Université Cheikh Anta Diop

Faculté Des Sciences Et Techniques
Département De Mathématiques-Informatique

Mémoire présenté pour l'obtention du diplôme de

Master en Informatique
Spécialité : **Business Intelligence**

Par

BOUMI CHIMI Franck Yannick

Sur le sujet

Exploitation et valorisation des données du projet ALSO-COVID-19

Soutenu le 5 juin 2021 devant le jury composé comme suit :

Président :	Pr Mbaye Sene, Professeur Titulaire, UCAD
Examinateurs	Dr Djamal A. N. Seck, Maître-Assistant, UCAD
	Dr Mamadou Thiongane, Maître-Assistant, UCAD
Directeur de mémoire :	Pr Idrissa Sarr, Professeur Titulaire, UCAD
Encadrants :	Pr Aliou Boly, Maître de Conférences, UCAD
	Dr Modou Gueye, Maître-Assistant, UCAD
	Dr Ndiouma Bame, Maître-Assistant, UCAD

Année universitaire 2019-2020

Ce mémoire est financé par le projet **ALSO-COVID-19**, lauréat du *Fonds Macky Sall pour la Recherche du CAMES*, première édition. Par conséquent, je voudrais exprimer toute ma gratitude à l'équipe de pilotage dudit projet et aux membres du WP5 sans lesquels ce mémoire n'aurait pas abouti dans les délais et avec la qualité y afférente.

Dédicaces

Je dédie ce mémoire à mes chers et tendres parents **BOUMI Nestor** et **BEGO Julienne** d'abord pour l'éducation qu'ils m'ont donnée, aussi pour leurs efforts et leur soutien incontournable.

Je dédie également ce mémoire à mes frères **Berthold BOUMI**, **Raoul BOUMI**, à mes soeurs **Sylvianne MALA**, **Guillainne BOUMI** pour leur amour qu'ils ont toujours porté à mon égard et les encouragements, enfin à mon très cher oncle feu **Lambert AMOUGOU**.

Remerciements

Je tiens tout d'abord à remercier le professeur **Idrissa SARR** pour la confiance qu'il a portée en mon égard en m'offrant cette opportunité de stage, également pour sa rigeur et ses conseils tout au long de ce travail.
Je tiens également à remercier mes encadreurs le professeur **Aliou BOLY**, le docteur **Modou GUEYE** et le docteur **Ndiouma BAME**, pour leur disponibilité malgré leur emploi de temps chargé, pour les conseils et les critiques tout au long de ce travail.

Mes sincères remerciements sont également exprimés au corps professoral de la section informatique pour sa qualité pédagogique rare et toutes les connaissances qu'il a su me transmettre.

Je tiens à remercier de façon particulière mon camarade **mohamodou NDIAYE** avec qui j'ai réalisé mon stage. C'était pas facile mais on a su se surpasser.
Je remercie tous mes camarades de Master avec qui nous avons pu créer un esprit de partarge de connaissances, particulièrement à **Martial KOULEYE**, **Evrard NGUEMEYOU**, **Naomie NOUMI**, **Junior MEDJEU**, **Iliassou NDAM**, **Asta NIANG**, **Fatou KINE** et **Marie DIOUF**.
Un remerciement particulier à **Carène ASSEMBE**, **Josiane KENGNE**, **Catherine DALLE**, **Gladys NAIBEI**, Dr **Felicite NGUAMGNE**, **Alain WANDJI**, **Damien SAMBO**, **Christian LEUMALE** pour leur soutien lors de la correction de ce travail.
Je remecie tous mes amis particulièrement **Raoul KOLOKO**, **Léonel ESSUTHI**, **Bernard KABENDE**, **Manuela BATTANG**, **Michelle SIMO** et **Cabrel NGAKO**.

Je tiens à remercier aussi tous ceux qui m'ont soutenu de près ou de loin.

Résumé

Un des éléments essentiels pour réaliser la riposte à une pandémie est de faire un suivi épidémiologique constant pour détecter l'arrivée des épidémies et agir sur les leviers d'actions appropriées dans les meilleurs délais. Dans cette perspective, un Système d'Information Décisionnel (SID) est indispensable pour collecter et analyser les données épidémiologiques afin d'éclairer les stratégies spécifiques à mettre en place.

L'objectif de ce travail est de mettre en place un ensemble d'outils pour l'exploitation et la valorisation de données issues de différentes sources. Il s'inscrit dans le contexte du projet ALSO- COVID-19 financé par le CAMES (Conseil Africain et Malgache pour l'Enseignement Supérieur) avec pour objectif principal de mieux comprendre la dynamique de la pandémie mondiale de la maladie de la COVID-19 dans les pays de l'espace CAMES et d'en tirer des leçons pour accroître la résilience de l'Afrique face aux maladies émergentes. Ce projet regroupe plusieurs workpackages dont les activités génèrent des données au format divers.

Dans une perspective d'intégrer toutes ces informations, les enrichir et les mettre dans un format accessible, compréhensible et plus exploitable, nous avons conçu une plate-forme avec deux grandes composantes. La première composante est destinée à la récupération et au stockage des données dans une base de données fédératrice. Pour ce faire, nous avons conçu une base de données NoSQL et implémenté un outil de collecte de données et de publications scientifiques issues de sites d'informations sanitaires. La seconde composante a une orientation d'aide à la décision. Elle propose un Outil d'analyse de l'évolution de l'épidémie et des tendances au niveau du continent africain.

Mots clés : Système d'Information Décisionnel, exploitation et valorisation de données, analyse des données, base de données NoSQL, collecte de données.

Abstract

One of the essential elements in carrying out the response to a pandemic is to do a constant epidemiological monitoring to detect the arrival of epidemics and to act on the appropriate levers of action as soon as possible. From this perspective, a Business Intelligence System (BIS) is essential to collect and analyze epidemiological data in order to inform the specific strategies to be put in place.

The objective of this work is to put in place a set of tools for the use and valuation of data from different sources. It is part of the ALSO- COVID-19 project funded by CAMES (African and Malagasy Council for Higher Education) with the main objective of better understanding the dynamics of the COVID-19 pandemic in the countries of the CAMES space and learn lessons to increase Africa's resilience to emerging diseases. This project brings together several work-packages whose activities generate data in various formats.

With a view to integrating all this information, enriching it and putting it in an accessible, understandable and more usable format, we have designed a platform with two main components. The first component is intended for the retrieval and storage of data in a unifying database. To do this, we designed a NoSQL database and implemented a tool for collecting data and scientific publications on health information sites. The second component has a decision support orientation. It offers an Analysis Tool for the evolution of the epidemic and trends on the level of the African continent.

Keywords: Business Intelligence System, data exploitation and valuation, data analysis, NoSQL database, data collection.

Table des matières

Table des figures

Introduction Générale

Depuis décembre 2019, le monde entier est frappé par une crise sanitaire due à l'apparition d'un nouveau virus, baptisé **SARS-CoV-2**[1], dans la ville de Wuhan en Chine. Très rapidement, ce virus s'est diffusé dans tous les continents. Cependant, il est fort intéressant de constater que sur le taux mondial de contaminés, 98.55% se trouvent hors du continent africain. Aussi, sur les 54 pays constituant l'Afrique, plus de la moitié des contaminés provient seulement des quatre pays que sont l'Égypte, l'Algérie, le Maroc et l'Afrique du Sud[2]. Cette disparité entre l'Afrique et le reste du monde d'une part, ainsi qu'entre les différentes régions du continent africain d'autre part reste encore un grand mystère pour nos scientifiques.
Pour faire face à la pandémie, plusieurs essais thérapeutiques ont été mis sur pied, à l'instar de l'utilisation d'antiparasitaires (hydroxychloroquine) antérieurement décrits comme étant efficaces sur les coronavirus [24] à l'utilisation d'antiviraux [11] ou encore de remèdes traditionnels africains [13]. Cependant, malgré les nombreux essais thérapeutiques, la mortalité en Asie, en Europe et en Amérique reste très élevée et suscite de nombreuses interrogations.

Au regard de tout ce qui précède, nous pouvons dire qu'il est nécessaire d'avoir une meilleure appréhension du profil épidémiologique, physiopathologique et clinique de cette pandémie dans le contexte africain, de même que leurs déterminants socio-culturel, économique et politique, dans l'optique de mettre en place une riposte adaptée et efficace.
Il est aussi important de noter que malgré le fait que la recherche sur le coronavirus ait été très tôt prolifique, on observe un manque criard de données concernant l'Afrique et produites par les chercheurs africains eux-mêmes.

Le projet **ALSO-COVID-19** (African Life Story of COVID-19) est né du consortium constitué de chercheurs et partenaires stratégiques des pays de l'espace **CAMES**[3] ayant pour objectif général de mieux comprendre la dynamique de la COVID-19 dans les pays de l'espace **CAMES**

1. Severe Acute Respiratory Syndrome Coronavirus 2
2. `https\protect\relax\let\let\penalty\@M\hskip0.5\fontdimen2\font=\penalty\@M\hskip0.5\fontdimen2\font:\@beginparpenalty=\@M\relax//covid19.who.int`
3. Conseil Africain et Malgache pour l'Enseignement Supérieur

et d'en tirer des leçons pour accroître la résilience du continent face aux maladies émergentes. Le projet est élaboré sur la base d'une approche intégrée et pluridisciplinaire selon les principes du concept *One Health*. Dans cette perspective, la pandémie de la COVID-19 sera analysée sous plusieurs axes tels que :

- **Les Aspects cliniques, immunologiques et épidémiologiques de la COVID-19** permettant d'évaluer les caractéristiques épidémiologiques et cliniques de l'infection à la COVID-19. Cette étude se déroule au Mali.

- **Les Déterminants environnementaux, socio-culturels, économiques et politiques** avec pour but de déterminer l'influence des facteurs contextuels sur la dynamique évolutive de la COVID-19 dans l'espace **CAMES**, plus particulièrement au Bénin.

- **L'étude génomique du virus SARS-CoV-2** réalisé au Burkina Faso et ayant pour objectif de comprendre l'évolution de la transmission du SARS-CoV-2 pendant la période de la pandémie et de sa propagation dans les pays de l'espace **CAMES**.

- **La Surveillance des coronavirus au niveau des hôtes réservoirs et intermédiaires** que sont les animaux domestiques, de ferme et de brousse. Le but étant d'évaluer le rôle épidémiologique des animaux domestiques dans la chaîne de transmission du SARS-CoV-2 et d'autres virus apparentés dans ledit espace, particulièrement le Gabon.

Pour atteindre tous ces objectifs, le projet a été structuré en plusieurs *workpackages* dont les quatre premiers correspondent aux axes de recherche énumérés plus haut. Ils sont centrés sur l'étude du profil épidémiologique, physiopathologique et clinique de la pandémie et ses déterminants socio-culturel, économique et politique.
Naturellement, de chacun de ces axes de recherche devra découler une importante quantité de données de natures différentes sur la covid-19 en Afrique. Notons en plus que des études sur cette pandémie ont été faites auparavant et de nouvelles ne cessent de s'y ajouter partout dans le monde. On trouve dès lors une bonne quantité de données sur le WEB ouvertes et accessibles aux analyses.

Bien évidemment, un des éléments essentiels pour réaliser une riposte sanitaire efficace est de faire un suivi épidémiologique constant pour détecter l'arrivée d'une épidémie et d'agir sur les leviers d'actions appropriées dans les meilleurs délais. Dans cette logique, un Système d'Information Décisionnel (SID) est indispensable pour collecter et analyser les données épidémiologiques

afin d'éclairer les stratégies spécifiques à mettre en place. Ainsi donc le cinquième workpackage du projet **ALSO-COVID-19** et qui concerne notre mémoire assure la gestion, la valorisation et l'exploitation des données du projet. Il a pour objectif de concevoir et de développer un système d'information intégré de gestion des données du projet en mettant en place une base de données fédératrice pour stocker et consolider les données des différentes études menées par les parties prenantes du projet et concevoir des tableaux de bord pour faire la synthèse et l'analyse des informations pertinentes du projet.

En résumé, ce présent travail entre dans le cadre de la réalisation des activités du workpackage 5 et répond à la question suivante : Quelles connaissances pouvons-nous tirer des données produites par le projet et celles des études préexistantes ?
De façon plus spécifique, il est question de savoir ce que nous pouvons ressortir comme informations à partir des données du projet et celles présentes sur le WEB afin d'accompagner les dirigeants dans la prise de décisions éclairées et efficaces qui tiennent parfaitement compte des réalités de l'Afrique en matière de prévention et de lutte.

De ce fait, l'objectif principal de ce travail est de mettre en place un ensemble d'outils permettant le stockage, la consolidation, l'exploitation et la valorisation de données issues des différents axes de recherches du projet et celles déjà existantes sur le WEB.

Afin de traiter notre sujet et pallier les interrogations sus évoquées, nous avons établi un plan de recherche bipartite. La première partie gère la collecte et le stockage des données du projet dans une base de données fédératrice. La seconde partie est consacrée à l' aide à la décision.

La suite de ce mémoire est organisée comme suit. Le premier chapitre décrit la conception de notre base de données fédératrice et les techniques d'intégration et de consolidation des données du projet. Dans le second chapitre, nous présentons les différents moyens et techniques que nous avons utilisés pour l'enrichissement des données du projet par celles récoltées du WEB. Dans le troisième chapitre, nous abordons les méthodes et technologies utilisées pour concevoir nos outils d'exploitation et valorisation des données du projet. En fin, nous présentons une conclusion qui va résumer tout ce que nous avons eu à aborder et dressons un ensemble de perspectives associées à ce travail.

Chapitre 1

Stockage et consolidation des données du projet

1.1 Introduction

Le stockage et la représentation des données font partie des éléments centraux de la réalisation du projet ALSO-COVID-19. Les données traitées et analysées dans ce projet proviennent de plusieurs sources que sont le Web et les quatre axes de recherche du projet à savoir:

- **Les Aspects cliniques, immunologiques et épidémiologiques de la COVID-19** permettant d'évaluer les caractéristiques épidémiologiques et cliniques de l'infection à la COVID-19. Cette étude se déroule au Mali.

- **Les Déterminants environnementaux, socio-culturels, économiques et politiques** avec pour but de déterminer l'influence des facteurs contextuels sur la dynamique évolutive de la COVID-19 dans l'espace **CAMES**, plus particulièrement au Bénin.

- **L'étude génomique du virus SARS-CoV-2** réalisé au Burkina Faso et ayant pour objectif de comprendre l'évolution de la transmission du SARS-CoV-2 pendant la période de la pandémie et de sa propagation dans les pays de l'espace **CAMES**.

- **La Surveillance des coronavirus au niveau des hôtes réservoirs et intermédiaires** que sont les animaux domestiques, de ferme et de brousse. Le but étant d'évaluer le rôle épidémiologique des animaux domestiques dans la chaîne de transmission du SARS-CoV-2 et d'autres virus apparentés dans ledit espace, particulièrement le Gabon.

Dans le but de faciliter l'accès et le partage de ces données, nous allons mettre en oeuvre une base de données fédératrice. Vu le fait que le format des données varie en fonction de la source de provenance[1], des méthodes de consolidation des données ont été implémentées afin de réussir à les stocker de façon intégrée dans la même base de données.
Nous abordons ces méthodes dans ce chapitre divisé en deux parties. La première présente les techniques de conception de notre schéma de stockage alors que la seconde détaille les différentes méthodes que nous avons utilisées pour consolider les données des différentes sources.

1.2 Conception du schéma de stockage

Il est question dans cette partie de proposer un schéma de stockage permettant de faciliter la sauvegarde et l'accès aux données du projet. Ce schéma prend en compte le fait que le format ou la structure de la donnée diffère en fonction de la source. Il doit nous permettre ainsi de stocker les données, ceci sans toutefois qu'on ait besoin de connaître la source de provenance.
Nous présentons dans la suite de cette partie, les éléments fondamentaux pour la conception et l'implémentation de notre schéma de stockage. Nous parlerons d'abord des concepts relatifs aux bases de données. A cela, nous ajouterons les raisons du choix du type de base de données que nous avons utilisée dans ce projet. En dernier point, nous procéderons à la modélisation de notre schéma de stockage.

1.2.1 Base de données

Une Base de données permet de stocker et de retrouver des informations en rapport avec un thème ou une activité. Celles-ci peuvent être de nature différente et plus ou moins reliées entre elles. Il existe actuellement plusieurs catégories de bases de données parmi lesquellles les bases de données SQL[2] et les bases de données NoSQL.

Le SQL vs Le NoSQL

Les systèmes de bases de données traditionnels sont basés sur le modèle relationnel. Ceux-ci sont largement connus sous le nom de bases de données SQL dénommées d'après le langage par lequel elles interagissent [20]. Il s'agit de systèmes bien établis et très utilisés pour stocker

1. les workpackages ont des formulaires d'enquête et des rapports d'expérimentation aux formats variés et diversifiés
2. Le terme *Bases de données relationnelles* est aussi très souvent usité.

et interroger de grands ensembles de données structurées. Ils sont basés sur le modèle relationnel, c'est-à-dire un ensemble de tableaux représentant chacun un ensemble d'objets aux caractéristiques similaires [14].
Ces systèmes ont été utilisés avec succès dans le traitement des transactions en ligne (OLTP) et des applications de traitement analytique en ligne (OLAP). Cependant, avec l'explosion des médias sociaux, le contenu axé sur l'utilisateur a connu une croissance rapide et a augmenté le volume et le type des données produites. En effet, avec la croissance de l'accès à Internet et la disponibilité d'un stockage bon marché, d'énormes quantités de données structurées, semi-structurées et non structurées sont capturées et stockées par une variété d'applications [21]. En plus, de nouvelles sources de données, telles que des capteurs, des GPS et d'autres systèmes de surveillance, génèrent régulièrement d'énormes volumes de données. Ce grand volume de données, également appelés *Big data* [31], a introduit de nouveaux défis et de nouvelles opportunités pour le stockage, la gestion, l'analyse et l'archivage des données. Dès lors, les bases de données relationnelles qui nécessitent une définition de schéma et des références relationnelles rendant rigide leur utilisation ne sont pas très adaptées pour gérer la variété des données générées.

A leur place, les bases de données NoSQL sont couramment utilisées pour gérer des données structurées et non structurées telles que des documents. Au cours des dernières années, les bases de données non relationnelles ont considérablement augmenté en popularité. D'après [21], les bases de données NoSQL sont devenues les références pour l'exploitation du Big Data. Cependant, afin d'obtenir une meilleure évolutivité, elles ne garantissent pas les propriétés **ACID** (**A**tomicité, **C**ohérence, **I**solation et **D**urabilité) standard qui sont généralement fournies par les bases de données relationnelles. Mais face au volume, à la vélocité et à la variété des données[3], le modèle relationnel peut difficilement lutter contre cette vague de données. Ainsi, le NoSQL s'impose naturellement dans ce contexte en proposant une nouvelle façon de gérer les données ne reposant pas sur le paradigme relationnel; d'où le terme *NoSQL* signifiant *Not Only SQL*.

En somme, les bases de données NoSQL ont été créées en réponse aux limitations de la technologie de bases de données relationnelles. Comparées aux bases de données relationnelles, les bases de données NoSQL sont plus évolutives et offrent des performances supérieures, et leur

3. C'est le concept des *3V*

modèle de données corrige plusieurs faiblesses du modèle relationnel.

Dans un contexte de bases de données, il est préférable d'avoir un langage de haut niveau pour interroger les données plutôt que tout exprimer en Map/Reduce [16]. Toutefois, avoir un langage de haut niveau comme SQL ne facilite toujours pas la manipulation des données. Et c'est en ce sens qu'on doit vraiment parler de *Not Only SQL*. Ainsi, le NoSQL est à la fois une autre manière d'interroger les données, mais aussi de les stocker. Les besoins de stockage et de manipulation dans le cadre d'une base de données sont variables et dépendent principalement de l'application que vous souhaitez intégrer. Pour cela, différentes familles de bases NoSQL existent : Clé/Valeur, colonnes, documents, graphes.

Chacune de ces familles répond à des besoins très spécifiques. Le tableau 1.1 fait une comparaison entre les différents types de bases de données en fonction du besoin de l'utilisateur.

TAB. 1.1 – *tableau de comparaison des bases données SQL et NOSQL*

SGBD / Caractéristiques	**SQL**	**NOSQL**			
	PostgreSQL, MySQL, MS SQL	Clé Valeur (Riak,Redis)	Document (Mongodb, CouchBase, couchDB)	Colonne (Cassandra, apacheHBASE)	Clé Valeur (Riak, Redis)
Gros volumes de données à traiter (Big Data)	Peu adapté	Très adapté	Très adapté	Très adapté	Très adapté
Besoins de richesse fonctionnelle	Très adapté	Peu adapté	Moyennement adapté	Peu adapté	Peu adapté
Haute disponibilité des données et tolérance aux pannes	Moyennement adapté	Très adapté	Très adapté	Très adapté	Très adapté
Capacités de montée en charge	Moyennement adapté	Très adapté	Très adapté	Très adapté	Très adapté
Traitement de données non structurées	Peu adapté	Très adapté	Très adapté	Très adapté	Très adapté
Traitement de données structurées	Très adapté	Peu adapté	Moyennement adapté	Peu adapté	Moyennement adapté
Gestion des données hétérogènes	Moyennement adapté	Très adapté	Très adapté	Très adapté	Très adapté
Respect de la structure et Cohérence des données	Très adapté	Peu adapté	Moyennement adapté	Peu adapté	Moyennement adapté

.

Choix du Système de gestion de base de données (SGBD)

Notre SGBD doit permettre de faciliter la sauvegarde et l'accès aux données du projet tout en prenant en compte le fait que le format (c-à-d., la structure) des données diffère en fonction des sources. L'une des sources des données est le premier axe de recherche du projet où les données collectées portent sur les aspects cliniques, immunologiques et épidémiologiques de la Covid-19. Les données reçues de la seconde source portent sur l'influence des facteurs contextuels (c-à-d., les déterminants environnementaux, socio-culturels, économiques et politiques). Les données de la troisième source quant à elles portent sur l'étude génomique du virus SARS-COV-2. Celles de la quatrième source dérivent de l'étude de la surveillance des coronavirus au niveau des hôtes réservoirs et intermédiaires que sont les animaux domestiques, de ferme et de brousse. Enfin, la dernière source de données n'est rien d'autre que le web où nous comptons collecter les articles et revues contenant les résultats des précédentes recherches faites sur la pandémie de la Covid-19. Toutes ces données collectées ont des points en commun mais conservent chacune des particularités.

Dans l'optique de stocker les données sans avoir à prendre en compte la source de provenance et après avoir passé en revue les différents types de base de données existants, nous avons porté notre choix sur **MongoDB** pour la gestion des données du projet.
MongoDB est un Système de gestion de base de données gratuite et *open source*, orientée document et multi-plateforme [16]. Les données stockées sont organisées en documents JSON équivalents aux enregistrements d'une table d'une base de données relationnelle avec des champs représentant les colonnes. Les documents sont regroupés en collections équivalentes aux tables relationnelles. Le stockage des données dans des documents permet à MongoDB de disposer d'un système de stockage flexible. Ce qui signifie que les objets stockés ne doivent pas nécessairement avoir la même structure (c-à-d., les mêmes champs).
Un des gros avantages de MongoDB par rapport au SQL, est sa capacité à gérer des systèmes de données complexes. On peux avoir des listes, des objets encapsulés sans avoir de soucis. Ce fonctionnement facilite grandement le développement d'applications qui gèrent beaucoup de données. De plus, aucune migration de schéma. Étant donné que MongoDB est sans schéma, votre code définit votre schéma. MongoDB possède également certaines fonctionnalités d'optimisation, qui distribuent les collections de données, constituant ainsi un système plus équilibré et plus performant. MongoDB dispose d'une grande communauté fortement active et d'une bonne documentation.

D'après le classement mensuel des systèmes de gestion de bases de données de DB-Engines[4] du mois de Mai 2021, MongoDB occupe la première place parmi les bases de données non relationnelles utilisées.

1.2.2 Méthodes et outils de modélisation

L'évolution des techniques de programmation a toujours été dictée par le besoin de concevoir et de maintenir des applications toujours plus complexes. Modéliser un système avant sa réalisation, permet de mieux comprendre son fonctionnement.

Méthodes de modélisation

Afin de proposer une représentation harmonisée et compréhensive des différentes entités intervenant dans notre plateforme ainsi que des interactions qui peuvent exister entre elles, nous avons choisi comme méthode de modélisation le langage de modélisation **UML**.
UML [15] est l'acronyme anglais de *Unified Modeling Language*. On le traduit par *Langage de modélisation unifié* . La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. il constitue un moyen d'exprimer des modèles objet en faisant abstraction de leur implémentation. *UML 2.5* comporte quatorze diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information [15]. Il est couramment utilisé en développement logiciel et en conception orientée objet. UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle.

Dans le cadre de la modélisation de notre système, nous nous contenterons du diagramme de classes afin de modéliser la structure générale de notre plateforme. En effet, le diagramme de classes permet de représenter les objets du système et leurs interactions. Ce diagramme fait partie de la vue statique d'UML car il fait abstraction des aspects temporels et dynamiques. Une classe décrit les propriétés, le comportement et le type d'un ensemble d'objets. Les classes peuvent être liées entre elles grâce au mécanisme d'association qui permet de mettre en évidence des relations entre elles. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

4. https://db-engines.com/en/ranking/document+store

Outils de modélisation

Draw.io est une application de création de diagrammes et schémas sous licence Apache disponible sous Windows, MacOs, Linux, sous forme d'application web et intégrée à des services cloud tels NextCloud ou Google Drive. Dans ce projet nous utilisons la version web disponible en ligne.

Draw.io propose différentes techniques de modélisation, chacune accessible aux informaticiens de tout niveau, parmi elles : Merise, UML, Data Warehouse, et processus métiers. Simple d'utilisation, personnalisable et dotée d'une interface intuitive, cette application optimise les productivités individuelle et collective. Bien sûr, il existe d'autres outils, notamment **PowerDesigner** (anciennement PowerAMC) qui est un logiciel de conception créé par la société SAP qui permet de modéliser les traitements informatiques et leurs bases de données associées.

1.2.3 Présentation du schéma de stockage du projet

Dans cette section, nous présentons le diagramme de classes conçu pour la réalisation de notre solution. Le diagramme de classes est une description statique du système focalisé sur le concept de classe et d'association. Une *classe* représente un ensemble d'objets qui possèdent des propriétés similaires et des comportements communs, décrits en terme d'attributs et d'opérations. Une *association* consiste à présenter les liens entre les instances de classe. Le schéma de stockage que nous proposons est un schéma très proche du schéma en étoile utilisé pour la modélisation des entrepôts de données [19].

L'objectif visé ici est d'implémenter la base de données avec un modèle évolutif et permettant une variété et un grand volume de données. La conception de la base de données est faite de sorte qu'on ait un ensemble de classes centrées autour d'un objet abstrait et qui permet de représenter les humains, les animaux, l'environnement et mieux, tout autre concept ayant un sens dans une approche one-health. Ainsi, dans notre schéma nous avons une classe centrale, à laquelle sont reliées toutes les autres classes. Les informations extraites de nos sources sont organisées dans ces autres classes. A partir de la classe centrale, nous pouvons reconstituer les données pour une restitution complète de celles-ci.

La figure 1.1 représente le diagramme de classes. Elle décrit les interactions existantes entre les différents objets de ce module. Ce diagramme permet d'illustrer toutes les informations dont on a besoin pour implémenter les données de chaque workpackage du projet. Les données collectées au niveau des différents workpackages sont stockées dans les classes de périphériques de couleur rose. Chaque classe contient des informations bien définies et selon une thématique très précise. Toutes ces classes sont liées à la classe *Objet* qui est la classe centrale. Cette classe permet de reconstituer la donnée. Une instance de cette classe peut être de type *Personne* si les informations stockées sont celles d'une personne, de type *Animal* si les informations sont celles d'un animal, ou de type *Virus*.
Le tableau 1.2 fait un récapitulatif de toutes les classes de notre schéma et leurs différents attributs.

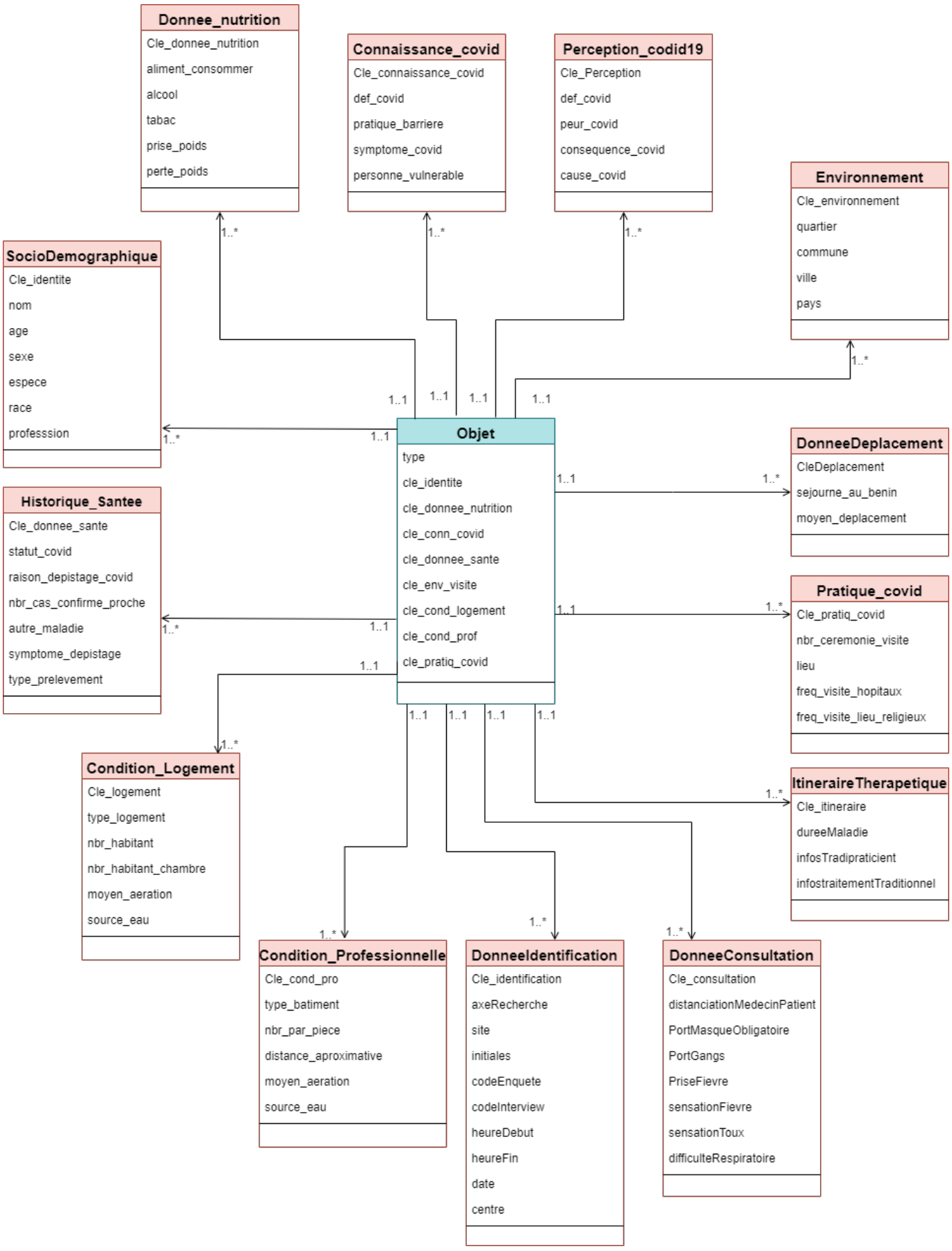


FIG. 1.1 – *schéma du diagramme de classes*

TAB. 1.2 – *Tableau récapitulatif des classes et leurs attributs*

Classes	Attributs	Commentaires
Objet	type	peut être un virus, animal,personne
	#Cle_nutrition	Clé étrangère de la classe nutrition
	#cle_conn_covid	Clé étrangère la classe connaissance du Covid
	#cle_donnee_sante	Clé étrangère de la classe historique de santée
	#cle_env_resid	Clé étrangère de la classe environnement de résidence
	#cle_env_visite	Clé étrangère de la classe environnement visité
	#cle_cond_logement	Clé étrangère de la classe condition de logement
	#cle_cond_prof	Clé étrangère de la classe condition professionnel
	#cle_identification	Clé étrangère de la classe regroupant les données identification
	#cle_pratiq_covid	Clé étrangère pratique covid19
	#cle_identite	Clé étrangère données de la classe SocioDemographique
DonneeDeplacement	cle_Deplacement	Clé primaire
	sejour_au_benin	
	moyen_deplacement	

HistoriqueSante	cle_donnee_sante	Clé primaire
	statut_covid	
	raison_depistage_covid	
	nbr_cas_confirme_proche	
	autre_maladie	
	symptome_depistage	
	sype_prelevement	
SocioDemographique	cle_identite	Clé primaire
	nom	
	age	
	sexe	
	espece	
	race	
	profession	
Condition_logement	cle_logement	Clé primaire
	type_logement	
	nbr_habitant	
	nbr_habitant_chambre	
	moyen_aeration	
Condition_Professionnelle	cle_cond_prof	Clé primaire
	type_batiment	
	nbr_par_piece	
	distance_approximative	
	moyen_aeration	
	Source_eau	
DonneeIdentification	cle_identification	Clé primaire
	axeRecherche	·
	site	
	initiales	
	codeEnquete	
	codeInterview	
	heureDebut	
	heureFin	
	date	
	centre	
DonneeConsultation	cle_consultation	Clé primaire
	distanciationMedecinPatient	
	portMasqueObligatoire	
	portGangs	
	priseFievre	
	sensationFievre	
	sensationToux	
	difficulteRespiratoire	
Pratique_covid	cle_pratiq_covid	Clé primaire
	nbr_ceremonie_visite	
	lieu	
	freq_visite_hopitaux	
	freq_visite_lieu_religieux	

ItineraireTherapeutique	cle_Itineraire	Clé primaire
	dureeMaladie	
	infoTradipraticient	
	infoTraitementTraditionnel	
Environnement	cle_environnement	Clé primaire
	quartier	
	commune	
	ville	
	pays	
Perception_covid19	cle_Perception	Clé primaire
	defCovid	
	Peur_covid	
	consequence_covid	
	cause_covid	
connaissance_covid	cle_connaissance_covid	Clé primaire
	defCovid	
	Pratique_barrière	
	symptome_covid	
	personne_vulnerable	
Donnee_nutrition	cle_donnee_nutrition	Clé primaire
	aliment_consomer	
	alcool	
	tabac	
	prise_poids	
	perte_poids	

A ce niveau, nous avons modélisé et conçu notre base de données fédératrice qui nous permettra de stocker toutes les données des différents workpackages du projet. Les données étant hétérogènes, il s'avère nécessaire d'effectuer un certain nombre de traitements, ceci pour les homogénéiser et réussir le stockage dans la base de données. Dans la section suivante, nous verrons les différentes méthodes de consolidation de données que nous avons implémentées pour arriver à stocker nos données.

1.3 Consolidation des données du projet

Dans le section précédente, nous avons présenté la structure de notre base de données. Celle-ci nous permettra de stocker les données des multiples sources du projet, aussi bien internes que externes. Nous avons conçu un schéma de stockage central. Ceci nous permettant de stocker la donnée sans toutefois avoir besoin de connaître au préalable sa source. Ainsi, pour réussir à effectuer ce stockage central, il est nécessaire pour nous, d'effectuer des traitements sur les données après leur extraction.
Les traitements faits sur ces données diffèrent en fonction de la source. Les données provenant des sources internes (les axes de recherche du projet) subissent des transformations différentes de celles extraites des sources externes au projet (le Web). Dans cette section, nous parlerons essentiellement de l'ETL (Extraction, Transformation and Load) effectué sur les données extraites des sources internes.

1.3.1 Définition de l'ETL

L'ETL est un processus par lequel les données sont extraites, transformées puis chargées dans un emplacement cible. Il s'agit d'un processus d'intégration de données qui permet de transformer les données brutes d'un systèmes source, de les préparer pour une utilisation en aval et de les envoyer vers une base de données, un entrepôt de données ou un serveur cible [3]. L'objectif d'un ETL est de rendre des données hétérogènes compatibles entre elles, mais aussi avec la base de données de référence de l'organisation, pour une meilleure analyse. Pour cela, l'ETL structure, nettoie, compile puis agrège les données extraites des différents composants du système de l'entreprise [7].
Le processus d'ETL permet d'effectuer des synchronisations massives d'informations entre bases de données. Il est très utile pour le traitement de vastes ensembles de données hétérogènes dans le cadre de l'analytique du Big Data et de l'informatique décisionnelle. Les processus d'ETL sont responsables de l'extraction des données à partir de sources de données opérationnelles hétérogènes, de leur transformation (conversion, nettoyage,etc.) et de leur chargement dans les bases de données décisionnelles. Par conséquent, il est clair que la conception et la maintenance de ces processus sont un facteur clé de succès dans les projets décisionnels [29].

Dans le cadre d'un projet comme le notre où les données proviennent de plusieurs sources, l'ETL nous permettra d'abord de les extraire, ensuite de les rendre homogènes en effectuant

un certain nombre de tâches, avant de les stocker dans la base. D'après [29], la conception d'un processus d'ETL est généralement composée de six tâches à savoir:

1. Sélectionner les sources à extraire : les sources de données (généralement plusieurs sources de données hétérogènes) à utiliser dans le processus ETL sont définies.
2. Transformer les sources : Une fois les données extraites des sources, elles peuvent êtres transformées ou de nouvelles données peuvent êtres dérivées. Certaines tâches de cette étape courante sont: filtration des données, conversion des codes, calcul des valeurs dérivées, transformation entre différents formats de données, génération automatique de numéros de séquence (clés de substitution),etc.
3. Joindre les sources : différentes sources peuvent êtres jointes afin de charger ensemble les données dans une cible unique.
4. Sélectionner la cible à charger : la cible (ou les cibles) à charger est sélectionnée.
5. Mapper les attributs sources aux attributs cibles : les attributs (champs) à extraire des sources de données sont mappés sur les attributs cibles correspondants.
6. Charger les données : la cible est remplie avec les données transformées.

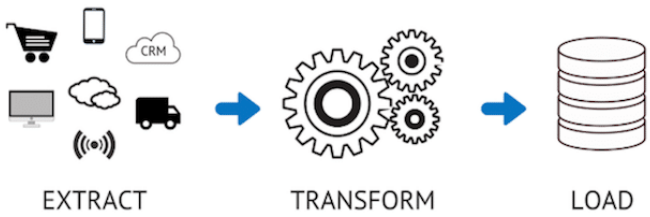


FIG. 1.2 – *schéma d'un processus d'ETL*

1.3.2 Outils d'ETL

Les outils d'ETL sont des logiciels disposant d'un environnement de développement intégré (IDE), mettant à disposition tous les éléments nécessaires au développement et à la mise en production, de flux de transformation et chargement de données. Il existe de nombreuses solutions disponibles sur le marché. Dans la suite de cette partie, nous parlerons essentiellement de trois outils. Il s'agit de SSIS, Talend OpenStudio et Pentaho. Nous avons fait le choix de ces trois outils pour deux raisons. D'abord parce qu'ils font partie des outils les plus utilisés sur le marché. Aussi parce que nous les utilisons régulièrement dans nos travaux académiques.

Talend OpenStudio

Talend OpenStudio est un logiciel opensource appartenant à l'éditeur de logiciel français *Talend*, spécialisé dans le développement de solutions propriétaires et des produits opensource. Ce logiciel met à notre disposition un outil d'ETL très complet et très simple à manipuler, grâce à son interface graphique ne nécessitant pas de code. Il permet aux utilisateurs de créer des flux de données de manière intuitive en glissé/déposé, génère par la suite et de façon automatique, un code en langage JAVA. Il propose une grande variété de connecteurs avec les principaux systèmes de gestion de base de données, de logiciels CRM (Customer Relationship Management) et suites marketing du marché. Il offre la possibilité aux entreprises de développer leurs propres fonctionnalités afin de répondre au mieux à leurs besoins. Talend OpenStudion dispose d'une documentation complète et une communauté particulièrement active et engagée.

SSIS

SSIS signifie SQL Server Intégration Services. *Microsoft* avait conçu cette solution pour concurrencer dans le domaine de la migration de données, de l'ETL et de la transformation. Elle peut exécuter des solutions complexes telle que la lecture de données à partir de différentes sources, l'analyse et le nettoyage des données, l'exécution de processus d'ETL pour mettre à jour les entrepôts de données, l'écriture de données dans différentes sources et l'envoi d'e-mails à une personne ou groupe de personnes particuliers [6]. Ce logiciel comprend un ensemble d'outils pour développer et tester des programmes d'intégration appelés outils de données SQL Server. Le processus d'ETL dans le logiciel SSIS est principalement effectué par des packages [25]. Un package est une collection de variables, de flux de contrôle et de flux de données. SSIS nous permet également d'exécuter les flux de données de façon parallèle. Un inconvénient est qu'il ne fonctionne pas sur un environnement autre que *windows*.

Pentaho Data Intégration

Pentaho est une plate-forme décisionnelle de Business Intelligence open-source, spécialisée dans la gestion et l'analyse des données d'entreprise. Le produit d'intégration de données de cet éditeur est **Pentaho Data Intégration**, autrefois appelé **Kettle**. Ce logiciel permet de concevoir des projets d'intégration et de transformation de flux de données pour les entreprises. Il fournit les capacités d'extraction, de transformation et de chargement qui facilitent le processus de capture, de nettoyage et de stockage des données à l'aide d'un format uniforme et cohérent, accessible et pertinent pour les utilisateurs finaux et les technologies *loT*. Il sépare les données traitées des données en cours de traitement. Cette approche appelé *metadata driven* permet de catégoriser les données en fonction de leur contenu plutôt que de leur lieu de provenance [9]. Le logiciel Pentaho Data Intégration est développé en langage de programmation JAVA et bénéficie d'une forte adaptabilité aux SGBD des entreprises. Il est adapté pour les ETI (Entreprises de Taille Intermédiaire), les PME (Petites et Moyennes Entreprises) et les développeurs indépendants.

1.3.3 Solution de consolidation des données du projet

Il était question dans ce module, de proposer une solution nous permettant de charger notre base de données fédératrice, à partir des données extraites des différentes sources du projet. Dans les sections précédentes, nous avons eu un bref aperçu de ce que représente un processus d'ETL, à quoi il sert et quelques outils nous permettant de le réaliser facilement. Dans cette section, nous présenterons la solution d'ETL implémentée dans le cadre de ce projet, tout en commençant par donner les raisons du choix de notre outil d'implémentation.

Choix de l'outil d'ETL

Nous avons vu dans la section précédente qu'il existait un bon nombre d'outils dont le but principal est de faciliter la réalisation d'intégration de données dans les entrepôts de données. Ces outils nous offrent plusieurs avantages, notamment le fait qu'ils disposent de connecteurs de bases de données, de web services et de fichiers plats prêts à l'emploi. Ils offrent une représentation graphique des flux et opérations. Ils facilitent la maintenance, l'évolution de l'ETL et sont extensibles à l'aide de scripts. Mais néanmoins, Ils nécessitent des développement de connaissances sur l'outil utilisé ou nécessitent un temps d'apprentissage non négligeable. De plus, le traitement de fichiers plats est beaucoup plus complexe, surtout pour un important volume de données.

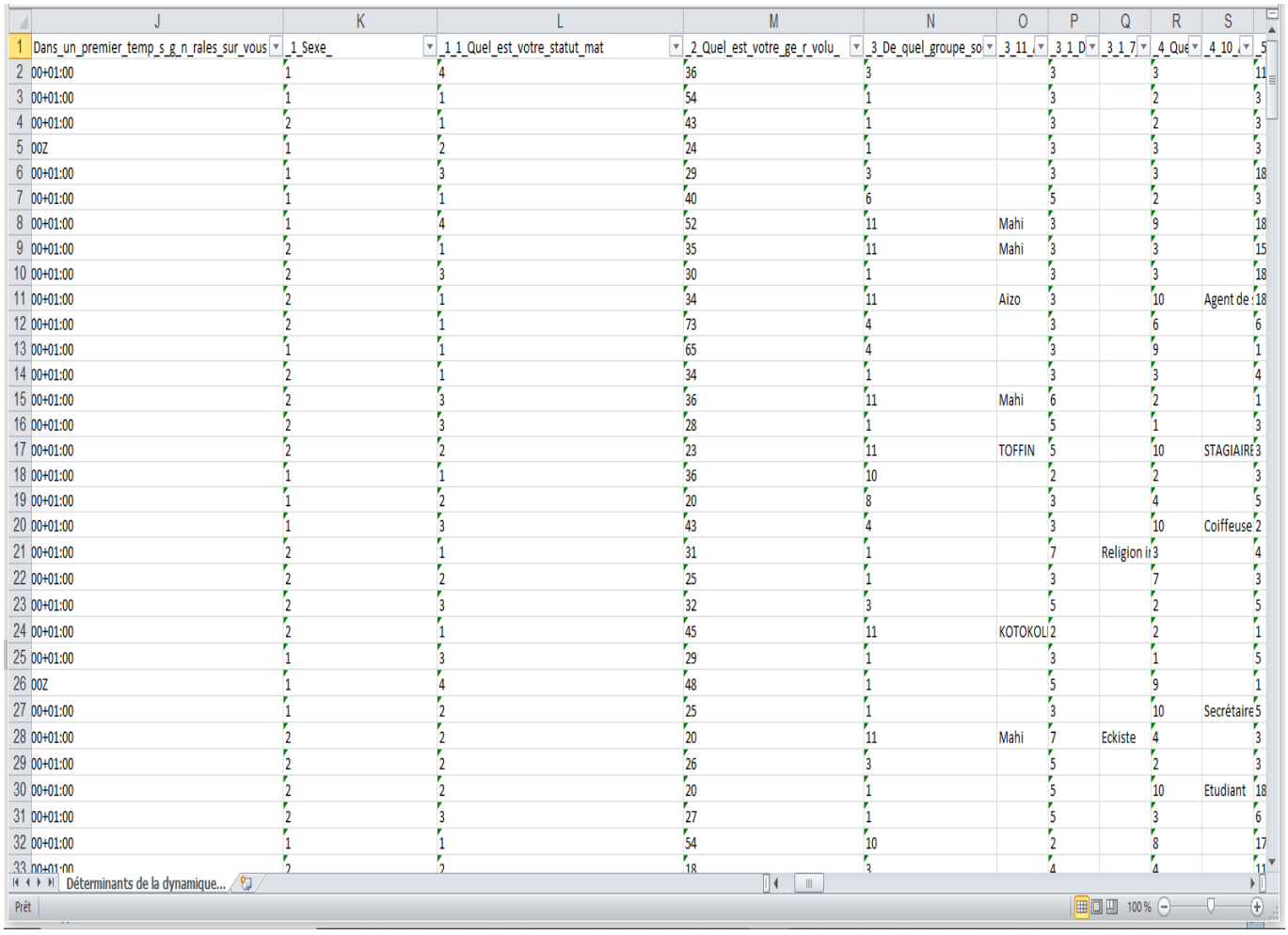
Ainsi, pour la réalisation de l'ETL de notre projet, nous utilisons l'approche scriptée. Elle consiste à développer l'outil nécessaire à la mise en place du traitement d'ETL, à l'aide d'un ou plusieurs langages de programmation. Cette approche renvoie généralement à une combinaison de procédures stockées au niveau des bases de données, ainsi que des différents scripts nécessaires au transport des données et transformations complexes. L'avantage de cette approche est qu'elle permet d'utiliser les langages que les équipes maîtrisent déjà, sans apprentissage et médiation d'un outil tiers. De plus, le traitement des fichiers plats est plus simplifié, surtout lorsqu'on utilise un langage de programmation facilitant l'analyse de données.

Présentation de la solution d'ETL

Les scripts que nous avons réalisés pour l'implémentation de notre processus d'ETL sont écrits en langage de programmation Python. Ce langage offre des librairies spécialisées dans l'analyse et le traitement de données, tel que *Pandas* et *Numpy*.

Les données recueillies au niveau des différents axes de recherche sont des résultats d'interviews faites à partir de formulaires de questions/réponses. Ces résultats sont stockés dans des fichiers plats. La lecture des données directement depuis ces fichiers peut paraître incompréhensible comme on peut voir à la figure 1.3. Ainsi, il s'avère nécessaire de faire des traitement après extraction, avant de charger dans la base de données. Nous utilisons Pandas pour extraire les données des fichiers. Celle-ci simplifie la lecture de fichiers de tout type: csv, xlsx, txt, json, etc. Les données lues sont chargées dans des *dataframes*. Ce sont des structures de données permettant de stocker les données en deux dimensions: lignes et colonnes.

Pandas nous offre des fonctions prédéfinies, nous permettant de traiter les colonnes de dataframes de façon très rapide et efficace. Nous utilisons ces fonctions pour nettoyer les données de nos dataframes. Nous faisons usage de la librairie *Pymongo* pour nous connecter à notre base de données. Cette librairie nous offre des fonctions déjà prédéfinies permettant d'effectuer toutes nos requêtes vers la base de données. La figure 1.4 nous présente les données après traitement.



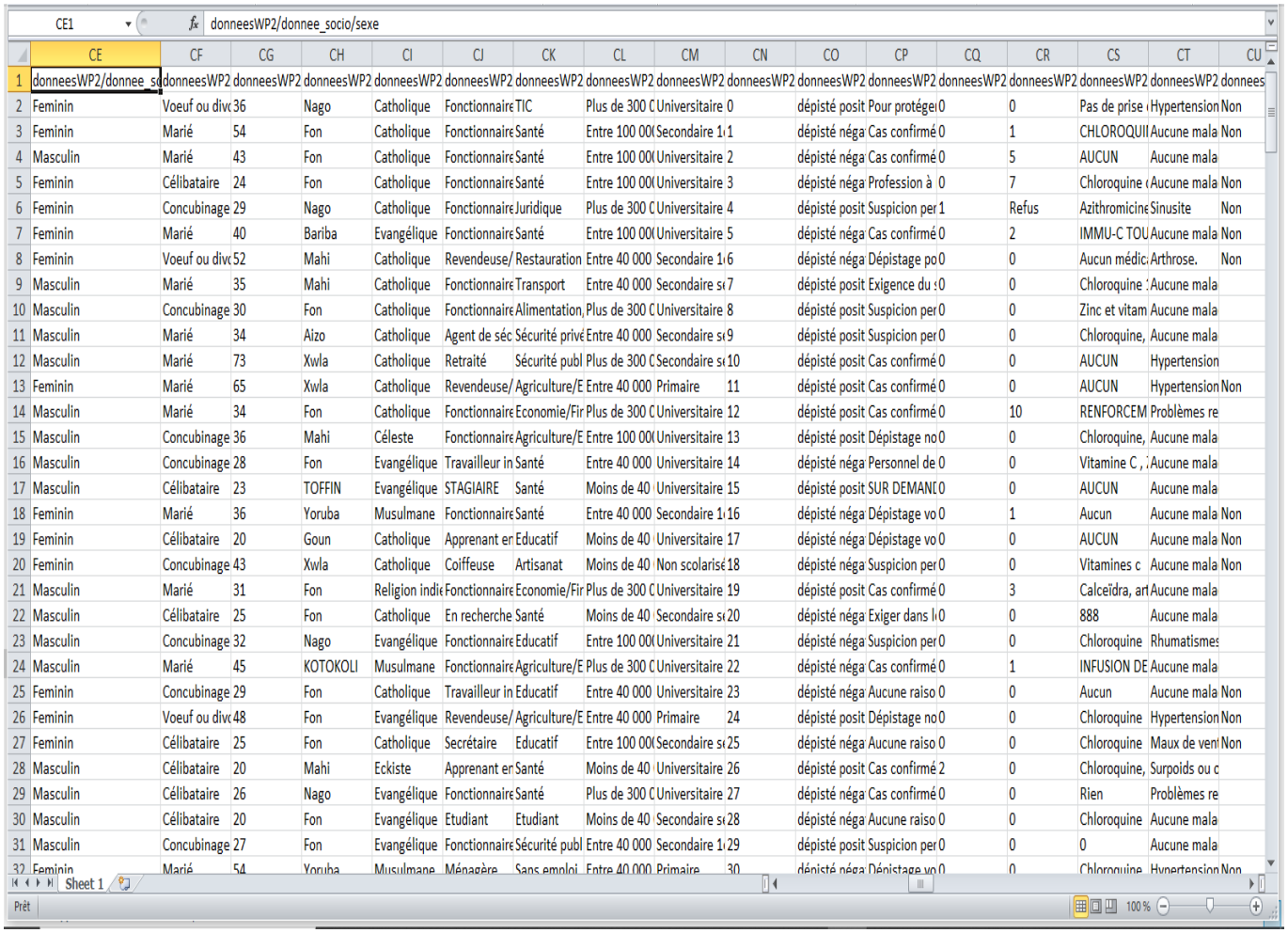
	J	K	L	M	N	O	P	Q	R	S	
1	Dans_un_premier_temp_s_g_n_rales_sur_vous	_1_Sexe_	_1_1_Quel_est_votre_statut_mat	_2_Quel_est_votre_ge_r_volu_	_3_De_quel_groupe_so	_3_11_	_3_1_D	_3_1_7	_4_Que	_4_10_	_5
2	00+01:00	1	4	36	3		3		3		11
3	00+01:00	1	1	54	1		3		2		3
4	00+01:00	2	1	43	1		3		2		3
5	00Z	1	2	24	1		3		3		3
6	00+01:00	1	3	29	3		3		3		18
7	00+01:00	1	1	40	6		5		2		3
8	00+01:00	1	4	52	11	Mahi	3		9		18
9	00+01:00	2	1	35	11	Mahi	3		3		15
10	00+01:00	2	3	30	1		3		3		18
11	00+01:00	2	1	34	11	Aizo	3		10	Agent de :	18
12	00+01:00	2	1	73	4		3		6		6
13	00+01:00	1	1	65	4		3		9		1
14	00+01:00	2	1	34	1		3		3		4
15	00+01:00	2	3	36	11	Mahi	6		2		1
16	00+01:00	2	3	28	1		5		1		3
17	00+01:00	2	2	23	11	TOFFIN	5		10	STAGIAIRE	3
18	00+01:00	1	1	36	10		2		2		3
19	00+01:00	1	2	20	8		3		4		5
20	00+01:00	1	3	43	4		3		10	Coiffeuse	2
21	00+01:00	2	1	31	1		7	Religion i			4
22	00+01:00	2	2	25	1		3		7		3
23	00+01:00	2	3	32	3		5		2		5
24	00+01:00	2	1	45	11	KOTOKOLI	2		2		1
25	00+01:00	1	3	29	1		3		1		5
26	00Z	1	4	48	1		5		9		1
27	00+01:00	1	2	25	1		3		10	Secrétaire	5
28	00+01:00	2	2	20	11	Mahi	7	Eckiste	4		3
29	00+01:00	2	2	26	3		5		2		3
30	00+01:00	2	2	20	1		5		10	Etudiant	18
31	00+01:00	2	3	27	1		5		3		6
32	00+01:00	1	1	54	10		2		8		17
33	00+01:00	2	2	18	3		4		4		11

Déterminants de la dynamique...

Prêt

100 %

FIG. 1.3 – *Données avant l'ETL*



	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU
1	donneesWP2/donnee_s	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donneesWP2	donnees
2	Feminin	Voeuf ou divc	36	Nago	Catholique	Fonctionnaire	TIC	Plus de 300 C	Universitaire	0	dépisté posit	Pour protégei	0	0	Pas de prise (	Hypertension	Non
3	Feminin	Marié	54	Fon	Catholique	Fonctionnaire	Santé	Entre 100 00(	Secondaire 1(	1	dépisté néga	Cas confirmé	0	1	CHLOROQUII	Aucune mala	Non
4	Masculin	Marié	43	Fon	Catholique	Fonctionnaire	Santé	Entre 100 00(	Universitaire	2	dépisté néga	Cas confirmé	0	5	AUCUN	Aucune mala	
5	Feminin	Célibataire	24	Fon	Catholique	Fonctionnaire	Santé	Entre 100 00(	Universitaire	3	dépisté néga	Profession à	0	7	Chloroquine (	Aucune mala	Non
6	Feminin	Concubinage	29	Nago	Catholique	Fonctionnaire	Juridique	Plus de 300 C	Universitaire	4	dépisté posit	Suspicion per	1	Refus	Azithromicine	Sinusite	Non
7	Feminin	Marié	40	Bariba	Evangélique	Fonctionnaire	Santé	Entre 100 00(	Universitaire	5	dépisté néga	Cas confirmé	0	2	IMMU-C TOU	Aucune mala	Non
8	Feminin	Voeuf ou divc	52	Mahi	Catholique	Revendeuse/	Restauration	Entre 40 000	Secondaire 1(	6	dépisté néga	Dépistage po	0	0	Aucun médic	Arthrose.	Non
9	Masculin	Marié	35	Mahi	Catholique	Fonctionnaire	Transport	Entre 40 000	Secondaire s(	7	dépisté posit	Exigence du s	0	0	Chloroquine :	Aucune mala	
10	Masculin	Concubinage	30	Fon	Catholique	Fonctionnaire	Alimentation,	Plus de 300 C	Universitaire	8	dépisté posit	Suspicion per	0	0	Zinc et vitam	Aucune mala	
11	Masculin	Marié	34	Aizo	Catholique	Agent de séc	Sécurité privé	Entre 40 000	Secondaire s(	9	dépisté posit	Suspicion per	0	0	Chloroquine,	Aucune mala	
12	Masculin	Marié	73	Xwla	Catholique	Retraité	Sécurité publ	Plus de 300 C	Secondaire s(	10	dépisté posit	Cas confirmé	0	0	AUCUN	Hypertension	
13	Feminin	Marié	65	Xwla	Catholique	Revendeuse/	Agriculture/E	Entre 40 000	Primaire	11	dépisté posit	Cas confirmé	0	0	AUCUN	Hypertension	Non
14	Masculin	Marié	34	Fon	Catholique	Fonctionnaire	Economie/Fir	Plus de 300 C	Universitaire	12	dépisté posit	Cas confirmé	0	10	RENFORCEM	Problèmes re	
15	Masculin	Concubinage	36	Mahi	Céleste	Fonctionnaire	Agriculture/E	Entre 100 00(	Universitaire	13	dépisté posit	Dépistage no	0	0	Chloroquine,	Aucune mala	
16	Masculin	Concubinage	28	Fon	Evangélique	Travailleur in	Santé	Entre 40 000	Universitaire	14	dépisté néga	Personnel de	0	0	Vitamine C ,	Aucune mala	
17	Masculin	Célibataire	23	TOFFIN	Evangélique	STAGIAIRE	Santé	Moins de 40	Universitaire	15	dépisté posit	SUR DEMANI	0	0	AUCUN	Aucune mala	
18	Feminin	Marié	36	Yoruba	Musulmane	Fonctionnaire	Santé	Entre 40 000	Secondaire 1(	16	dépisté néga	Dépistage vo	0	1	Aucun	Aucune mala	Non
19	Feminin	Célibataire	20	Goun	Catholique	Apprenant en	Educatif	Moins de 40	Universitaire	17	dépisté néga	Dépistage vo	0	0	AUCUN	Aucune mala	Non
20	Feminin	Concubinage	43	Xwla	Catholique	Coiffeuse	Artisanat	Moins de 40	Non scolarisé	18	dépisté néga	Suspicion per	0	0	Vitamines c	Aucune mala	Non
21	Masculin	Marié	31	Fon	Religion indi	Fonctionnaire	Economie/Fir	Plus de 300 C	Universitaire	19	dépisté posit	Cas confirmé	0	3	Calceïdra, art	Aucune mala	
22	Masculin	Célibataire	25	Fon	Catholique	En recherche	Santé	Moins de 40	Secondaire s(	20	dépisté néga	Exiger dans l	0	0	888	Aucune mala	
23	Masculin	Concubinage	32	Nago	Evangélique	Fonctionnaire	Educatif	Entre 100 00(	Universitaire	21	dépisté néga	Suspicion per	0	0	Chloroquine	Rhumatismes	
24	Masculin	Marié	45	KOTOKOLI	Musulmane	Fonctionnaire	Agriculture/E	Plus de 300 C	Universitaire	22	dépisté néga	Cas confirmé	0	1	INFUSION DE	Aucune mala	
25	Feminin	Concubinage	29	Fon	Catholique	Travailleur in	Educatif	Entre 40 000	Universitaire	23	dépisté néga	Aucune raiso	0	0	Aucun	Aucune mala	Non
26	Feminin	Voeuf ou divc	48	Fon	Evangélique	Revendeuse/	Agriculture/E	Entre 40 000	Primaire	24	dépisté posit	Dépistage no	0	0	Chloroquine	Hypertension	Non
27	Feminin	Célibataire	25	Fon	Catholique	Secrétaire	Educatif	Entre 100 00(	Secondaire s(	25	dépisté néga	Aucune raiso	0	0	Chloroquine	Maux de vent	Non
28	Masculin	Célibataire	20	Mahi	Eckiste	Apprenant en	Santé	Moins de 40	Universitaire	26	dépisté posit	Cas confirmé	2	0	Chloroquine,	Surpoids ou c	
29	Masculin	Célibataire	26	Nago	Evangélique	Fonctionnaire	Santé	Plus de 300 C	Universitaire	27	dépisté néga	Cas confirmé	0	0	Rien	Problèmes re	
30	Masculin	Célibataire	20	Fon	Evangélique	Etudiant	Etudiant	Moins de 40	Secondaire s(	28	dépisté néga	Aucune raiso	0	0	Chloroquine	Aucune mala	
31	Masculin	Concubinage	27	Fon	Evangélique	Fonctionnaire	Sécurité publ	Entre 40 000	Secondaire 1(	29	dépisté posit	Suspicion per	0	0	0	Aucune mala	
32	Feminin	Marié	54	Yoruba	Musulmane	Ménagère	Sans emploi	Entre 40 000	Primaire	30	dépisté néga	Dépistage vo	0	0	Chloroquine	Hypertension	Non

Fig. 1.4 – *Données après l'ETL*

1.3.4 Conclusion

En somme, il nous fallait réaliser un processus d'ETL permettant de récupérer les données au niveau des différentes sources internes du projet, les traiter et les charger dans notre base de données fédératrice. Nous avons présenté un certains nombre d'outils facilitant la réalisation du processus d'ETL. Ce pendant, pour notre projet nous implémentons notre propre outils d'ETL à partir des librairies Python. Celles-ci nous facilitent le chargement et le traitement de données depuis des fichiers. Néanmoins, l'ETL que nous avons implémenté ne traite que les données provenant des axes de recherche du projet. Une partie des données analysées dans ce projet provient du Web. Dans le chapitre suivant, nous présenterons les techniques utilisées pour collecter et intégrer les données externes (provenant du Web).

Chapitre 2

Collecte des données sur le WEB

2.1 Introduction

Depuis l'apparition du virus **SARS-CoV-2** en Chine au cours du mois de décembre 2019, plusieurs études ont été menées et une quantité non négligeable d'informations a été publiée sur le WEB. Malgré le fait que les différents axes de recherche du projet généreront une grosse quantité de données sur la pandémie de la **COVID-19**, il s'avère nécessaire d'enrichir ces données avec celles des précédentes études menées sur la pandémie et présentes sur le WEB. Pour y arriver, nous avons utilisé la technique du **WEB Scraping**.

Dans ce chapitre, nous abordons la solution que nous avons mise en place pour enrichir les données récoltées par le projet. Nous présentons d'abord une liste non exhaustive des différents outils de **WEB Scraping** et détaillons par la suite notre module d'enrichissement des données du projet.

2.2 Le WEB Scraping

Le **WEB Scraping** désigne l'ensemble des techniques d'extraction de contenu de pages web. C'est une activité de récupération d'informations sur divers sites web [28]. Elle permet ainsi de collecter et d'enregistrer des informations de différentes natures afin de les analyser ou de les utiliser à des fin personnelles.

D'après [22], le **WEB Scraping** trouve son application dans la recherche de certaines informations, dans l'indexation de pages Web ou l'analyse et le suivi de données. Il trouve également son application dans la lutte concurrentielle pratiquée par les entreprises.

Le **WEB Scraping** peut être fait avec ou sans la permission des propriétaires d'un site. Son processus peut être automatisé ou effectué manuellement [28]. On parlera de *scraping manuel* dans le cas où on fait des copies et insertions manuelles des données dans nos fichiers de stockage. Cette technique devient très laborieuse lorsqu'on souhaite l'appliquer sur une grande quantité de données. Les auteurs de [12] présentent deux raisons qui peuvent nous amener à automatiser nos collectes de données sur le Web et à maximiser l'absence d'apport humain dans la conception denotre méthode de collecte de données.

Premièrement, l'apport humain est chronophage et source d'erreurs. Une seule page Web utilisée pour l'entraînement peut potentiellement contenir un grand nombre de valeurs de données intéressantes et le l'entraîneur humain doit identifier chacune d'elles. En plus, l'entraînement pourrait nécessiter de nombreuses pages annotées par des humains.

Deuxièmement, beaucoup de collections de pages sont semis-structurées et contiennent des attributs optionnels. Si un attribut optionnel apparaît très rarement, il est possible pour l'entraîneur humain de le manquer complètement.

Tous les deux problèmes ci-dessus sont encore aggravés par le fait que, dans la pratique les modèles changent très fréquemment, ce qui nécessite des interventions répétées des humains.

Le scraping automatisé consiste à concevoir des scripts ou des programmes généralement appelés **robots**, qui se chargeront de récupérer toutes les informations depuis une page spécifiée en entrée. Mieux, certains robots ne se limitent pas à une seule page. Ils peuvent, après avoir récupéré les informations sur une page donnée, passer à une autre pour répéter le même processus : on parle dans ce cas de **Web crawling**. Les auteurs de [26] présentent l'algorithme de web crawling comme suit : étant donné un ensemble de localisateurs de ressources uniformes (URL) de départ, un robot télécharge toutes les pages adressées par les URL, extrait les hyperliens contenus dans les pages et télécharge de manière itérative les pages Web adressées par ces hyperliens.

De façon plus détaillé et d'après [23], La technique de **Web crawling** consiste à :

1. Sélectionner une URL à explorer
2. Récupérer et analyser la page cible
3. Enregistrer son contenu important avant d'extraire les URLs de la page et de les ajouter dans une file d'attente
4. Répéter toutes ces étapes pour chacun des URLs de la file d'attente

La majorité des outils de collecte de données sur le web offrent les deux approches (scraping et crawling). Dans la suite, nous présentons une liste non exhaustive d'outils de web scraping.

2.3 Les Outils de WEB Scraping

Depuis la naissance du Web scraping, plusieurs outils ont été développés pour faciliter l'automatisation de la collecte de données. Certains de ces outils sont payants et d'autres sont gratuits et accessibles au grand public. Dans cette section, nous présentons les trois outils open sources les plus utilisés que sont **Scrapy**, **BeautifulSoup** et **Selenium**. Ils ont été développés en langage Python.

2.3.1 Scrapy

Scrapy est un framework d'exploration et de scraping Web rapide de haut niveau, utilisé pour explorer des sites Web et extraire des données structurées de leurs pages. Il peut être utilisé pour un large éventail d'objectifs, de l'exploration de données à la surveillance et aux tests automatisés [4] . Son architecture est construite autour des *spiders*, qui sont des robots autonomes qui reçoivent un ensemble d'instructions. Suivant l'esprit des autres frameworks, tels que *Django*, il est plus facile de construire et de mettre à l'échelle de grands projets d'exploration en permettant aux développeurs de réutiliser leur code. Scrapy fournit également un shell d'exploration du Web, qui peut être utilisé par les développeurs pour tester leurs hypothèses sur le comportement d'un site [10]. Ce framework vient avec un bon nombre d'outils facilitant le Web scraping et le Web crawling. Il présente des performances remarquables et il figure parmi les bibliothèques les plus puissantes. L'un des plus gros avantages de ce framework est qu'il est construit sur **Twisted** qui est une bibliothèque de réseau asynchrone. Scrapy est implémenté en utilisant un code non bloquant pour la concurrence, rendant ainsi les performances de ses robots très remarquables. Il est important de préciser que le framework scrapy ne supporte pas la version 3 de python.

2.3.2 BeautifulSoup

BeautifulSoup comme son nom l'indique, est une bibliothèque très intéressante pour le Web scraping, au vu de ses fonctionnalités de base. Il est très facile d'extraire les données des

pages HTMLs à partir de cette bibliothèque. L'un de ses avantages est qu'elle est très facile à prendre en main pour un débutant. Elle a une très bonne documentation, nous permettant ainsi d'apprendre très rapidement. Mais il est nécessaire de noter qu'il a besoin de certaines bibliothèques telles que **Request** pour faire des requêtes vers le serveur, et un analyseur externe pour analyser les données téléchargées des pages HTML et XML.

2.3.3 Selenium

Selenium contrairement aux autres bibliothèques précédentes, est une bibliothèque créée initialement pour automatiser les tests pour les applications Web. Les développeurs s'en servent pour écrire des tests dans les langages de programmation populaires tels que **Java**, **Python**, **Ruby**, **C#**, etc. Il s'agit d'un framework créé pour l'automatisation du navigateur. Cependant, il a depuis été intégré au grattage Web. Il peut ainsi effectuer des requêtes Web et possède un analyseur syntaxique. On peut dès lors, à partir de cette bibliothèque, extraire du contenu d'un document HTML. L'avantage de cette bibliothèque est qu'elle est capable de charger du **Javascript**, ce qui permet d'accéder aux données derrière un code **Javascript**, sans avoir à faire des requêtes supplémentaires. De ce fait, certains utilisateurs des autres outils de Web Scraping font parfois recours à **Selenium** pour accéder aux données ne pouvant être disponibles que après un chargement de fichiers **Javascript**.

Dans le cadre de la réalisation du module d'enrichissement des données du projet **Also-Covid 19**, nous utilisons l'outil de Web scraping **Scrapy**. Dans la partie qui va suivre, nous présenterons notre solution, en commençant par vous donner les raisons du choix de l'outil **Scrapy**.

2.4 Presentation du module d'enrichissement des données du projet

L'objectif de ce module est d'enrichir les données du projet, par celles existantes sur le net, résultantes des précédentes études effectuées sur la Covid 19. Pour ce faire, nous avons de petits robots qui se chargent pour nous, de récupérer sur le Web les données que nous leurs spécifions et de les charger dans notre base de données.

2.4.1 Pourquoi choisir Scrapy ?

Plusieurs raisons nous ont poussé à faire le choix de l'outil **Scrapy** pour l'implémentation de nos robots au détriment des autres outils. Déjà, comme nous l'avons vu dans la section précédente, Scrapy est une bibliothèque complète pour le téléchargement, le traitement et l'enregistrement des données des pages Web. Elle nous propose plusieurs classes de Spider que nous pouvons faire hériter à nos robots, en fonction du besoin. C'est l'exemple de la classe **Crawler**, qui gère automatiquement la récupération d'URL sur une page et le passage d'une URL à une autre. En faisant hériter cette classe à notre robot, tout ce dont nous avons à faire est de définir les données à récupérer sur les pages. Notre robot gérera automatiquement la récupération et le suivi d' URL.
L'architecture de Scrapy est bien conçue pour nous permettre de personnaliser nos robots, en ajoutant nos propres fonctionnalités, les rendant ainsi plus robustes et plus flexibles. La bibliothèque Scrapy peut parcourir une liste d'URL en un temps très réduit et très facilement, du fait qu'elle utilise le framework **Twister**, qui fonctionne de manière asynchrone. D'après [30], Il y a principalement sept composants dans Scrapy parmis lesquels on peut citer :

1. **Scrapy Engine** : il est responsable du contrôle du flux de données entre tous les composants du système.
2. **Scheduler** : il reçoit les demandes du moteur et les met en file d'attente pour les alimenter plus tard lorsque le moteur en fait la demande.

3. **Downloader** : il est chargé de récupérer des pages Web et les envoyer au moteur qui, à son tour les fournit aux *spiders*.
4. **Les spiders** : ce sont des classes personnalisées que nous avons mises en place pour analyser les réponses et en extraire des éléments ou des URL supplémentaires à suivre.
5. **Item Pipeline**: il est responsable du traitement des articles une fois qu'ils ont été extraits par les spiders.
6. **Downloader middlewares** : Ce sont des crochets spécifiques qui se trouvent entre le **Scrapy Engine** et le **Downloader**. Ils fournissent un mécanisme pratique pour étendre les fonctionnalités de Scrapy en branchant du code personnalisé

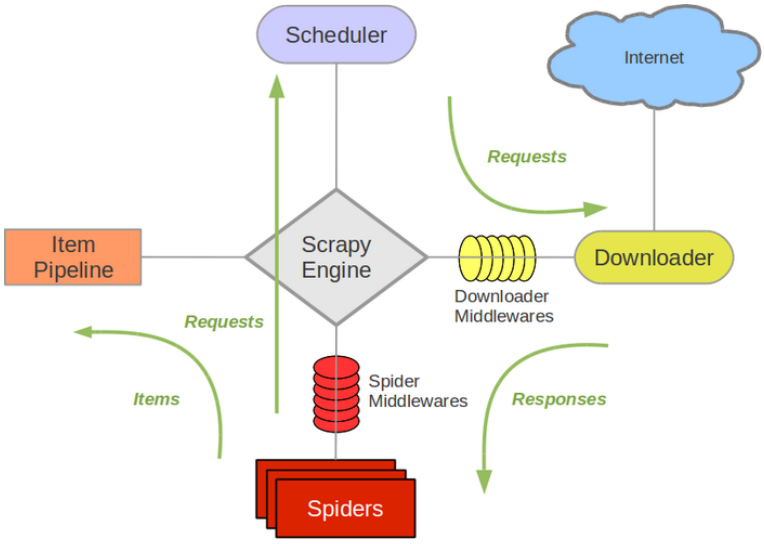


FIG. 2.1 – *Architecture de Scrapy [30]*

2.4.2 Présentation de notre solution

Dans le cadre de l'implémentation de ce module, nous avons réalisé au total sept robots de collecte de données. Nous pouvons les organiser en trois grandes catégories, en fonction de leurs objectifs et du type de données qu'ils récupèrent sur le Web.

OMS Spider

OMS Spider est un robot qui est chargé de récupérer les informations relatives à la pandémie de la Covid 19, sur le site de *l'Organisation Mondiale pour la Santée* (**OMS**). Il fait du *Web Crawling* car sur chaque page qu'il scrape des données, il récupère également les liens présents et charge ceux-ci dans la file d'attente.

En résumé, nous pouvons présenter le fonctionnement de ce robots comme suit : Initialement, il se positionne sur la page d'accueil du site de l'OMS. Vu le fait que les données du site ne sont pas totalement structurées, aussi qu'elles varient d'une page à une autre, nous récupérons tout le contenu textuel de la page. Par la suite, nous détectons en quelle langue est écrit le contenu de cette page, en faisant une simple analyse du texte avec des méthodes du *Traitement du Langage Naturel*. Un filtre est effectué à ce niveau, car nous nous intéressons qu'aux pages avec un contenu en français et en anglais. Pour les pages écrites en l'une de ces deux langues, toujours en nous servant des méthodes du *Traitement du Langage Naturel* [18], nous générons les mots clés et leur nombre d'occurrence sur chaque page. Ceci permettra par la suite de réussir à catégoriser nos différentes pages. Une fois ceci fait, nous récupérons les fichiers images, vidéos et documents présents sur la page. Toutes ces informations sont par la suite stockées dans notre base de données et le robot passe alors à la prochaine URL de la file d'attente, pour refaire ce même processus, ceci jusqu'à ce qu'il n'y ait plus d'URL non parcourue dans la file d'attente .

Rappelons que notre robot effectue la tâche de *Crawling* automatiquement. Nous n'avons pas eu à implémenter cette tâche, tout ce dont nous avons fait est de faire hériter à notre robot la classe *Crawler* de la librairie *Scrapy*. Le schéma suivant résume les différentes fonctionnalités du robot **OMS Spider**.

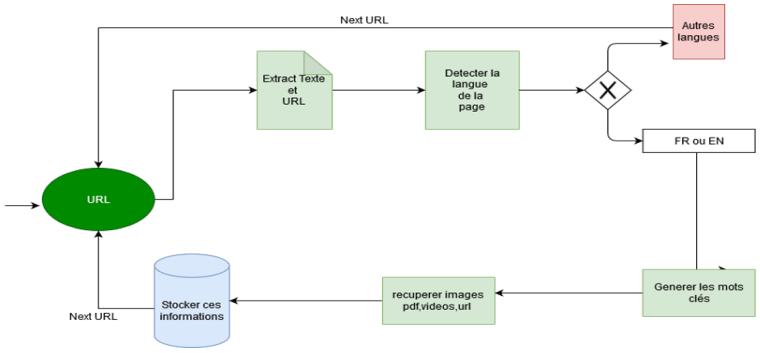


FIG. 2.2 – *Schémas de fonctionnement du Robot OMS Spider*

Article Spider

Article Spider est un groupe de 5 robots dont le rôle est de scraper les articles scientifiques. Chaque robot récupère les articles d'un site bien précis (tous font du *Web Crawling*). Les sites ciblés ici sont *PlosOne*, *Science Direct*, *Nature*, *PubMed* et *Elsevier*. Contrairement aux données du site de l'OMS, les données de ces sites sont complètement structurées. Ainsi, on définit au préalable les données que nous souhaitons récupérer sur les pages.

Le fonctionnement de ces robots peut être résumé comme suit : Au début, ils se positionnent sur la page d'accueil du site correspondant. nous parcourons cette page, récupérons pour chaque article ses métadonnées tel que le type d'article, le titre, la date de publication, les auteurs, l'URL de l'article et si possible le livre qui contient l'article. Par la suite, nous extrayons les mots clés de ces métadonnées à l'aide des techniques du *traitement de langage naturel*. Précisons que nous avons défini au préalable, une liste de mots propres à chaque Workpackage (axe de recherche du projet). Ainsi, après avoir ressorti les mots clés de l'article, nous les comparons à ceux de chaque workpackage, ceci dans le but de catégoriser chaque article en fonction des workpackages du projet. Ces informations sont alors stockées dans la base de données et les robots passent au prochain article s'il y en a encore sur la page, sinon passent à la prochaine page du site. Ils répètent ce processus jusqu'à la dernière page du site.

Notons que pour certains de ces sites, il existe une rubrique spécifique pour des articles concernant la pandémie de la Covid 19. Les robots qui récupèrent les informations sur ces sites se positionnent directement dans cette rubrique, car il s'agit d'articles qui nous intéressent particulièrement.

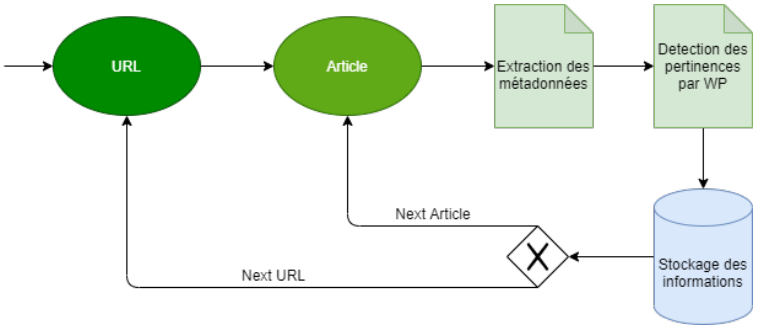


FIG. 2.3 – *Schémas de fonctionnement des Robots Article Spider*

InfosCovid Spider

InfosCovid Spider est un robot dont le rôle est de récupérer les indicateurs sur la pandémie de la Covid 19, tels que le nombre de nouveaux cas confirmés, le nombre de nouveaux décès, le nombre de nouveaux tests, le total de cas confirmés, le total de décès, etc.
https://ourworldindata.org/coronavirus-source-data est un site qui propose des données relatives à la pandémie de la Covid 19 pour presque tous les pays du monde. Ces données vont du 28 février de l'année 2020, jusqu'au jour précédent le jour courant et sont mis à jour quotidiennement. Les données sur ce site nous sont présentées sous trois formats : CSV, XSLS, JSON. Notre robot parcourt ainsi les données de ce site, pour extraire les données correspondantes à celles des pays de l'Afrique, membres de l'espace *CAMES*.

De façon plus détaillée, nous pouvons présenter le fonctionnement de ce robot comme suit : nous parcourons la page du site et récupérons l'URL du fichier de données au format CSV. Par la suite, nous téléchargeons le contenu de ce fichier csv, puis effectuons un premier filtre pour ne garder que les pays membres de l'espace CAMES. Après avoir récupéré les informations relatives à ces pays, nous effectuons un second filtre qui consiste à ne garder que les données dont la date est supérieure à la date la plus récente de notre fichier de données. Les données retenues sont alors stockées dans les fichiers *senegal.csv*, *gabon.csv*, *burkina.csv*, *mali.csv*, *benin.csv*, *autre.csv* .Chaque fichier contient les données relatives à son pays. Le fichier *autre.csv* contient les données des autres pays membres de l'espace CAMES.

Précisons que ce robot est programmé à s'exécuter tous les jours à partir de 4h du matin, afin d'actualiser les données. Ainsi, le filtre sur la date des données nous évite de dupliquer les données dans les fichiers données. Lors de la première exécution du robot, toutes les données des pays membres du CAMES sont chargées dans les fichiers. Pour les prochaines exécutions nous ne sauvegardons que les données ayant une date supérieure à la date la plus récente dans nos fichiers de stockage.

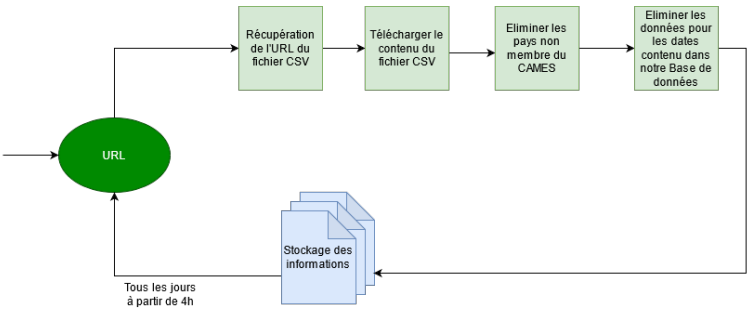


FIG. 2.4 – *Schémas de fonctionnement des Robots InfosCovid Spider*

2.5 Conclusion

En résumé, nous avons présenté dans ce chapitre la technique dite *Web Scraping*, que nous utilisons dans ce projet pour récupérer les données sur le Web. Cette technique nous a permis d'automatiser la collecte d'informations sur la pandémie de la covid-19. Nous avons fait le tour de quelques outils développés dans le langage de programmation Python. Pour des raisons de robustesse, de flexibilité, de facilité de prise en main, etc, nous avons fait le choix d'utiliser l'outil **Scrapy** pour l'implémentation de nos programmes de collecte. Nous avons dès lors conçu deux catégories de programmes. La première est chargée de récupérer les articles scientifiques. Nous avons réussi à récupérer au total quarante huit mille articles scientifiques, hors mis ceux de l'organisation mondiale de la santé. La seconde catégorie est chargée de récupérer chaque jour, les informations relatives à l'évolution de la pandémie de la covid-19 telles que le nombre de nouveaux cas confirmés, le nombre de nouveaux décès, le nombre de tests, le total de cas, le total de décès, etc, ceci pour tous les pays de l'espace CAMES. Ces informations vont du 28 février 2020 jusqu'à aujourd'hui. Ce chapitre clôture ainsi l'extraction et la sauvegarde des données du projet. Néanmoins, il serait aussi judicieux de trouver les meilleurs moyens de présenter ces données aux utilisateurs. Dans le chapitre qui va suivre, nous présenterons les outils de visualisation que nous avons conçus.

Chapitre 3

Exploitation et valorisation des données globales du projet

3.1 Introduction

Dans les chapitres précédents, nous avons réussi à créer notre base de données fédératrice, extraire les données des multiples sources du projet, les consolider et les sauvegarder dans la base de données. La réussite de cette sauvegarde représente une avancée considérable et un gros objectif atteint parmi tant d'autres. Néanmoins, il est important de proposer un meilleur moyen de visualisation et de restitution de ces données aux utilisateurs et membres du projet. Pour ce faire, nous avons développé une interface d'exploitation des données, pour le partage des données et la collaboration entre les workpackages du projet. Nous proposons également des tableaux de bord présentant un certain nombre d'analyses faites sur les données. Dans la suite de ce chapitre, nous n'entrerons pas en profondeur sur la conception de cette interface car elle fait l'objet d'un autre travail. Ainsi, nous montrerons juste un bref aperçu de l'interface de visualisation, par la suite nous verrons l'importance des tableaux de bord dans l'analyse des données et comment les concevoir. Nous présenterons également et de façon détaillée, nos solutions et un cas illustratif de valorisation de données que nous avons pu réaliser à partir des données du projet.

3.2 Présentation de l'interface d'exploitation des données

Dans le but de présenter les données et informations que nous avons pu recueillir tout au long de la réalisation de ce projet, nous avons conçu une interface d'exploitation de données. Cette interface est organisée en modules, présentant chacun des données bien ciblées. L'implémentation de certains de ces modules est déjà terminée et opérationnelle. D'autres sont partiellement opérationnels, avec des fonctionnalités en cours de traitement et donc pas disponibles. Il y en a d'autres modules qui sont entièrement indisponibles.

3.2.1 Module Home

C'est le module d'accueil. Au démarrage du site, on est redirigé vers ce module. Il présente les différents articles que nous avons récupérés sur le site de l'organisation Mondiale de la Santé. Pour chaque article, il nous donne son titre, une image représentative et une brève description de l'article comme nous pouvons le voir sur la figure 3.1 . L'utilisateur a la possibilité de cliquer sur le titre de l'article pour avoir plus d'informations sur cet article. Dans ce module, on donne aussi la possibilité d'effectuer des filtres sur les articles en fonction de l'année et de certaines maladies telles que la covid-19, la Malaria, la Tuberculose, etc. L'utilisateur peut également faire une recherche directe dans le contenu des articles en saisissant des mots dans une barre de recherche.

3.2.2 Module Info-Covid

Le but de ce module est de présenter l'évolution de la pandémie de la covid-19 et les tendances au niveau du continent africain. Il est divisé en deux rubriques: **Comparatif** et **Pays pilotes**.Nous y reviendrons sur la particularité de chacune des rubriques dans la suite lors de la présentation des solutions. Chaque rubrique présente un tableau de bord sur l'évolution de la covid-19 dans les pays africains membres de l'espace *CAMES*. Chaque tableau de bord est une application conçue, déployée et pouvant être accessible directement en renseignant l'URL lui correspondant. Nous reviendrons sur la conception de ces tableaux de bord dans la suite de ce chapitre.

3.2.3 Module Données

Ce module permet de visualiser les données de chaque workpackage. Il est organisé en quatre rubriques, chaque rubrique représentant un workpakage du projet. Au moment où nous écrivons

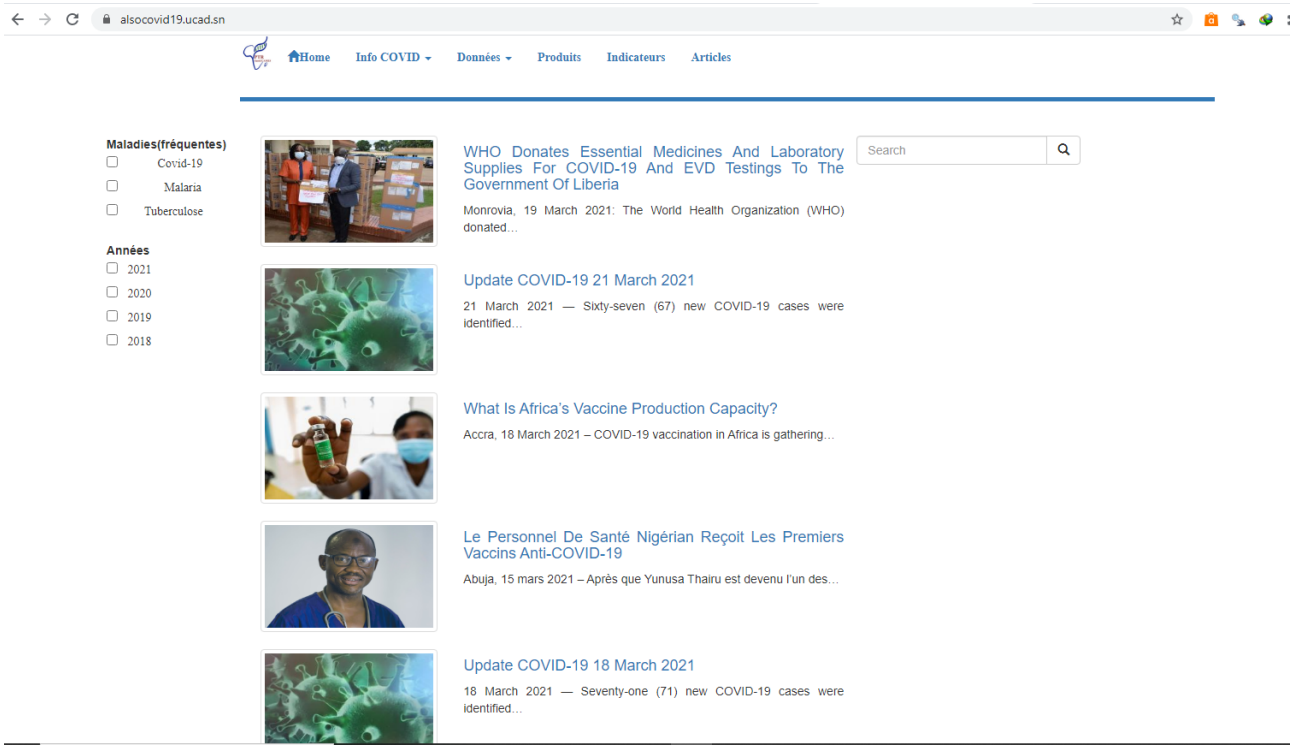


FIG. 3.1 – *schéma du module Home*

ces lignes, seule la rubrique du workpackage 2 est fonctionnelle car elle est la seule à avoir fourni ses données de recherche. Ainsi, dans cette rubrique, nous affichons sous forme d'un tableau, les données résultantes des recherches de ce workpackage, portant sur les déterminants socio-culturels, économiques, environnementaux et démographiques associés à la dynamique de la covid-19 en Afrique. L'utilisateur peut également télécharger ces données sous le format *json*, *csv* ou *xlsx*. A coté de ce tableau, nous présentons des images représentant un certains nombre de statistiques sur les données, comme on peut le voir à la figure 3.2 . Nous donnons aussi la possibilité aux utilisateurs de choisir les données dont ils souhaitent visualiser les statistiques. Ceci se fait grâce au tableau de bord dynamique que nous avons conçu et déployé comme les autres. Nous reviendrons sur ce tableau de bord dans la suite de ce chapitre. La figures 3.2 illustre la présentation de ce module.

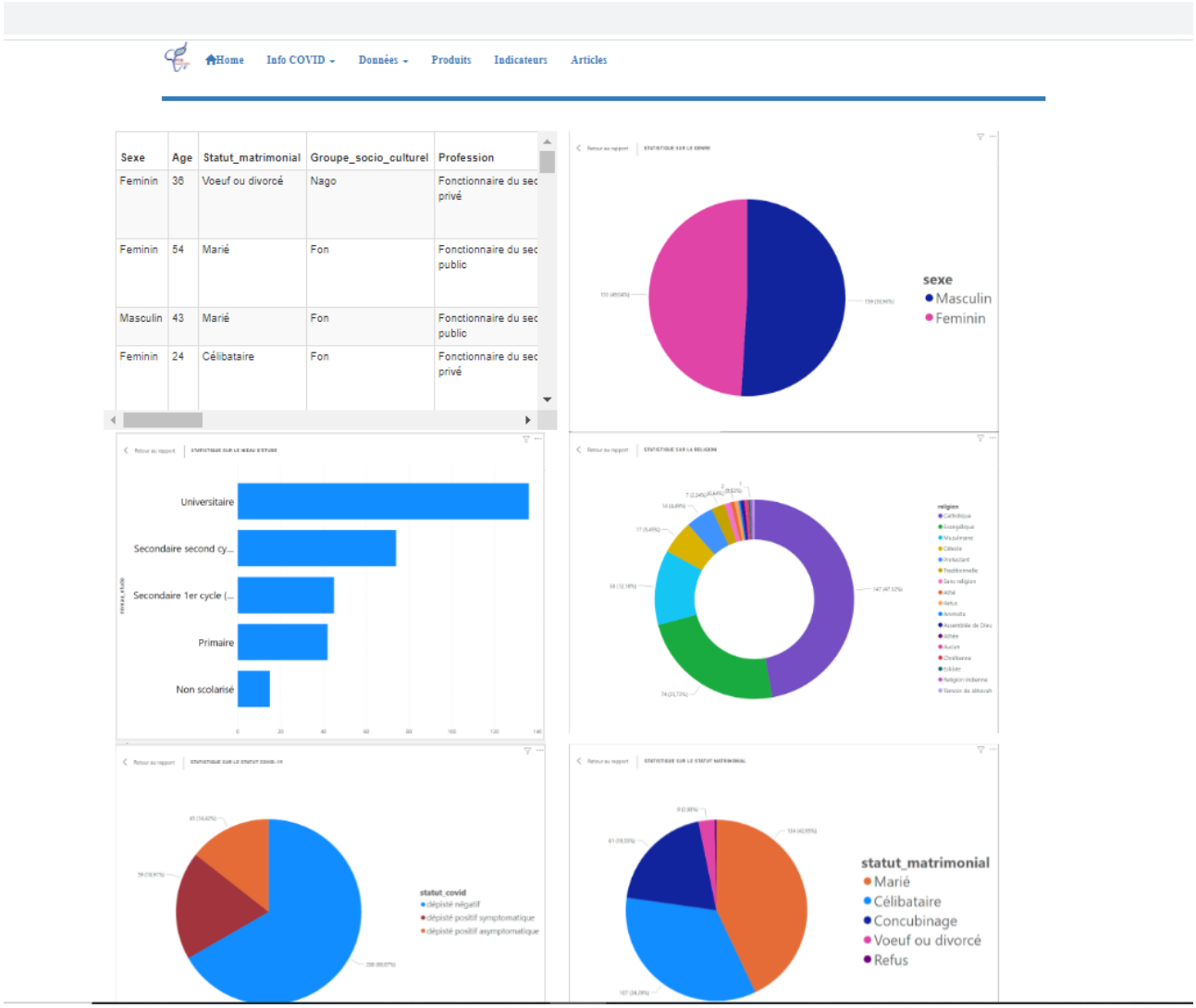


FIG. 3.2 – *schéma du module Données*

3.2.4 Module Produits et le Module Indicateurs

Ces modules ont pour rôle de présenter toutes les découvertes pertinentes et publications scientifiques faites tout au long de la réalisation de ce projet. Ces deux modules sont indisponibles pour l'instant car nous n'avons encore aucune découverte vraiment concrète, mais travaillons sur certaines idées. Néanmoins, nous comptons présenter dans le module Produits, les articles scientifiques que nous aurons à publier durant ce projet. Le module Indicateur contiendra les éléments tels que des modèles mathématiques caractérisants l'évolution de la covid-19 dans les pays de l'espace CAMES. Nous reviendrons sur ces modèles dans le cas illustratif présenté plus loin.

3.2.5 Module Article

Dans le chapitre précédent, nous avons réussi à récupérer des articles et revues scientifiques de certains sites de recherche. Dans ce module, nous présentons ces articles. Pour chaque article, nous affichons son type, qui peut être soit revue, journal, recherche, etc. Nous donnons le titre de l'article, la date de publication, les auteurs et le livre contenant l'article. L'utilisateur peut également cliquer sur le titre pour avoir plus de détails sur l'article. Il peut effectuer des filtres sur les articles tels que: l'année de publication, la source de l'article (le site de provenance de l'article) et le type d'article. Il a également la possibilité d'effectuer des recherches dans le contenu des articles, à partir de la barre de recherche. La figure 3.3 illustre la présentation de ce module.

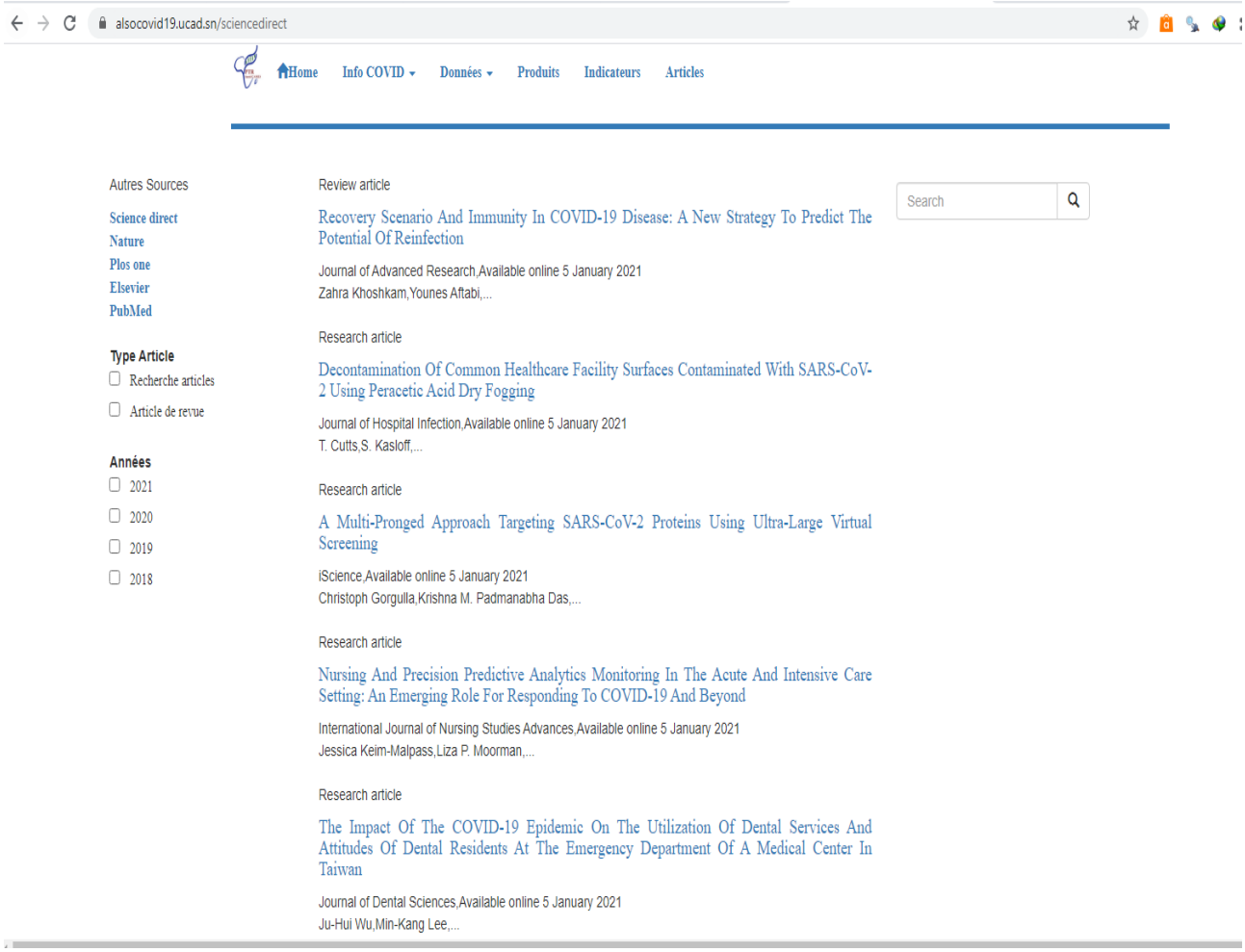


FIG. 3.3 – *schéma du module Articles*

3.3 Conception de nos outils d'analyses de données

Dans la section précédente, nous avons vu que certains modules tels que **Info-covid** et **Données** présentaient des tableaux de bords. En effet, les données brutes peuvent parfois être très difficiles à regarder. Toutes ces lignes et colonnes deviennent souvent impossibles à traiter et à comprendre. Les tableaux de bord transforment les données en informations, en créant plusieurs graphiques, tableaux et autres éléments visuels qui nous donnent une vue d'ensemble des données. Ils présentent un maximum d'informations dans un minimum de place, en faisant appel à une grande interactivité et multiples composants graphiques. Les tableaux de bords que nous avons implémentés contiennent des diagrammes présentant des statistiques sur les données du projet, des courbes d'évolution de la pandémie et des cartes géographiques regroupant des informations sur le covid-19, de chaque pays d'Afrique membres du CAMES.

3.3.1 Outils de conception de Tableaux de bord

Il existe de nombreux outils sur le marché permettant de réaliser les tableaux de bord et les reportings. Ces outils permettent l'automatisation des reportings, rendant ainsi très rapide la production de tableaux de bord clairs et directement opérationnels. En guise d'exemple, nous pouvons citer:

Microsoft Excel

Il est idéal pour débuter dans la création de tableaux de bord. La tâche devient plus difficile lorsqu'on souhaite ajouter l'interactivité dans nos tableaux de bord. Ceci nécessite l'utilisation de macros (**VBA**), le langage de programmation utilisé dans Excel. En combinant Excel et PowerPoint, le partage de nos tableaux de bord devient plus facile. Nous pouvons tous simplement l'enregistrer en tant que diaporama powerPoint et l'envoyer par courrier électronique à des collègues.

Pentaho

Nous avons présenté cet outil dans le chapitre d'extraction et transformation de données. Il offre également un reporting classique, à travers une plate-forme qui supervise les indicateurs de performance les plus importants. Il permet de créer des tableaux de bord avec une interface visuelle attrayante et intuitive. **Pentaho** nous aide à découvrir, analyser et visualiser les données pour trouver des réponses à nos besoins, même pour les utilisateurs sans expérience de programmation. Pour ceux avec une expérience avérée, l'outil vous propose un API vous permettant de personnaliser les rapports, les requêtes et les transformations afin d'étendre les fonctionnalités.

Tableau

Tableau est une plate-forme analytique visuelle qui transforme la manière d'utiliser les données pour répondre à des problématiques. Il s'agit d'une implémentation de visualisation de données actuellement très utilisée dans l'industrie de la Business Intelligence. On l'utilise pour transformer les données brutes en une forme compréhensible. Il ne nécessite aucune connaissance technique ou compétence en programmation. Il produit des visuels sous forme de consoles, tableaux, feuilles de calculs, etc.

Power BI

Power BI est une application d'analyse de données de Microsoft, permettant de se connecter à des données, de les transformer et de les visualiser. Elle offre une interface assez simple, facilitant la création de composants de visualisation de données personnalisés et interactifs. Elle fournit également des services de Business Intelligence hébergés sur le cloud, appelés *Services Power BI*, facilitant le partage de tableaux de bord avec d'autres utilisateurs.

Tous les outils que nous avons énumérés ci-dessus font partie des outils les plus utilisés sur le marché. Ceci à cause de leurs facilités de manipulation et créations de visuels interactifs. Cependant, pour des raisons que nous vous présenterons dans la suite, nous avons opté pour des librairies du langage *Python* pour l'implémentation de nos tableaux de bord.

3.3.2 Choix de l'outil d'implémentation

Pour l'implémentation de nos tableaux de bord, nous avons utilisé deux puissantes librairies du langage de programmation *Python*, qui sont **Plotly** [2] et **Dash** [1]. Le langage de programmation *Python* fait partie des langages les plus utilisées dans le domaine du traitement et d'analyse des données. D'après [27], l'utilisation de *Python* dans le domaine de la science des données a atteint des niveaux sans précédent, en particulier dans le domaine des outils et librairies disponibles gratuitement. *Python* met à notre disposition plusieurs bibliothèques pour la visualisation de données. On peut citer *Seaborn* [5], *Matplotlib* [17], *Plotly*, etc.

Plotly est une librairie de graphiques Python open source, idéale pour créer de belles visualisations interactives [8]. Elle semble être la plus puissante dans le domaine de visualisation de données [27]. Elle prend en charge la plupart des diagrammes standards utilisés en *Data Mining* et *Machine learning*. Plotly nous permet de créer des diagrammes interactifs, contrairement aux autres librairies. Grâce à son API *Plotly Express*, il est assez simple de générer des visuels avancés. En plus, nous pouvons intégrer les diagrammes générés depuis Plotly dans des pages web à partir de la librairie *Dash*.

Dash est une librairie Python open source également, permettant la création d'applications d'analyse web. Écrite au-dessus de **Flask**, **Plotly.js** et **React.js**, Dash est idéal pour créer des applications de visualisation de données avec des interfaces utilisateur hautement personnalisées

en Python. Elle permet de concevoir des tableaux de bord purement Python et facilement accessible au le grand public. Grâce à quelques modèles simples, Dash fait abstraction de toutes les technologies et protocoles nécessaires pour créer une application web interactive [1]. Les applications implémentées avec Dash peuvent être déployées sur des serveurs, puis partagées via des URLs. Par défaut, toutes les applications tournent sur le port 8050. Mais on a la possibilité de changer le port de l'application dans le code de création de celle-ci. Ainsi, il est important de noter que si nous avons plusieurs tableaux de bord ou applications Dash qui doivent tourner au même moment, chacun doit avoir un port différent des autres. Dans la suite, nous ferons un petit cas d'exemple de création d'une toute petite application avec nos deux librairie *Dash* et *Plotly*.

3.3.3 Exemple illustratif d'utilisation de Dash et Plotly

Nous allons ici créer une application qui affiche un tableau de bord dynamique permettant de visualiser des informations statistiques dans un mini jeu de données que nous avons créé et qui est présenté à la figure 3.4. Ce tableau de bord est dynamique dans le sens où l'utilisateur a la possibilité de choisir l'attribut qu'il souhaite visualiser, mieux encore, choisir également avec quel diagramme il souhaite visualiser cet attribut. Cet exemple a pour but de montrer comment il est facile et rapide de créer des applications Web de visualisation de données avec les librairies *Dash* et *Plotly*.

	A	B	C	D
1	**Sexe**	**Statut Matrimonial**	**Statut Covid-19**	
2	Homme	Marié	Positif	
3	Femme	Marié	Négatif	
4	Homme	Divorcé	Négatif	
5	Femme	Célibataire	Négatif	
6	Homme	Célibataire	Positif	
7	Femme	Divorcé	Négatif	
8	Femme	Marié	Positif	
9	Femme	Marié	Négatif	
10	Homme	Célibataire	Négatif	
11	Femme	Marié	Positif	
12				
13				
14				

FIG. 3.4 – *Jeu de données pour l'exemple*

Configuration de l'environnement

Pour créer notre application, nous utilisons l'IDE *Pycharm*. Nous avons choisi de travailler dans cette IDE pour la simple raison que chaque projet que nous créons dans cette IDE a son propre *environnement vituel*. La notion d'*environnement vituel* renvoie au fait que toutes les librairies que nous installerons dans ce projet ne seront installées que dans cet environnement et donc n'appartiendront qu'a ce projet.

Une fois que le projet est créé dans l'IDE *Pycharm*, nous installons les librairies *Dash*, *Plotly* et *Pandas* (celle-ci nous permet de nous connecter aux données). Ces installations se font en exécutant les instructions suivantes dans le terminale de l'IDE :

pip install plotly

pip install dash

pip install pandas

La création des applications *Dash* se fait géneralement en trois étapes:

1. **Connection à la source de données** :

   Dans cette étape, nous accédons aux données à visualiser, soit en se conectant à notre base de données, soit en accédant aux fichiers de données à partir de *Pandas* comme dans notre cas d'exemple. La figure 3.5 nous présente le code de connexion aux données, y compris les importations de packages nécessaires pour l'implémentation de notre application.

```
main.py
3   import dash_core_components as dcc
4   import dash_html_components as html
5   from dash.dependencies import Input, Output
6   import plotly.express as px
7
8   # This dataframe has 244 lines, but 4 distinct values for `day`
9   data = pd.read_excel ("E:/COUR/Master2BI/These/diagramme/dataSetTest.xlsx",
10                        engine='openpyxl')
11  valeurs_Att = [
12      ('Sexe','Sexe'),
13      ('Statut Matrimonial','Statut Matrimonial'),
14      ('Statut Covid-19',"Statut Covid-19")
15  ]
16
17  valeurs_gra = [
18      ('diagramme circulaire','pie-chart'),
19      ('diagramme en bande','bar-chart')
20  ]
```

FIG. 3.5 – *code de connection à la source de données*

2. **Création de l'interface de l'application** :

   La création de l'interface de notre aplication se fait à partir des deux composants **dash_core**

components et **dash_html_components** que nous avons eu à importer plus haut. Le package **dash_html_components** fournit un ensemble de composants HTML qui peuvent être rendus via python. Pas besoin d'avoir de fortes connaissances de base en HTML et CSS pour créer nos interfaces. On appelle juste des composants de ce package et il génère l'élément HTML correspondant dans notre application. Le package **dash_core_components** quand à lui, contient un ensemble de composants de haut niveau tels que des listes déroulantes, des graphiques, des blocs de démarque, etc. On crée ainsi notre interface en choisissant un ensemble de composants de ces packages. La figure 3.6 présente le code de création de l'interface de notre application.

main.py

```
]

# Initialisation de l'application

app = dash.Dash(__name__)

# Création de l'interface
app.layout = html.Div([
    html.H3("Attribut à visualiser:"),
    dcc.Dropdown(
        id='attribut',
        value='Sexe',
        options=[{'value': x, 'label': y}
                 for (x,y) in valeurs_Att],
        clearable=False
    ),
    html.H3("Type de graphe:"),
    dcc.Dropdown(
        id='diagram',
        value='pie-chart',
        options=[{'value': y, 'label': x}
                 for (x, y) in valeurs_gra],
        clearable=False
    ),
    dcc.Graph(id="graphe"),
])
```

FIG. 3.6 – *code de création de l'interface de l'application*

3. **Interactivité via des rappels :**

Dans cette étape, nous définissons comment doit se comporter l'application après la mise à jour d'un composant déclaratif. Dans notre cas d'exemple, nous avons deux composants déclaratifs qui sont des *check-list* de type **dcc.Dropdown**. L'enjeu ici est de définir le comportement de l'application après la modification de ces composants par l'utilisateur dans l'interface. Précisons que chaque composant que nous créons dans notre application a un identifiant que nous définissons de façon unique.

Pour donc définir notre interactivité, nous redéfinissons l'élément **@app.callback** de notre application. Dans cet élément, nous spécifions nos variables d'entrée et nos variables

de sortie. Les variables d'entrées sont les identifiants de nos composants déclaratifs (les *check-list* pour notre cas d'exemple). Les variables de sortie sont les identifiants des composants à mettre à jour (les diagrammes dans notre cas). Après avoir redéfini cet élément, nous définissons une fonction qui permettra de mettre à jour les différents composants de notre application. Cette fonction prend en paramètre toutes les variables d'entrée et retourne les variables de sortie. La figure 3.7 présente le code d'implémentation de l'interactivité dans notre application.

```
# Implémentation de l'interactivité

@app.callback(
    Output("graphe", "figure"),
    [Input("attribut", "value"),Input("diagram", "value")])
def generate_chart(attribut,diagram):

    if diagram == 'pie-chart':
        fig = px.pie(data, names=attribut)
    else:
        newData = data.groupby([attribut]).count()
        mydata = get_axis(newData)
        fig = px.bar(mydata, x='nom', y='valeur')
    return fig
```

FIG. 3.7 – *code de création de l'interactivité de l'application*

Après avoir implémenté ces trois étapes, nous pouvons déployer notre application sur les serveurs en ajoutant simplement la ligne d'instruction suivant dans notre code :

app.run_server()

Une fois que ceci est fait et que nous exécutons notre application dans notre IDE, nous pouvons visualiser notre application dans un navigateur en accédant à l'adresse **http://127.0.0.1:8050/**. L'application se présente comme sur les figures suivantes :

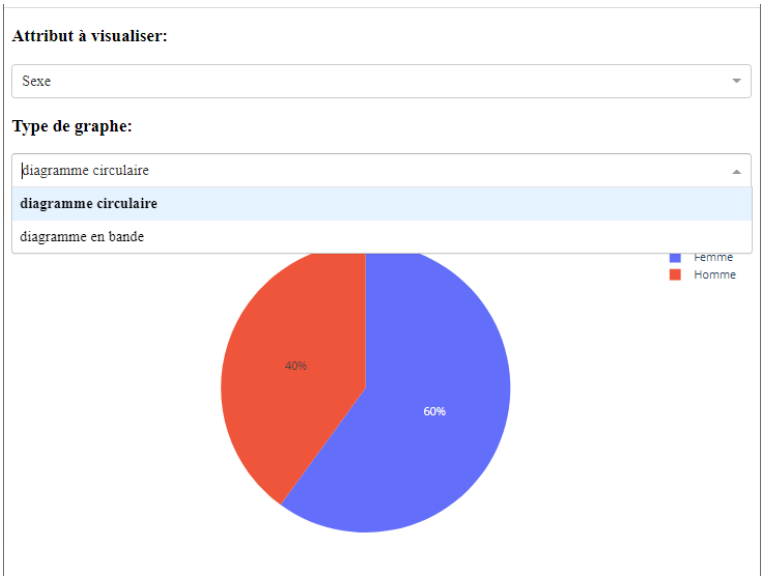


FIG. 3.8 – *schéma de l'application d'exemple avec l'attribut sexe et le diagramme circulaire*

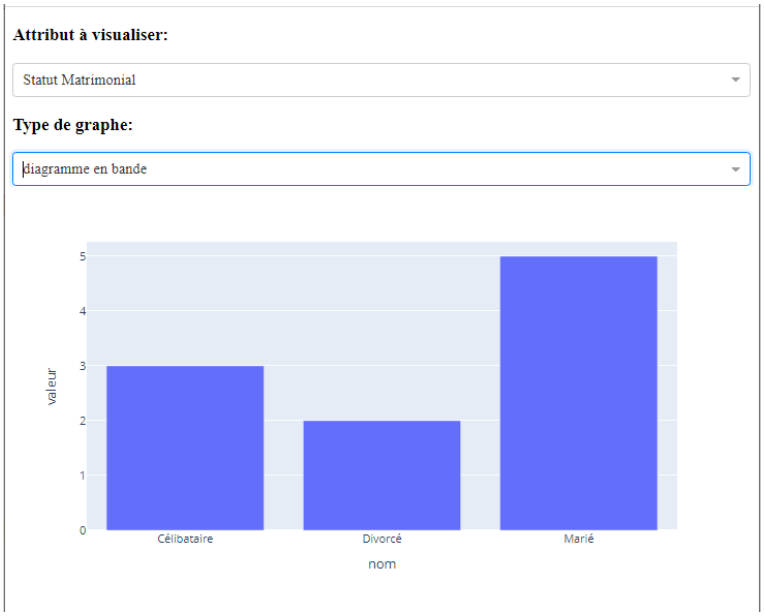


FIG. 3.9 – *schéma de l'application d'exemple avec l'attribut statut matrimonial et le diagramme en bande*

3.4 Présentation de nos solutions

Dans cette section, nous présentons les différentes solutions de visualisation de données que nous avons implémentées et un cas illustratif de valorisation de données du projet. Le but de cette illustration est de montrer comment on compte utiliser ces données sauvegardées pour l'aide à la prise de décision.

Pour la visualisation des données, nous avons conçu au total trois tableaux de bord dynamiques. Les deux premiers se trouvent dans le module *Info-covid*. Ils permettent d'analyser l'évolution

de la pandémie et les tendances au niveau du continent africain. Il s'agit du tableau de bord **Comparatif** et du tableau de bord **Pays Pilotes**. Le troisième tableau, situé dans le module *Données*, permet d'avoir une vue agrégée au niveau des données collectées dans le workpackage 2. Dans la suite, nous détaillerons le contenu de chaque tableau de bord.

3.4.1 Tableau de bord Pays Pilotes

Dans le chapitre précédent, nous avons présenté un robot qui collectait les données sur la covid-19 pour les pays membres de l'espace CAMES. Ces données renseignaient les informations sur le nombre de nouveaux cas, le nombre de nouveaux décès, le nombre de tests réalisés, le total de cas confirmés, le total de décès, etc. Le robot actualisait ces informations chaque jour. Ainsi, le tableau de bord **Pays Pilotes** présente ces informations sur la covid-19, pour les cinq pays réalisant le projet *ALSO-COVID-19* à savoir le Sénégal, le Mali, la Gabon, le Burkina et le Bénin. Vue le fait que chaque pays a ses informations propres pour cette pandémie, le tableau de bord possède une c*check-list* donnant la possibilité à l'utilisateur de choisir le pays dont il souhaite visualiser les informations, comme le montre la figure 3.10.
Le tableau nous montre la date de la dernière mise à jour des données. Nous avons également une carte géographique de l'Afrique, montrant tous les pays membres de l'espace CAMES. Au survol de la souris sur l'un de ces pays, le tableau nous présente les informations sur la covid-19 relatives à ce pays.
En plus de tout ceci, nous avons conçu pour chaque pays pilote, le graphe d'évolution de nouveaux cas (figure 3.11), le graphe d'évolution du total de cas par million d'habitants (figure 3.12) et le graphe d'évolution du total de décès par million d'habitants (figure 3.13). Dans l'optique de rendre tous ces graphes plus interactifs possibles, pour permettre une meilleur visualisation de l'information, nous avons prévu quatre niveaux d'agrégation des données dans ces graphes: l'agrégation des données par jour, par semaine, par mois et par années.
L'utilisateur a aussi la possibilité de paramétrer l'étendue de visibilité des données. Si il choisit le niveau d'agrégation jour, il peut également choisir sur combien de jours il veut voir les informations sur la covid-19. On peut accéder à ce tableau de bord en entrant l'URL *https://alsocovid19.ucad.sn:8743*.

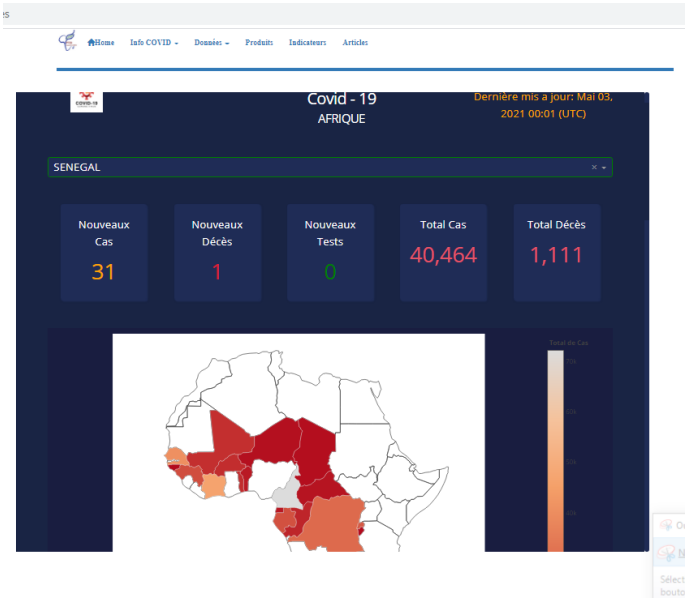


FIG. 3.10 – *schéma du Tableau de bord Pays Pilotes pour les informations sur le covid-19*

FIG. 3.11 – *schéma du Tableau de bord Pays Pilotes pour graphe d'évolution de nouveaux cas de covid-19*

FIG. 3.12 – *schéma du Tableau de bord Pays Pilotes pour graphe d'évolution du total de cas de covid-19*

FIG. 3.13 – *schéma du Tableau de bord Pays Pilotes pour graphe d'évolution du total de décès de covid-19*

3.4.2 Tableau de bord Comparatif

Ce tableau de bord traite les mêmes données que le précédent. Les deux ont en commun la carte d'Afrique qui présente les informations de la covid-19 sur les pays membres de l'espace CAMES. La différence entre le tableau *Pays Pilotes* et le tableau *Comparatif* réside à deux niveaux. D'abord, contrairement au tableau *Pays Pilotes* qui affichait les informations sur la covid-19 propre à chaque pays pilote du projet, le tableau *Comparatif* affiche le total de chacune des informations des cinq pays comme le présente la figrue 3.14.

Aussi, au niveau des graphes d'évolution de la pandémie, nous présentons les courbes propres à chaque pays pilote dans chaque graphe comme le montre les figures 3.15 ,3.16,3.17. Ceci nous permet d'avoir une idée comparative de l'évolution de la pandémie de la covid-19 dans les cinq pays pilotes du projet. Le tableau de bord *Comparatif* offre également le même mécanisme d'agrégation des données que le précédent. Il est accessible également directement en entrant l'URL *https://alsocovid19.ucad.sn:8081*.

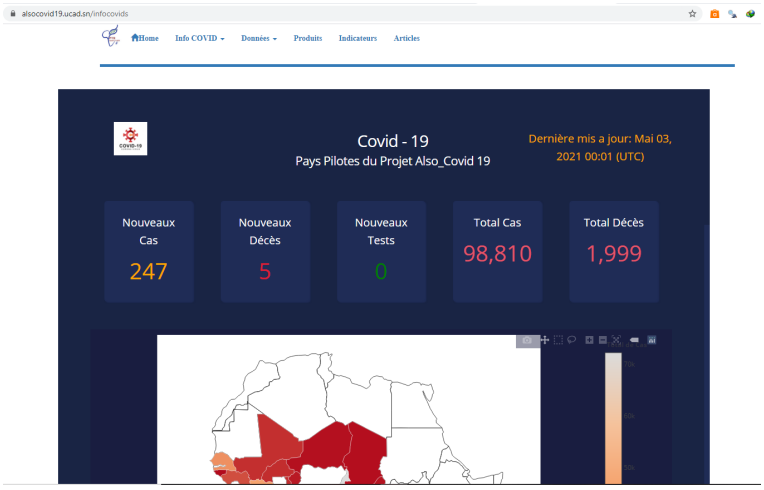


FIG. 3.14 – *schéma du Tableau de bord comparatif pour les informations sur le covid*

FIG. 3.15 – *schéma du Tableau de bord comparatif pour le graphe d'évolution de nouveaux cas*

FIG. 3.16 – *schéma du Tableau de bord comparatif pour le graphe d'évolution du total de cas*

FIG. 3.17 – *schéma du Tableau de bord comparatif pour le graphe d'évolution du total de décès*

3.4.3 Tableau de bord Données workpackage 2

Dans le deuxième chapitre, nous avons réussi à sauvegarder les données résultantes des recherches du workpackage 2. Ces données étaient relatives à l'étude des déterminants environnementaux, socio-culturels, économiques et démographiques associés à la dynamique de la covid-19 en Afrique, particulièrement au Bénin. Ce tableau de bord offre une visualisation dynamique de ces données. Dynamique dans le sens que l'utilisateur a la possibilité de faire le choix des informations qu'il souhaite visualiser. Pour chacune de ces informations, il lui est présenté une vue agrégée sous forme d'un diagramme circulaire et un diagramme à bande (figure 3.18). Le choix de l'information à visualiser se fait à partir d'une *check-list* qu'offre le tableau de bord.

FIG. 3.18 – *schéma du Tableau de bord Données workpackage 2*

3.4.4 Cas illustratif de valorisation de données

Nous avons vu plus haut que le *Tableau de bord Pays Pilotes* nous permettait de visualiser des graphes d'évolution de la pandémie, particulièrement celui de l'évolution du nombre de cas confirmés et du nombre total de cas de covid-19 pour les pays pilotes du projet. Sur ce même tableau, nous pouvons constater que le graphe du Gabon est depuis le 8 janvier 2021 en plein ascension. Les figures 3.19 et 3.20 montrent respectivement l'évolution du nombre de cas confirmés et du nombre total de cas de covid-19 au Gabon, sur les 359 jours compris entre le 14 mars 2020 et le 6 mars 2021 .

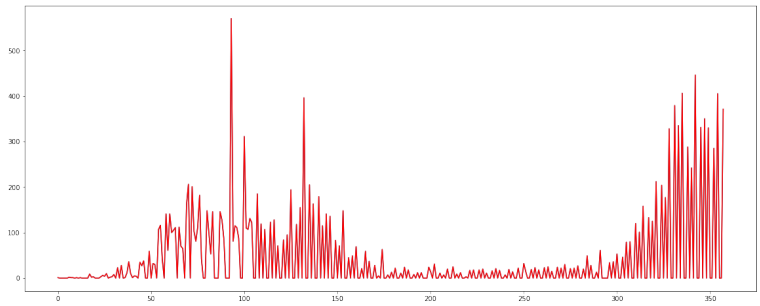


FIG. 3.19 – *schéma d'évolution du nombre de cas de covid-19 confirmés sur les 359 jours au Gabon.*

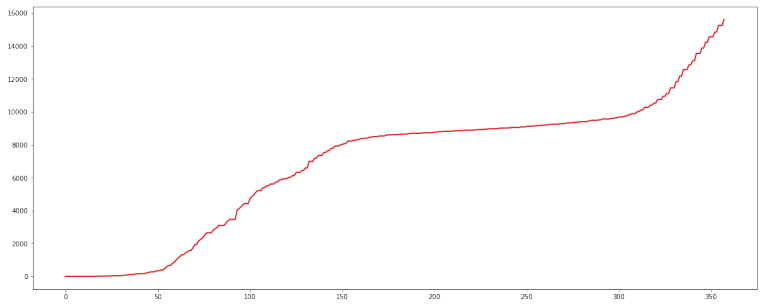


FIG. 3.20 – *schéma d'évolution du nombre total de cas de covid-19 sur les 359 jours au Gabon.*

Nous cherchons dans un premier temps à identifier les différentes périodes à forte pente sur ces graphes d'évolution. Sur les figure 3.19 et 3.20, nous pouvons identifier deux périodes pendant lesquelles les courbes croissent plus rapidement. Ce qui veut dire qu'il y a eu plus de contaminés dans ce pays pendant ces deux périodes. La première période va du 50ème jour qui représente le 3 mai 2020 au 150ème jour qui représente le 10 août 2020. La seconde va du 300ème jour qui représente le 8 janvier 2021 au 359ème jour qui est le 6 mars 2021. Les figures 3.21 et 3.23 présentent respectivement l'évolution du nombre de cas confirmés et du total de cas de covid-19 pour la première période, et les figures 3.22 et 3.24 présentent celles de la seconde période. Dans la prmière période, nous avons une moyenne de 77 nouveaux cas confirmés de covid-19 par jour,

avec un écart type de 92. Dans la seconde période, nous avons une moyenne de 120 nouveaux cas confirmés de covid-19 par jour, avec un écart type de 154.

Une fois que nous avons identifié ces différentes périodes, la suite consiste à analyser celles-ci en fonction de certains paramètres tels que **la température**, **le climat**, **les mesures barrières**, **la virulence du virus**, etc. En examinant nos deux périodes, nous avons constaté qu'en ces périodes, le climat était **froid**, les mesures barrières étaient **relâchées** et il n'y avait **pas de confinement**.

L'objectif de cette analyse est de trouver dans un premier temps les bon paramètres pouvant influencer l'évolution de la pandémie de la covid-19 au Gabon. Par la suite il s'agira de definir des modèles mathématiques capables de prédire le nombre de cas confirmés du covid-19 au Gabon pour une date future. Notons que si nous parvenons à définir un modèle pour les cas confirmés du Gabon, il ne sera pas compliqué de généraliser celui-ci pour en avoir un modèle mathématique pour la prédiction des cas des autres pays de l'espace CAMES.

Cette recherche est actuellement en cours de réalisation et nous espérons avoir des résultats satisfaisants dans un futur proche.

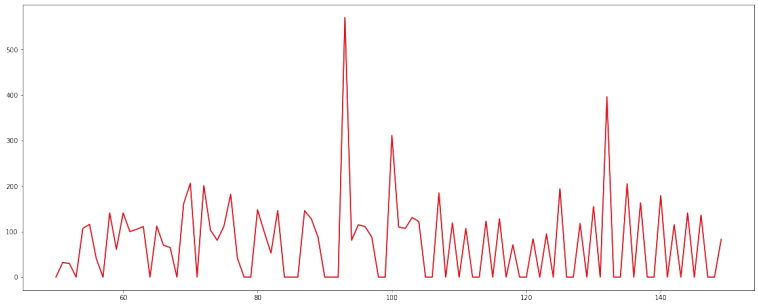


FIG. 3.21 – *schéma d'évolution du nombre de cas de covid-19 confirmés sur la première période.*

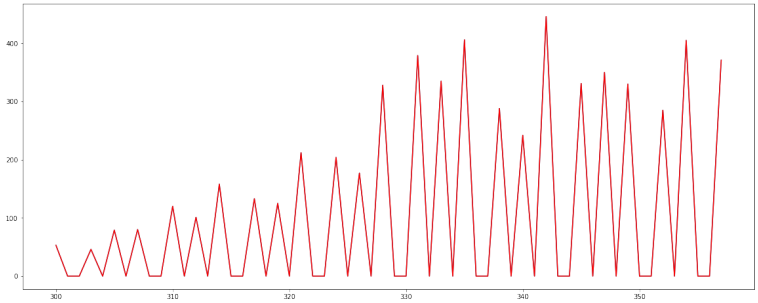


FIG. 3.22 – *schéma d'évolution du nombre de cas de covid-19 confirmés sur la seconde période.*

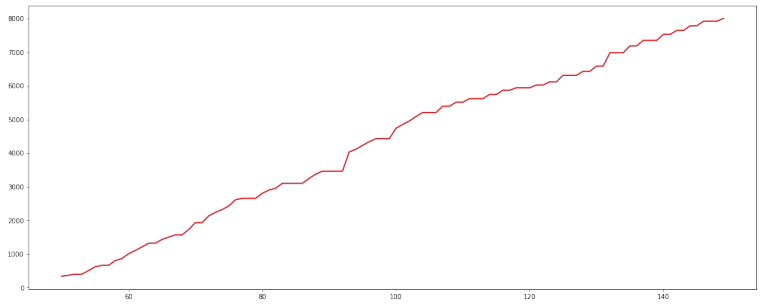


FIG. 3.23 – *schéma d'évolution du nombre total de cas de covid-19 sur la première période.*

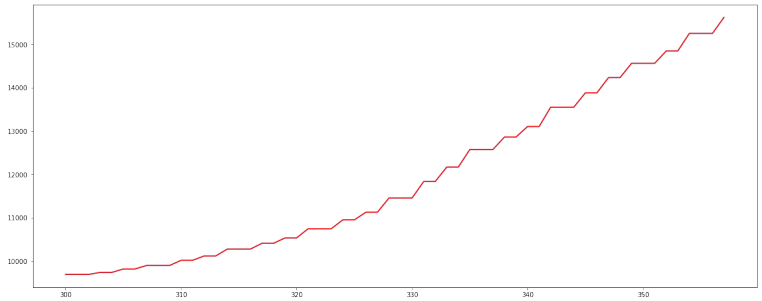


FIG. 3.24 – *schéma d'évolution du nombre total de cas de covid-19 sur la seconde période.*

3.5 Conclusion

Dans ce chapitre que nous venons de terminer, nous avons présenté les différents outils que nous avons implémentés pour visualiser les données du projet. Ces outils étant un site web et des tableaux de bord d'analyse de données. Nous nous sommes plus attardés sur la conception des tableaux de bord car cette étude est plus orientée vers l'analyse des données. Ainsi, nous avons dans un premier temps présenté certains outils existant sur le marché et facilitant l'automatisation de la création de tableaux de bord interactifs. Mais pour des raisons de travailler sur des solutions open source et de possibilité de partage sans restriction, nous avons utilisé les librairie *Plotly* et *Dash* pour réaliser nos tableaux de bord. Nous avons ainsi implémenté trois tableaux de bord. Le tableau *Pays Pilotes* qui présente les informations sur la pandémie, relatives à chaque pays pilote du projet. Le tableau *Comparatif* qui fait comme son nom l'indique, une présentation comparative des informations sur la covid-19 dans ces pays pilotes. Le tableau *Données Workpackage 2* ayant pour objectif d'offrir une visualisation dynamique et agrégée des données obtenues au niveau du workpackage2. Tous ces tableaux de bord sont accessibles depuis le site web et au niveau du navigateur, l'URL propre à chacun.

Conclusion générale et perspectives

Ce travail avait pour objectif de mettre en place un ensemble d'outils permettant l'exploitation et la valorisation de données issues des différents axes de recherches du projet **ALSO-COVID-19** et celles des études précédemment effectuées sur la pandémie de la covid-19. Pour ce faire, nous avons conçu une base de données fédératrice de type NoSQL pour la sauvegarde des données du projet. Par la suite, nous avons développé un outil de collecte quotidienne de données sur la pandémie, pour les pays membres de l'espace CAMES. Nous avons également développé un outil de collecte de publications scientifiques sur des sites d'informations sanitaires et des bases de données scientifiques. Nous avons également mis sur pied un outil d'analyse de l'évolution de la pandémie et des tendances au niveau du continent africain.
La présentation de notre travail dans ce présent rapport a été structurée en trois parties. Dans la première partie, nous avons décrit les différents moyens et technologies utilisés pour la conception de notre base de données et le chargement des données hétérogènes. Dans la seconde partie, nous avons présenté la technique dite *Web Scraping* utilisée dans ce projet pour enrichir les données collectées au niveau des différents workpackages. Cette technique nous a permis de collecter au total quarante huit mille publications scientifiques, hors mis ceux de l'organisation mondiale de la santé. Dans la dernière partie, nous avons présenté les différents moyens et outils que nous avons mis sur pied pour permettre l'exploitation et la valorisation des données du projet. Dans cette partie, nous nous sommes plus attardés sur la conception des tableaux de bords. A l'aide des librairies *Dash* et *Plotly*, nous avons conçu au total trois tableaux de bord. Le premier tableau de bord *Pays Pilote* fait un suivi individuel de chaque pays pilote du projet de l'évolution de la pandemie de la covid-19. Le second tableau *Comparatif* fait un suivi comparatif de la pandémie pour tous les pays pilotes. Enfin,le tableau de bord *Données Wp2* nous permet de visualiser de façon dynamique les données collectées au niveau du workpackage 2 du projet.

En guise de perspectives, nous comptons finir l'intégration des données des autres workpa-

ckages et concevoir également pour chacun, un tableau de bord de visualisation. Nous comptons également générer un modèle mathématique comme décrit dans le *Cas illustratif* du chapitre trois. Ce modèle doit pouvoir prédire le nombre de cas de personnes contaminées par le virus SARS-CoV-2 à une date donnée. L'intérêt de ce modèle de prédiction est de réussir à identifier les paramètres agissants sur l'évolution de la pandémie, afin de proposer une meilleure aide à la prise de décision .

Bibliographie

[1] Dash Documentation & User Guide | Plotly.

[2] Getting Started with Plotly.

[3] Que signifie ETL (et ELT)? - Definition IT de Whatis.fr.

[4] Scrapy 2.4 documentation — Scrapy 2.4.1 documentation.

[5] seaborn: statistical data visualization — seaborn 0.11.1 documentation.

[6] Talend vs SSIS | Top 8 Amazing Comparisons You Need To Know, Apr. 2018.

[7] Comment bien choisir une solution ETL pour un projet d'intégration ?, Oct. 2019.

[8] Plotly Python Tutorial for Machine Learning Specialists, Dec. 2020.

[9] Talend Data Integration vs Penthao Data Integration : qui choisir ?, Jan. 2020.

[10] Scrapy, Feb. 2021. Page Version ID: 1004776394.

[11] B. A. Ahidjo, M. W. C. Loe, Y. L. Ng, C. K. Mok, and J. J. H. Chu. Current Perspective of Antiviral Strategies against COVID-19. *ACS Infectious Diseases*, 6(7):1624–1634, July 2020. Publisher: American Chemical Society.

[12] A. Arasu and H. Garcia-Molina. Extracting structured data from Web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 337–348, New York, NY, USA, June 2003. Association for Computing Machinery.

[13] A. F. Attah, A. A. Fagbemi, O. Olubiyi, H. Dada-Adegbola, A. Oluwadotun, A. Elujoba, and C. P. Babalola. Therapeutic Potentials of Antiviral Plants Used in Traditional African Medicine With COVID-19 in Focus: A Nigerian Perspective. *Frontiers in Pharmacology*, 12, Apr. 2021.

[14] E. Baralis, A. Dalla Valle, P. Garza, C. Rossi, and F. Scullino. SQL versus NoSQL databases for geospatial applications. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3388–3397, Dec. 2017.

[15] X. Blanc and I. Mounier. *UML 2 pour les développeurs : Cours avec exercices corrigés*. Eyrolles, 2006.

[16] R. Bruchez. *Mettre en oeuvre une base de données noSQL : Comprendre et mettre en oeuvre Ed. 1*. Eyrolles, 2013.

[17] J. D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(03):90–95, May 2007. Publisher: IEEE Computer Society.

[18] A. Kao and S. R. Poteet. *Natural Language Processing and Text Mining*. Springer Science & Business Media, Mar. 2007. Google-Books-ID: CVtxFWbKT7wC.

[19] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Aug. 2011. Google-Books-ID: XoS2oy1IcB4C.

[20] K. Kline, D. Kline, and B. Hunt. *SQL in a Nutshell: A Desktop Quick Reference Guide*. O'Reilly Media, Inc., Nov. 2008. Google-Books-ID: F4dzMjYmLRIC.

[21] Y. Li and S. Manoharan. A performance comparison of SQL and NoSQL databases. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 15–19, Aug. 2013. ISSN: 2154-5952.

[22] M. Malairau, D. Redko, and S. Raș. Web Scraping. Comment etre proteges contre la collecte automatique de donnees. 2020. Accepted: 2020-06-03T12:05:31Z ISBN: 9789975456326 Publisher: Tehnica UTM.

[23] K. Matsudaira. Capturing and structuring data mined from the web. *Commun. ACM*, 57(3):10–11, Mar. 2014.

[24] S. A. Meo, D. C. Klonoff, and J. Akram. Efficacy of chloroquine and hydroxychloroquine in the treatment of COVID-19. *European Review for Medical and Pharmacological Sciences*, 24(8):4539–4547, Apr. 2020.

[25] R. Mukherjee and P. Kar. A Comparative Review of Data Warehousing ETL Tools with New Trends and Industry Insight. In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 943–948, Jan. 2017. ISSN: 2473-3571.

[26] C. Olston and M. Najork. *Web Crawling*. Foundations and trends in information retrieval. Now Publishers, 2010.

[27] I. Stančin and A. Jović. An overview and comparison of free Python libraries for data mining and big data analysis. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 977–982, May 2019. ISSN: 2623-8764.

[28] K. T., K. Sekaran, R. D., V. V, and B. Jeyakumar. Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques. *International Journal of Web Portals*, 11:41–52, July 2019.

[29] J. Trujillo and S. Luján-Mora. A UML Based Approach for Modeling ETL Processes in Data Warehouses. In I.-Y. Song, S. W. Liddle, T.-W. Ling, and P. Scheuermann, editors, *Conceptual Modeling - ER 2003*, Lecture Notes in Computer Science, pages 307–320, Berlin, Heidelberg, 2003. Springer.

[30] J. Wang and Y. Guo. Scrapy-based crawling and user-behavior characteristics analysis on taobao. In *2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 44–52, 2012.

[31] P. Warden. *Big Data Glossary*. O'Reilly Media, Sept. 2011. Google-Books-ID: vujmwAEACAAJ.