnement de développement pour permettre son fonctionnement

Les autres fichiers présents dans le dépôt ne sont pas utiles aux fonctionnements de l'application et sont juste des fichiers de rendu d'autre support comme le MCD, les commandes de la machine virtuel ainsi que la Visualisation et un dossier en faisant l'explication

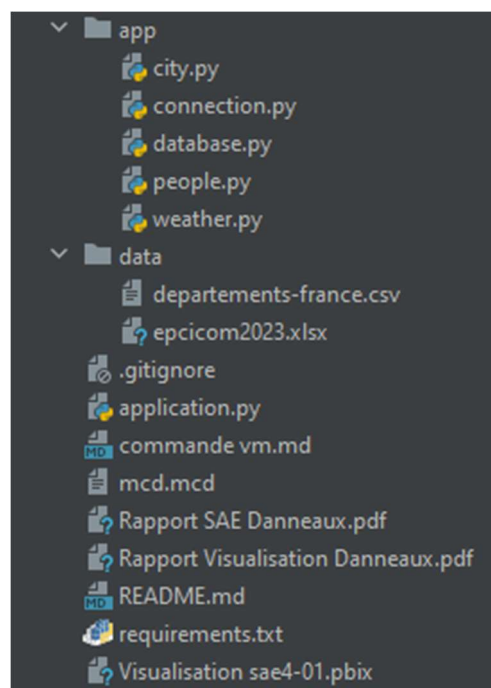


Figure 1 Composition du répertoire du projet

Modèle de la base de données

Pour la base de données de l'application, j'ai choisi de mettre en place une base de données relationnelle car il s'agissait selon moi du modèle le plus économique en ressource pour le stockage des données, de plus les relations entre certaines entités étant au cœur de l'application (comme conditionMétéo-EPCI). Enfin choisir un modèle relationnel était sécurisant étant le modèle que nous avons le plus appris à concevoir et à mettre en place pour le moment.

Le modèle de la base de données de l'application est basé sur le MCD suivant

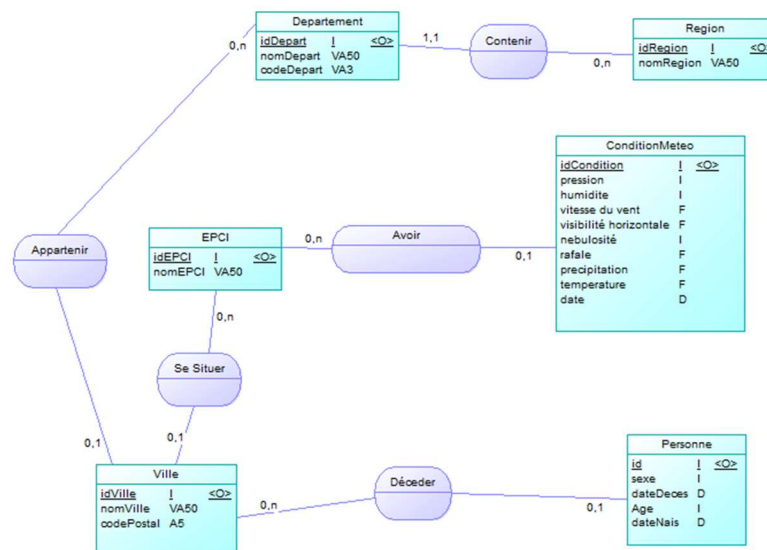


Figure 2 MCD de l'application

Outils de Gestion de la base de données

Gestion et la visualisation des données MySQL de la base de données à l'aide de l'application phpMyAdmin qui est un outil simple d'utilisation et d'installation sur la machine virtuelle

Table	Action	Lignes	Type	Interclassement	Taille	Perte
conditionmeteo	★ Parcourir Structure Rechercher Insérer Vider Supprimer	~239 034	InnoDB	utf8mb4_general_ci	19,0 Mio	-
departement	★ Parcourir Structure Rechercher Insérer Vider Supprimer	57	InnoDB	utf8mb4_general_ci	32,0 kio	-
epci	★ Parcourir Structure Rechercher Insérer Vider Supprimer	51	InnoDB	utf8mb4_general_ci	16,0 kio	-
personne	★ Parcourir Structure Rechercher Insérer Vider Supprimer	~2 203 548	InnoDB	utf8mb4_general_ci	115,2 Mio	-
region	★ Parcourir Structure Rechercher Insérer Vider Supprimer	18	InnoDB	utf8mb4_general_ci	16,0 kio	-
ville	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1 863	InnoDB	utf8mb4_general_ci	208,0 kio	-
6 tables	Somme	~2 444 571	InnoDB	utf8mb4_general_ci	134,5 Mio	0 o

Figure 3 Vue des tables sur phpMyAdmin

Outils de Visualisation

Concernant le choix du logiciel pour la réalisation des visualisations, j'ai choisi d'utiliser le logiciel Power BI afin de réaliser une présentation des données collectés sur les personnes décédés et les données météorologiques (voir détails de la visualisation dans le document Visualisation.pdf) car c'est un logiciel que j'ai déjà utiliser pour la réalisation d'autre projet en plus de l'avoir manier à de nombreuses reprises durant ce semestre, et qui m'a permis de mettre en place des graphiques bien légendé là ou des outils python aurait demandé beaucoup plus de temps pour être mis en place.

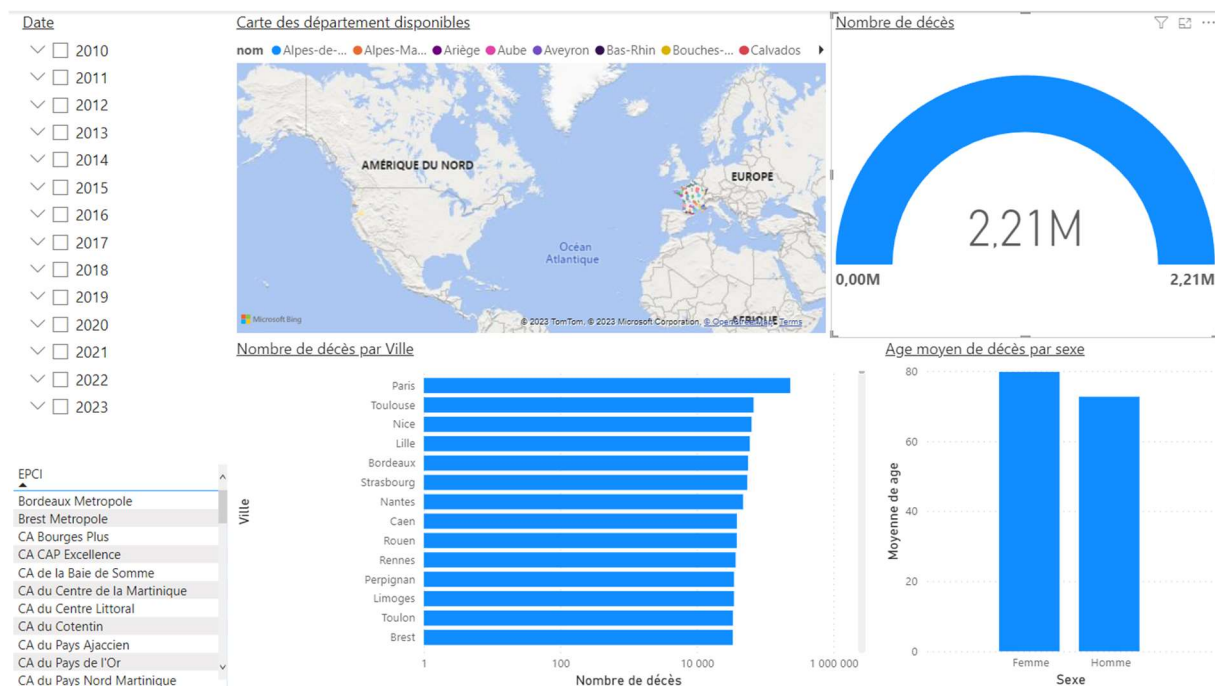


Figure 4 Première Page de la Visualisation