**Equipe :**  
Hichem CHERAFI  
Tom NORMAND  
Ilhan KOROGLU

Nassim DJOUBRI

Sulivan TULOUP

PROJET Cloud Healthcare Unit

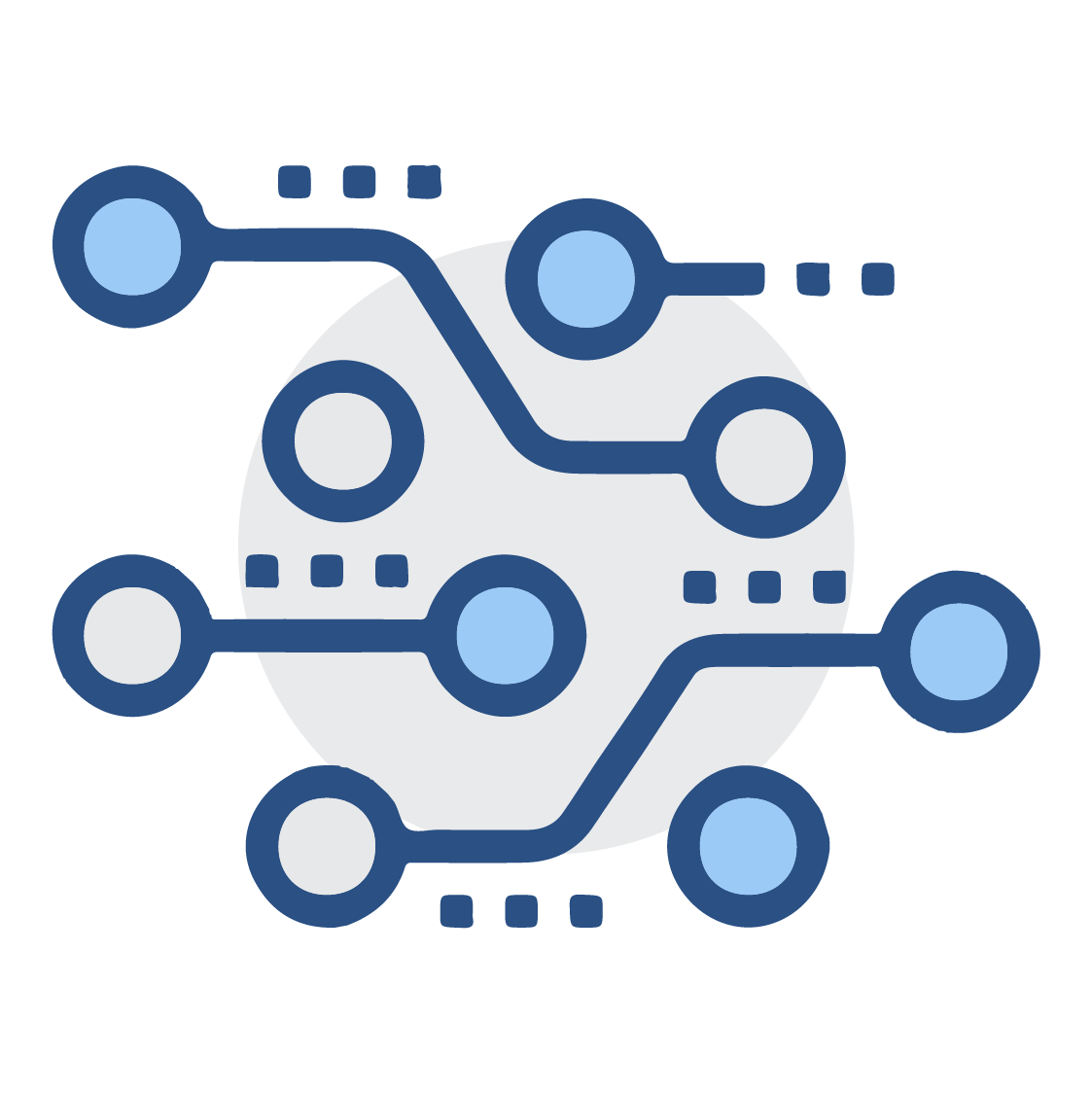


Table des matières

[Contexte 2](#_Toc132194606)

[Contexte du Projet 2](#_Toc132194607)

[Contexte du Livrable 3](#_Toc132194608)

[ARCHITECTURE 4](#_Toc132194609)

[MCD 5](#_Toc132194610)

[Développement des jobs d’alimentation du schéma décisionnel 7](#_Toc132194611)

[CSV 9](#_Toc132194612)

[Enquête de satisfaction 10](#_Toc132194613)

# Contexte

## Contexte du Projet

Le groupe CHU (Cloud Healthcare Unit) a pris conscience de l'importance de la transformation digitale dans le secteur hospitalier. Ils ont donc sollicité notre service pour les aider à mettre en place leur propre entrepôt de données. L'objectif est d'exploiter les données générées par les systèmes de gestion de soins et les systèmes FTP pour répondre aux besoins et exigences d'accès et d'analyse des utilisateurs.

Notre mission est de fournir une solution complète en termes de modèles, d'outils et d'architecture pour extraire, stocker, explorer et visualiser les données selon différents critères. Nous devons également intégrer les données des fichiers distribués dans une source unique persistante et recenser les besoins des utilisateurs en termes d'analyse des données pour l'exploitation directe sur le suivi des patients au niveau national et à long terme.

Nous devons proposer des outils d'intégration, de stockage et de logiciel de visualisation adapté ainsi que l'exploration de données en toute sécurité pour favoriser une meilleure prise de décision. Tout cela doit être réalisé en respectant les exigences d'infrastructure imposées par les données médicales, telles que le ratio coût-efficacité, la sécurité, l'élasticité et la scalabilité.

Les données à exploiter proviennent d'historiques des systèmes de gestion des soins (incluant les fichiers de satisfaction et de décès sur FTP) sur plusieurs années. Nous devons donc gouverner l'infrastructure de données avec une haute sécurité étant donné la sensibilité des informations traitées.

Les utilisateurs souhaitent des analyses sur le taux de consultation et d'hospitalisation des patients en France, par sexe, par âge, par diagnostic, par professionnel, par localisation, ainsi que le taux de satisfaction par région.

Les grandes phases du projet comprennent l'étude de l'architecture à mettre en place, la modélisation conceptuelle des données, l'implémentation physique des données, l'exploitation, la mesure de performance et les optimisations.

## Contexte du Livrable

Le Livrable 2 de ce projet sera composé de trois parties principales : le modèle physique des données, les différents jobs répondant aux besoins utilisateurs, et les éléments justifiant la performance de nos outils.

Le premier script consiste à créer et charger les données dans les tables de l'entrepôt de données, en utilisant le modèle physique de données conçu pour répondre aux besoins utilisateurs. La vérification des données présentes et l'accès aux données à travers les tables sont également nécessaires pour assurer la qualité des données et garantir une bonne performance d'accès aux données.

Un script de peuplement des tables est également requis pour alimenter les tables en données historiques et récentes, en utilisant les différentes sources de données.

Le partitionnement et les buckets doivent être mis en place pour améliorer la performance d'accès aux données, en répartissant les données sur plusieurs nœuds.

Des graphes doivent être générés pour évaluer la performance d'accès à l'entrepôt de données, en mesurant les temps de réponse pour différentes requêtes. Ces graphes permettront de visualiser les résultats et d'identifier les points à améliorer.

Enfin, des requêtes doivent être développées pour évaluer la performance de l'entrepôt de données en termes de temps de réponse, en se basant sur les besoins de l'entreprise en matière d'analyse des ventes, des stocks, des comportements d'achat des clients, etc.

## JOBS

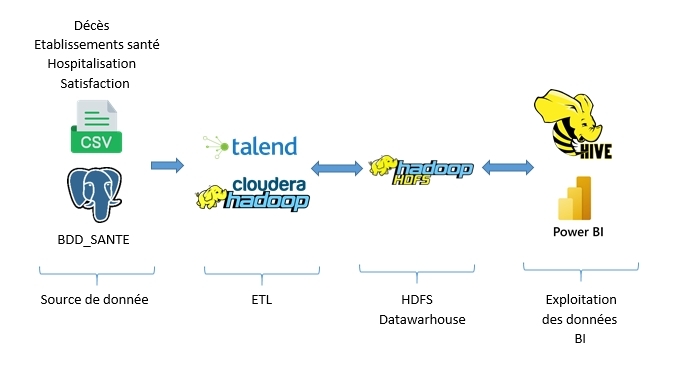
Voici l’architecture que le CHU possédait avant notre arrivée :

Une image contenant diagramme

Description générée automatiquement

Celle-ci ne répondait pas ne répondait pas aux besoins car le stockage des données s’effectuaient dans l’HDFS avant le traitement de celle-ci.

Nous avons donc mis en place une nouvelle architecture :



La nouvelle architecture permettra alors de traiter les données avant de stocker celles-ci afin de

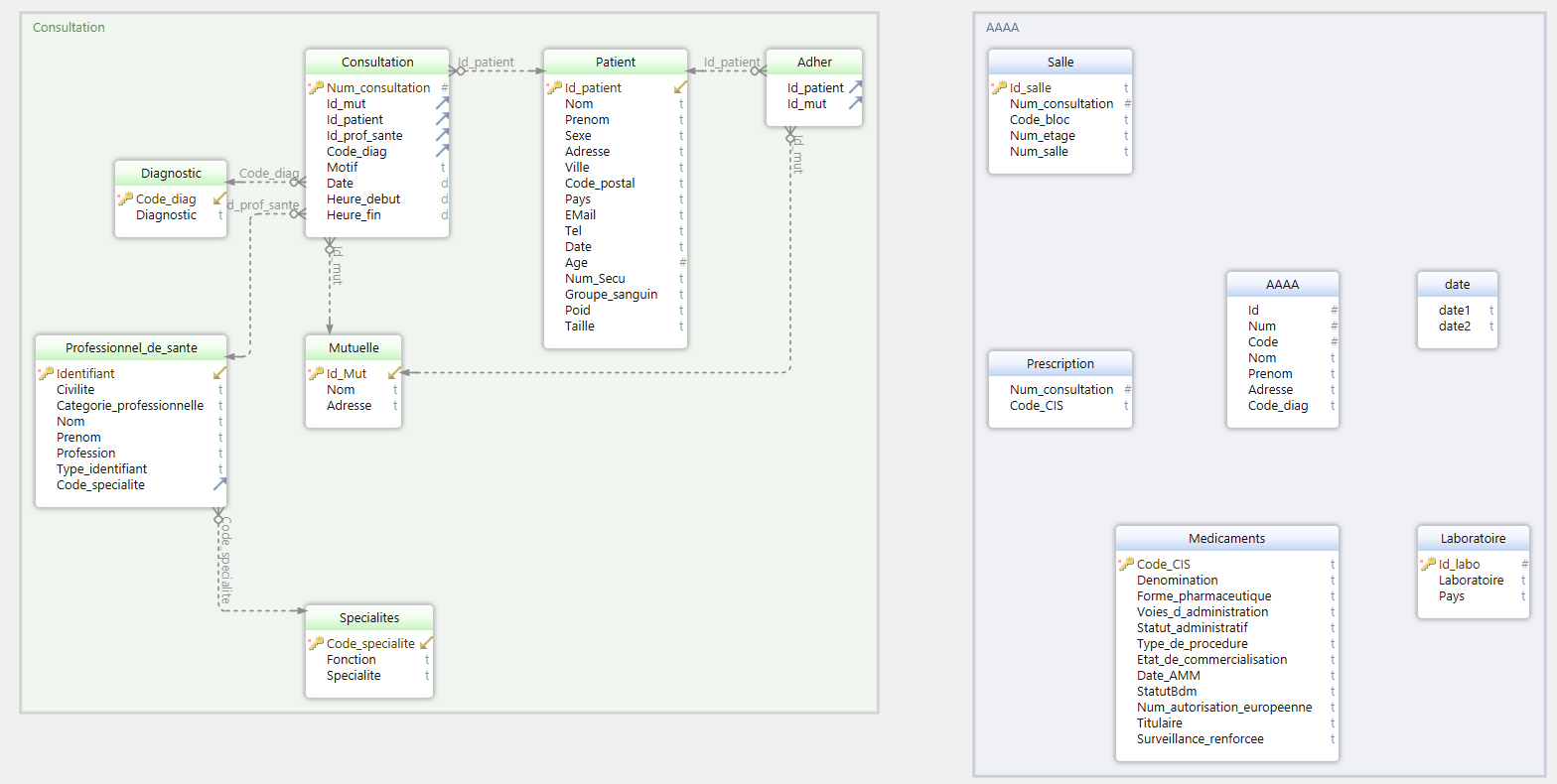
répondre aux besoins du CHU.

## MCD

Nous avons commencé ce travail par un recensement de toutes les données que l’on avait à traiter.

Nous avons trouvé :

-Une BDD Postgre accessibles avec des identifiant/mots de passe fournir par le client. (Nous avons donc utilisé un outil pour extraire le mcd de cette base pour avoir une meilleure visualisation de cette dernière)



-Des CSV de données

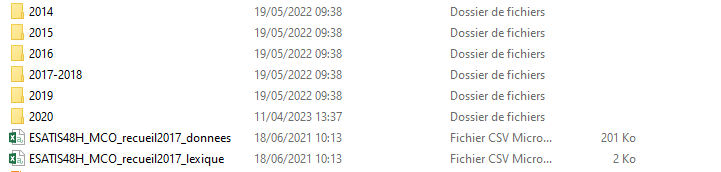
Insertion de l’image...

Une image contenant texte

Description générée automatiquement



-Des CSV/Excel de retour d'enquête satisfaction



Une fois ces plusieurs source de donnée repéré nous avons procéder au recensement de toutes les données qu’elles contenaient pour pouvoir dresser un dictionnaire de données global :

(CF : Excel “dictionnaire de données” disponible en pj)

Suite à ce grand travail de recensement nous sommes passé à l‘analyse des données pour repérer les données redondante, inutilisé, mal indenté ou au mauvais endroit. Puis nous avons proposé une amélioration.

Voici notre MCD amélioré qui respecte les règles de modélisations et intègre les données provenant csv (table en vert).

Une image contenant diagramme, schématique

Description générée automatiquement

(Cf : MCD v3.1)

# Développement des jobs d’alimentation du schéma décisionnel

Un job en Big Data qui implique de transférer un fichier CSV vers un système de fichiers Hadoop HDFS est habituellement lié à une opération appelée "ingestion de données". Cette étape est essentielle dans l'analyse de données en environnement Big Data car elle implique la migration des données brutes depuis une source externe vers un système de stockage distribué, tel que HDFS.

**POSTGRE :**

Concernant notre base de données sous Postgre, elle contient plusieurs tables, les types sont déjà définis de base mais certains ne sont pas bonsrectes. Pour remédier à ce problème, nous allons utiliser des jobs.

* Table “Consultation” :

La table consultation comprend les données liées aux consultations, tels que le numéro de consultation, l’id du patient, l’id du professionnel etc...

Les types sont bons, nous utiliserons un job pour extraire les données.

* Table “Diagnostic” :

La table diagnostic comprend les données liées aux diagnostics, tels que le diagnostic en lui-même et le code de ce dernier.

Les types sont bons, nous utiliserons un job pour extraire les données.

* Table “Medicaments” :

La table médicaments comprend les données liées aux médicaments, tels que son titulaire ou son type de procédure.

L’attribut Date\_AMM est en varchar, nous utiliserons un job pour transformer le type “varchar” en “Date” uniformisé sous la forme JJ/MM/AAAA et extraire les données.

* Table “Mutuelle” :

La table mutuelle comprend les données liées aux mutuelles, tels que leur nom et leur adresse.

Les types sont bons, nous utiliserons un job pour extraire les données.

* Table “Patient” :

La table patient comprend les données liées aux patients, tels que leur nom, prénom, sexe, adresse etc...

L’attribut Date est en varchar, nous utiliserons un job pour transformer le type “varchar” en “Date” uniformisé sous la forme JJ/MM/AAAA et extraire les données. L’attribut Poids est en varchar, nous utiliserons un job pour transformer le type “varchar” en “float” car le poids peut contenir une virgule.

Nous allons extraire les données.

* Table “Professionnel\_de\_sante” :

La table professionnelle de sante contient des données tels que le nom, prénom, et fonction du professionnel.

Les types sont bons, nous utiliserons un job pour extraire les données.

* Table “Salle” :

La table Salle comprend des données liées aux salles, tels que son identifiant, numéro d’étage etc...

L’attribut “Id\_Salle” est en varchar, nous utiliserons un job pour transformer le type “varchar” en “int” car il ne contient que des chiffres.

Nous allons extraire les données.

* Table “Specialites” :

La table spécialité comprend des données liées aux spécialités, tels que la fonction et la spécialité en elle-même.

Les types sont bons, nous utiliserons un job pour extraire les données.

## CSV

Concernant les fichiers CSV, aucun format n’est défini pour les données. On se doit de créer un job qui extrait les données selon un certain format pour chaque fichiers CSV.

* Fichier : Décès.csv :

Ce fichier répertorie tous les décès avec différentes informations sur le patient. Les données sont extraites et normalisées dans une table décès.

Le nom, le prénom seront sous le format varchar.   
Le sexe actuellement écris par 1 pour homme et 2 pour femme seront remplacés par H et F.   
Les dates de naissance seront uniformisées sous la forme JJ/MM/AAAA.  
Les deux codes postaux code\_lieu\_naissance et code\_lieu\_deces sont des varchar car ils peuvent être composés de chiffres et de lettres.  
Le lieu de naissance, le pays de naissance sont eux aussi des varchar.  
La date de décès est une date au format JJ/MM/AAAA.  
Le numéro d’acte de décès est un varchar.

Ce job cherche les données dans ce fichier et les extraits dans la table Décès (cf. MCD).

* Fichier : Etablissement\_sante.csv :

Ce fichier liste les différents établissements de santé. Nous avons normalisé les données comme suit.

L’adresse, le cedex, le code de la commune, le code postal, la commune, le complément destinataire, le complément de point géographie, l’email, l’enseigne commerciale du site, l’identifiant d’organisation, l’indice de répétition de voie, la mention de distribution, le pays, la raison sociale du site, le type de voie et la voie sont extraits en tant que varchar.

Le finess de l’établissement juridique, le finess du site, le numéro de voie, le siren du site et le siret du site, la télécopie, les numéros de téléphone sont eux extraits en entier.

Ce job cherche les données dans ce fichier et les extraits dans la table Etablissement\_sante (cf. MCD).

* Fichier : Professionnel \_sante :

Nous avons également un CSV en plus de la base de données POSTGRE, nous allons donc effectuer un job qui va comparer la base de données POSTGRE ainsi que le fichier CSV pour récupérer tous les professionnels de santé.

Ce job cherche les données dans ce fichier et les extraits dans la table Professionnel\_de\_sante (cf. MCD).

## Enquête de satisfaction

Ce fichier permet de répertorier l’avis des gens en fonction des années, les patients ont répondu à un questionnaire et plusieurs informations sont ressortis.

Nous allons donc garder les informations utiles, les questionnaires sont différents entre les patients restés plus de 48h et ceux de moins de 48h. Nous avons la table esatisca48H-mco et esatisca-mco.

Les données récupérées sont les mêmes pour les deux tables.

* Finess est un code qui permet d’identifier les établissements sanitaires et sociaux, il est donc en entier long.
* Région nous permet d’obtenir le nom d’une région donc en varchar.
* Score\_all\_rea\_ajust et score\_all\_ajust nous donne le résultat final en pourcentage de l’enquête de satisfaction, c’est donc en réel.
* Année est en entier sous la forme JJ/MM/AAAA.