

Livrable 2



Antoine Sire (chef de projet)

Yann Subts

Tayana Petro

CONTEXTE

Pour continuer dans l'amélioration des performances et la qualité des soins, les praticiens (médecin, personnel infirmier) et administrateurs doivent pouvoir accéder directement aux informations exploitable dans les données médicales.

Il est donc impératif de s'occuper d'informations relatives inexploitées, en effectuant des analyses basées sur des données agrégées, consolidées, historiques et synthétisées.

Le secteur doit s'investir dans le développement des systèmes informatiques évolutifs, qui comprennent un ensemble d'outils et de mécanismes pour charger, extraire et traiter les données médicalement.

Le groupe CHU (Cloud Healthcare Unit) a ainsi pris conscience de l'intérêt, voire de la nécessité d'une transformation digitale majeure.

Notre service est donc sollicité pour l'aider à mettre en place son propre entrepôt de données en répondant à des besoins et exigences d'accès et d'analyses des utilisateurs.

TABLE DES MATIERES

Contexte	2
Introduction	4
Scripts et requêtes	5
Conclusion	9
Bilans personnels	10
Webographie	11

INTRODUCTION

Suite à une représentation des données de manière conceptuelle, ainsi qu'une description de l'architecture des données, nous allons poursuivre dans le développement des jobs d'alimentation.

Notre mission consiste à effectuer le modèle physique et l'évaluation de performance par rapport au temps de réponse des requêtes appliquées sur les tables.

Pour ce faire, nous réaliseront des scripts, des graphes et des requêtes qui permettraient de répondre à ces demandes.

SCRIPTS ET REQUETES

Pour continuer dans l'exploitation et l'implémentation des données, il est primordial de représenter un modèle physique et de faire une évaluation de performance par rapport aux temps de réponse des requêtes réalisées sur les tables.

Pour ce faire, des scripts, graphes, vérifications et des requêtes sont à effectuer.

Commençons par illustrer quelques scripts et requêtes utilisés à travers les différentes tables :

- Script pour la création et le chargement des données dans les tables

Exemple table Hospitalisation :

```
CREATE TABLE hospitalisation (
    num_hospitalisation INT,
    id_patient INT,
    identifiant_organisation STRING,
    code_diagnostic STRING,
    suite_diagnostic_consultation STRING,
    date_entree STRING,
    jour_hospitalisation INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ';'
```

La table est créée avec la commande « CREATE TABLE », suivie des différents champs qui la constituent, accompagnés de leurs types.

Le « ROW FORMAT DELIMITED » est utilisé pour stocker les données dans la ligne HDFS à un format délimité.

On a également « FIELD TERMINATED BY ‘;’ » qui est utilisé pour stocker les données dans le HDFS où les colonnes sont séparées par le caractère spécifié.

Les champs et les lignes se terminent donc par un point-virgule « ; »

Exemple table Satisfaction :

```
CREATE TABLE Satisfaction (
    finesse STRING,
    rs_finesse STRING,
    region STRING,
    score_all FLOAT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ';'
```

Exemple table Consultation :

```
CREATE TABLE consultation (
    num_consultation INT,
    id_mut INT,
    id_patient INT,
    id_prof_sante STRING,
    code_diag STRING,
    motif STRING,
    date_consul STRING,
    heure_debut STRING,
    heure_fin STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ';'
LOAD DATA INPATH '/user/cloudera/projet2/Consultation'
INTO TABLE consultation
```

Les données dans Hive ont été chargées depuis HDFS vers la ruche à l'aide de la commande « LOAD DATA INPATH » avec comme chemin d'entrées des données '/user/cloudera/projet2/Consultation' dans la table « Consultation ».

➤ Script montrant le peuplement des tables

Exemple table Hospitalisation :

```
INSERT OVERWRITE TABLE hospitalisation_buckets
PARTITION (annee)
SELECT num_hospitalisation, id_patient, identifiant_organisation,
code_diagnostic, suite_diagnostic_consultation, jour_hospitalisation,
YEAR(date_entree) FROM hospitalisation
```

L'instruction « INSERT OVERWRITE TABLE » écrase les données existantes dans la table « hospitalisation_buckets » à l'aide des nouvelles valeurs. Les enregistrements de la partition spécifiée sont supprimés et remplacés par des nouveaux.

Exemple table Satisfaction :

```
INSERT OVERWRITE TABLE satisfaction_buckets
PARTITION (regions)
SELECT finesse, rs_finesse, score_all, region
FROM satisfaction
```

➤ Script pour le partitionnement et les buckets

Partitionner permet d'avoir un accès plus rapide aux données pour améliorer les performances. Une fois la table partitionnée, elle permet ainsi d'optimiser le stockage d'informations.

Exemple table Hospitalisation :

```
CREATE TABLE hospitalisation_buckets (
    num_hospitalisation INT,
    id_patient INT,
    identifiant_organisation STRING,
    code_diagnostic STRING,
    suite_diagnostic_consultation STRING,
    jour_hospitalisation INT)
PARTITIONED BY (annee STRING)
CLUSTERED BY (num_hospitalisation) INTO 5 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
```

Le « Clustered By » divise les tables dans des buckets au nombre de cinq.

Exemple table Satisfaction :

```
CREATE TABLE satisfaction_buckets (
    finesse STRING,
    rs_finesse STRING,
    score_all FLOAT)
PARTITIONED BY (regions STRING)
CLUSTERED BY (finesse) INTO 5 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
```

Exemple table Consultations :

```
CREATE TABLE consultation_buckets (
    num_consultation INT,
    id_mut INT,
    id_patient INT,
    id_prof_sante STRING,
    code_diag STRING,
    motif STRING,
    heure_debut STRING,
    heure_fin STRING)
PARTITIONED BY (annee STRING)
CLUSTERED BY (num_consultation) INTO 5 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ';'
LOAD DATA INPATH '/user/cloudera/projet2/Satisfaction'
INTO TABLE satisfaction
```

La vérification des données présentes et l'accès à travers les tables est vue avec un « SELECT * FROM table ».



The screenshot shows a database interface with the following details:

- Query History: Shows a single query: "SELECT * FROM consultation".
- Saved Queries: None.
- Results (100+): This is the active tab.
- Table Headers: The table has ten columns: consultation.num_consultation, consultation.id_mut, consultation.id_patient, consultation.id_prof_sante, consultation.code_diag, consultation.motif, consultation.date_consul, and consultation.
- Data Rows: Five rows of data are displayed, each corresponding to a consultation entry with unique values for all columns.

	consultation.num_consultation	consultation.id_mut	consultation.id_patient	consultation.id_prof_sante	consultation.code_diag	consultation.motif	consultation.date_consul	consultation.
1	1059023405	123	5725	10000012483	A066	Consultation	2016-10-30	01-01-1970
2	1059023406	75	19907	10000034487	A064	Consultation	2020-11-06	01-01-1970
3	1059023407	137	34905	10000084516	A065	Consultation	2018-07-09	01-01-1970
4	1059023408	182	50924	10000100205	K610	Prélèvement laboratoire	2017-05-19	01-01-1970
5	1059023409	218	1222	10000108083	K612	Mammographie	2018-02-14	01-01-1970

CONCLUSION

Après avoir appliqué différentes requêtes et mis en place une modélisation physique ainsi qu'une évaluation des performances, les tables sont donc optimisées et respectent les différentes analyses effectuées depuis le début de notre projet.

Nous devrons présenter les résultats d'analyse à travers un tableau de bord, dans lequel sera construit un récit basé sur les requêtes finales à travers des graphiques et des tableaux.

BILANS PERSONNELS

Bilan chef de projet : Les tâches réparties ont été similaires, étant donné la concentration maximale sur l'évaluation des performances et le modèle physique.

Antoine SIRE : Pour ce livrable, je me suis chargé de réaliser les requêtes permettant la construction des tables et leur peuplement, conjointement à mes camarades de projet.

Yann SUBTS : Lors de ce livrable j'ai travaillé avec mon groupe sur les scripts pour le partitionnement et les buckets et sur le peuplement des tables.

Tayana PETRO : Je me suis tournée vers la réalisation des requêtes qu'on a pu mettre en commun avec chaque membre du groupe.

WEBOGRAPHIE

1. Ressources des prosits
2. Corbeilles et Workshops
3. <https://cesi.moodle.fr>
4. [Talend](#)