UNIVERSITÉ DE PARIS
Université Paris Diderot

RAPPORT DU PROJET DE " BASES DE DONNÉES AVANCÉES "

de 1^e année de Master en Informatique
parcours IMPAIRS + DATA
soutenu par

Djamel ALI
et
Jéremy DAMOUR

le 18 mai 2021

(Projet fait en 2^{nd}*semestre* de M1)

Base de données en lien avec les hospitalisations COVID-19

L'équipe enseignante

Cours
Mme. Cristina SIRANGELO

Travaux Dirigés et Travaux Pratiques
M. Sylvain SCHMITZ
M. Wieslaw ZIELONKA
M. Yan JURSKI

Année universitaire 2020-2021

Remerciements

Bonjour,

Avant tout développement et analyse de ce qui a été fait dans ce projet, il apparaît opportun de commencer ce rapport par des remerciements, à tous ceux qui nous ont beaucoup appris au cours de ce semestre d'étude.

Nous remercions particulièrement nos enseignants de ce cours : *Mme. Cristina SIRANGELO*, *M. Sylvain SCHMITZ*, *M. Wieslaw ZIELONKA* et *M. Yan JURSKI* qui nous ont formés et accompagnés tout au long du semestre avec beaucoup de patience et de pédagogie.

1. Introduction

Nous nous sommes concentrés sur les données hospitalières relatives à l'épidémie de COVID-19.

Ces données sont mises à jour tous les jours (vers 17 h) par le gouvernement français. On peut donc avoir un bon suivi de l'épidémie de COVID-19 dans les hôpitaux. Et voir ainsi l'impact du virus sur les sexes et différentes tranches d'âges par Département et Région (métropolitaine et outre-mer).

On s'est penché sur la mise en place d'une base de données, initialisable facilement avec des données locales, mais aussi la possibilité de la mettre à jour.

2. Source des données

➔ Données hospitalières relatives à l'épidémie de COVID-19
https://www.data.gouv.fr/fr/datasets/donnees-hospitalieres-relatives-a-lepidemie-de-covid-19/#_)

- **donnees-hospitalieres-nouveaux-covidXXX.csv**
- ***covid-hospit-incid-reg-XXX.csv (permet la vérification)***
- **donnees-hospitalieres-covid19XXX.csv**
- **donnees-hospitalieres-classe-age-covid19-XXX.csv**
- **donnees-hospitalieres-etablissements-covid19-XXX.csv**

➔ Pour la liste de tous les départements de France métropolitaine et DOM:
https://www.data.gouv.fr/en/datasets/departements-de-france/

- departements-france.csv

➔ Pour la liste de toutes les régions de France métropolitaine et DOM:

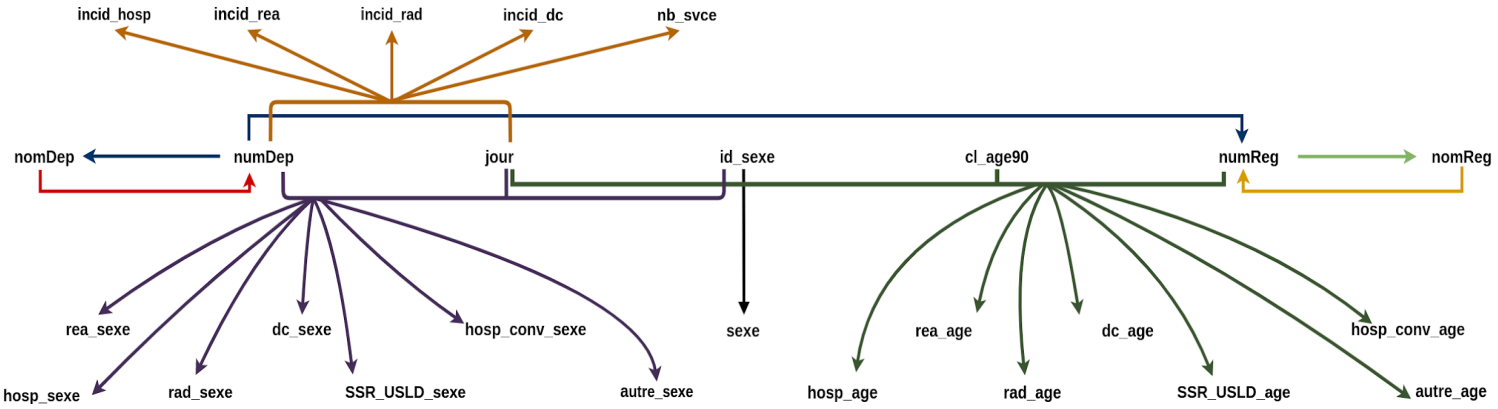
- regions-france.csv

(couleur utilisée plus bas)

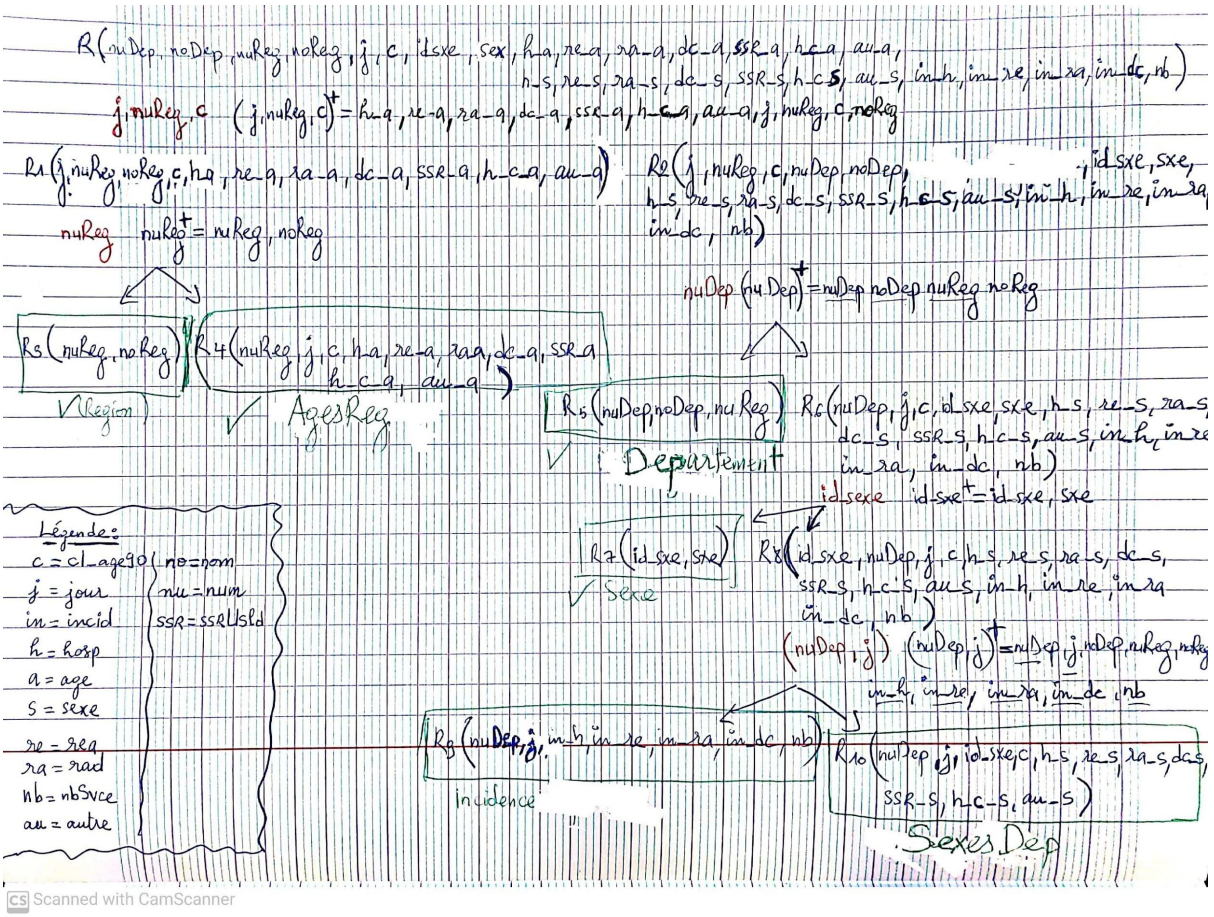
3. Dépendances fonctionnelles

F= {
numDep → nomDep, numReg
nomDep → numDep
numReg → nomReg
nomReg → numReg
idSexe → sexe
jour, numReg, clAge90 → hospAge, reaAge, radAge, dcAge, ssrUsldAge, hospConvAge, autreAge
jour, numDep, idSexe → hospSexe, reaSexe, radSexe, dcSexe, ssrUsldSexe, hospConvSexe, autreSexe
jour, numDep → incidHosp, incidRea, incidRad, incidDc, nbSvce
}

a. Diagramme résumant les DFs



4. Décomposition FNBC



5. Schéma relationnel

Region(**numReg**, nomReg)

Sexe(**idSexe**, sexe)

Departement(**numDep**, nomDep, numReg*)

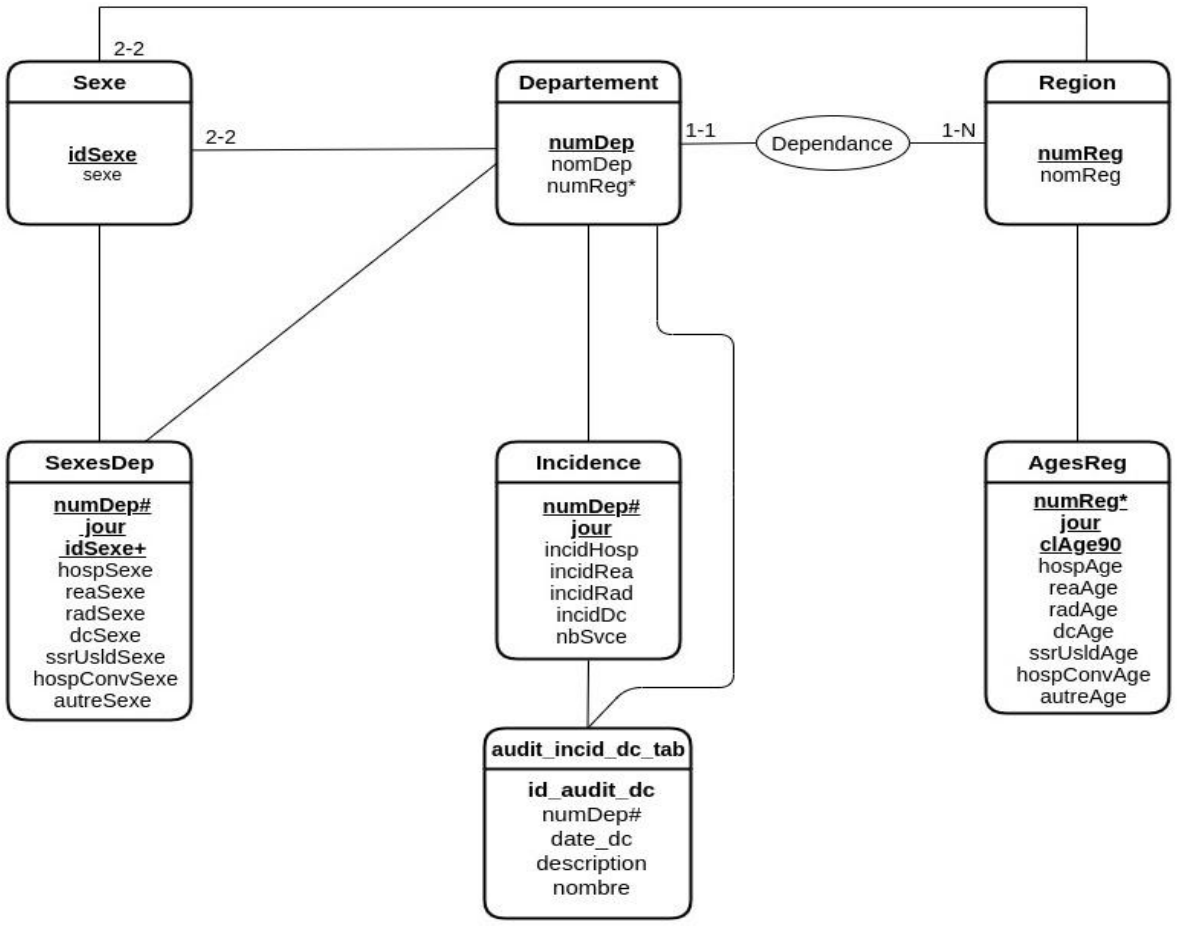
Incidence(**numDep#, jour**, incidHosp, incidRea, incidRad, incidDc, nbSvce)

AgesReg(**numReg*, jour, clAge90**, hospAge, reaAge, radAge, dcAge, ssrUsldAge, hospConvAge, autreAge)

SexesDep(**numDep#, jour, idSexe+**, hospSexe, reaSexe, radSexe, dcSexe, ssrUsldSexe, hospConvSexe, autreSexe)

audit_incid_dc_tab(**id_audit_dc**, numDep#, date_dc, description, nombre)

6. Schéma entité / relation



7. Contraintes

(voir les noms des fichiers sources des données en suivant la couleur de la colonne, bleu est manuel).

	Colonne	Type	Contrainte	Description
Region	numReg	UNSIGN INT	PRIMARY KEY	Numéro de la région
	nomReg	VARCHAR	NOT NULL	Nom de la région
Departement	numDep	UNSIGN INT	PRIMARY KEY	Numéro du département
	nomDep	VARCHAR	NOT NULL	Nom du département
	numReg	UNSIGN INT	FOREIGN KEY REFERENCE Region (numReg)	Numéro de la région
Sexe	idSexe	INTEGER	PRIMARY KEY CHECK IN (1, 2)	Sexe du patient (1: hommes, 2: femmes)
	sexe	VARCHAR	UNIQUE CHECK IN (“masculin”, “feminin”)	Sexe masculin ou féminin
Incidence	numDep	UNSIGN INT	FOREIGN KEY REFERENCES Departement (numDep)	Numéro du département
	jour	DATE	PRIMARY KEY	Date de notification
	incidHosp	UNSIGN INT	NOT NULL	Nombre quotidien de personnes nouvellement hospitalisées
	incidRea	UNSIGN INT	NOT NULL	Nombre quotidien de nouvelles admissions en réanimation
	incidRad	UNSIGN INT	NOT NULL	Nombre quotidien de nouveaux retours à domicile
	incidDc	UNSIGN INT	NOT NULL	Nombre quotidien de personnes nouvellement décédées
	nbSvce	UNSIGN INT	NOT NULL	Nombre cumulé de services hospitaliers ayant déclaré au moins un cas pendant la journée dans le département
AgesReg	numReg	INTEGER	FOREIGN KEY REFERENCES Region (numReg)	Numéro de la région
	jour	DATE	PRIMARY KEY	Date de notification

	clAge90	UNSIGN INT	NOT NULL	Classe d'âge de la personne (par intervalle de 10 ans / classe)
	hospAge	UNSIGN INT	NOT NULL	Nombre de personnes actuellement hospitalisées
	reaAge	UNSIGN INT	NOT NULL	Nombre de personnes actuellement en réanimation ou soins intensifs
	radAge	UNSIGN INT	NOT NULL	Nombre cumulé de personnes retournées à domicile
	dcAge	UNSIGN INT	NOT NULL	Nombre cumulé de personnes décédées
	ssrUsldAge	UNSIGN INT		Le nombre de personnes actuellement en Soins de Suite et de Réadaptation (SSR) ou Unités de Soins de Longue Durée(USLD)
	hospConvAge	UNSIGN INT		Le nombre de personnes actuellement en hospitalisation conventionnelle
	autreAge	UNSIGN INT		Le nombre actuellement de personnes hospitalisées dans un autre type de service
SexeDep	numDep	UNSIGN INT	FOREIGN KEY REFERENCES Departement (numDep)	Numéro du département
	jour	DATE	PRIMARY KEY	Date de notification
	idSexe	UNSIGN INT	FOREIGN KEY REFERENCES Sexe (id_sexe)	Sexe du patient (1: hommes, 2: femmes)
	hospSexe	UNSIGN INT	NOT NULL	Nombre de personnes actuellement hospitalisées
	reaSexe	UNSIGN INT	NOT NULL	Nombre de personnes actuellement en réanimation ou soins intensifs
	radSexe	UNSIGN INT	NOT NULL	Nombre cumulé de personnes retournées à domicile
	dcSexe	UNSIGN INT	NOT NULL	Nombre cumulé de personnes décédées
	ssrUsldSexe	UNSIGN INT		Le nombre de personnes actuellement en Soins de Suite et de Réadaptation (SSR) ou Unités de Soins de Longue Durée(USLD)
	hospConvSexe	UNSIGN INT		Le nombre de personnes actuellement en hospitalisation conventionnelle
	autreSexe	UNSIGN INT		Le nombre actuellement de personnes hospitalisées dans un autre type de service

Remarque :

Les attributs de la forme :

{hosp | rea | rad | dc | …}_{sexe | age} = Nombre cumulé de personnes {hosp | rea | rad | dc | …} **filtré par sexe ou par classes d'ages**.

8. Scripts

On a décidé d'utiliser des scripts *Python* pour peupler les tables, pour contourner ces différents problèmes lors du peuplement de la base :

- *COPY* est une opération nécessitant d'être le super-user de la base *postgresql*.
- *COPY* nécessite un chemin absolu du fichier CSV que l'on veut copier dans la table de destination
- Téléchargement et insertions des données facilement.

On a 2 scripts principaux écrits en python :

- *init.py* qui sert à initialiser la base et peuple la base avec les données existantes
- *download.py* sert à télécharger et insérer les nouvelles données dans la base.

Ces scripts python vont juste utiliser des templates dans le langage SQL et les compléter si nécessaire et donc créer la base de donnée adéquate. Ces templates sont des fichiers SQL qui se trouvent dans le dossier "src/templates".

Ces scripts servent donc juste à lancer les différents fichiers, ou encore à insérer les données brutes dans des tables temporaires. Elle ne fait donc aucunement du travail sur les données à insérer. Tout le travail de traitement pour l'insertion est fait par des Triggers, et des fonctions écrites en plpgsql.

9. Index

Des index ont été créés sur les tables principales du projet :

- SexeDep
- Incidence
- AgeReg

Ces index permettent notamment d'effectuer des joints plus rapidement entre ces différentes tables.

10. Triggers et Fonctions

On a des fonctions de *Trigger* servant à ne pas rajouter des lignes déjà existantes et renvoyant juste un “Raise Notice” informant l’utilisateur que l’insertion de sa ligne n’a pas été faite, car elle était déjà présente.

Pour insérer les données dans la table *SexesDep*. On doit retirer les lignes avec *idSexe* à 0 (qui est le cumul d’homme et femme). On va donc insérer toutes les données dans une table temporaire, et appeler une fonction qui va insérer que les lignes ayant ses sommes entre ses lignes homme et femme correctes dans la table définitive.

Pour insérer les données dans la Table *Incidence*. On a besoin de 3 fichiers :

- incidence dans le département
- incidence dans la région
- nombre de services touchés dans le département

On doit donc contrôler si l’incidence dans la région est bien la somme des incidences dans ses départements. Puis d’insérer la jointure avec les services touchés dans la table finale Incidence si le contrôle effectué est correct.

On a mis aussi en place une table *audit_incid_dc_tab* pour maintenir une trace de tous les changements qui pourraient affecter l'attribut *incidDc* de la table Incidence ; principalement pour avoir un meilleur suivi du nombre de décès dans chaque département.
les attributs la table *audit_incid_dc_tab* et leurs significations :

- *id_audit_dc* : identifiant (clé primaire) des entrées de cette table.
- *numDep* : numéro du département où ce *nombre* décès a eu lieu.
- *date_dc* : le jour (*date_dc*) où ce nombre de décès a été relevé dans ce département (*numDep*) en question.
- *description* : elle peut prendre 4 valeurs possibles qui sont :
  - ★ 'NOUVEAUX DC' : lors d’insertion d’une nouvelle entrée dans *Incidence.*
  - ★ *'MAJ + DC' :* lors d’une mise à jour d’une entrée dans *Incidence.*
  - ★ 'MAJ - DC' : lors d’une mise à jour d’une entrée dans *Incidence.*
  - ★ 'SUPPRESSION' : lors d’une suppression d’une entrée dans la table *Incidence*.
- *nombre :* Nombre de décès ce jour en question (*date_dc*) et dans ce département en question (*numDep*); (s’il est négatif, c’est qu’il y a une mise à jour d’une entrée dans cette table, et qu’il y a eu lieu finalement moins de décès que ce qui a été déclaré auparavant).