

Projet big data mbds

*Marrakech de 20-01-2023 a 15-04-2023*

Nom des étudiants :

Wiam LAQSSIR

Badreddine TIRGANI

Achraf [ABOUELAOUD](https://lms.univ-cotedazur.fr/2022/user/view.php?id=36298&course=15577)

Enseignants encadreurs :

Gabriel MOPOLO

Sergio Simonian

Nicolas Pasquier

**Résumer**

Ce résumé présente le travail réalisé dans le cadre d'un projet Big Data. Le but de ce travail était de traiter, de stocker et d'analyser de grandes quantités de données pour en extraire des informations exploitables.

Le projet Big Data est dans le cadre d’approfondir nos connaissances dans les trois modules qu’on a vu les derniers mois cette années scolaire au niveau de Master 2 MIAGE MBDS. Dans ce sens on a construit un groupe de trois personnes.

Afin de réaliser ce projet on a utilisé une machine virtuelle au niveau de Vagrant, ou on a installé tous les outils qu’on a besoin tell que : Hadoop, Spark, KV store, Hive, MongoDB, R…

Les principaux résultats de ce travail étaient la capacité à identifier des tendances, des modèles et des corrélations dans les données, ainsi que la possibilité de prédire des comportements futurs.

En résumé, ce projet Big Data a permis de tirer des enseignements importants sur la manière de collecter, de traiter, de stocker et d'analyser de grandes quantités de données pour en extraire des informations précieuses.

**Abstract:**

This project gives us a chance to explore a very important side of our work and how we can manage it. The data and all its approaches (collect, analyses…) consider as a skeleton of all the software engineering. During a month and a half, we are using the data already collected by our school to construct a lake of data after preparing and adapting all the data in the compatible structure to use it in a future prediction which we are going to talk about in the next chapiters in details.

**Liste des figures**

[Figure 1, 1 : clonner la machine virtuelle en guthub 6](#_Toc132242442)

[Figure 1, 2 : lancer vagrant 6](#_Toc132242443)

[Figure 1, 3 : installer les composants 6](#_Toc132242444)

[Figure 1, 4 : les fichiers CSV 7](#_Toc132242445)

[Figure 1, 5 : l’affichage des fichier csv en vagrant 7](#_Toc132242446)

[Figure 1, 6 : accéder a vagrant 8](#_Toc132242447)

[Figure 1, 7 : lacement de HDFS 8](#_Toc132242448)

[Figure 1, 8 : KVSTORE 8](#_Toc132242449)

[Figure 1, 9 : SQL shell 8](#_Toc132242450)

[Figure 1, 10 : script du table client NOsql 9](#_Toc132242451)

[Figure 1, 11 : table marketing NOsql 9](#_Toc132242452)

[Figure 1, 12 : script du table immatriculation NOsql 9](#_Toc132242453)

[Figure 1, 13 : l’erreur de l’espace et l’accent 10](#_Toc132242454)

[Figure 1, 14 : création du table client 10](#_Toc132242455)

[Figure 1, 15 : l’affichage des tables NoSql 11](#_Toc132242456)

[Figure 1, 16 : l’affichage de la table nosql catalogue 11](#_Toc132242457)

[Figure 1, 17 : l’affichage de la table nosql marketing 11](#_Toc132242458)

[Figure 1, 18 : l’affichage de la table nosql immatriculation 12](#_Toc132242459)

[Figure 1, 19 : l’affichage de la table nosql client 12](#_Toc132242460)

[Figure 1, 20 : lancement de mongoDB 12](#_Toc132242461)

[Figure 1, 21 : remplissage de collection immatriculation 12](#_Toc132242462)

[Figure 1, 22 : l’affichage de la collection immatriculation 13](#_Toc132242463)

[Figure 1, 23 : remplissage de la collection client 13](#_Toc132242464)

[Figure 1, 24 : affichage de la collection client 13](#_Toc132242465)

[Figure 1, 25 : remplissage de la collection catalogue 13](#_Toc132242466)

[Figure 1, 26 : l’affichage de la collection catalogue 13](#_Toc132242467)

[Figure 1, 27 : l’affichage de la collectiom marketing 14](#_Toc132242468)

[Figure 1, 28 : l’affichage de fichier marketing.csv en hdfs 14](#_Toc132242469)

[Figure 1, 29 : l’affichage de tous les fichiers de projet big data 15](#_Toc132242470)

[Figure 1, 30 : Hive 15](#_Toc132242471)

[Figure 1, 31 : connexion a beeline 15](#_Toc132242472)

[Figure 1, 32 : table externe versimmatriculation nosql 16](#_Toc132242473)

[Figure 1, 33 : table externe vers client nosql 17](#_Toc132242474)

[Figure 1, 34 : table externe vers marketing nosql 18](#_Toc132242475)

[Figure 1, 35 : table externe vers catalogue nosql 18](#_Toc132242476)

[Figure 1, 36 : table externe vers catalogue mongoDb 19](#_Toc132242477)

[Figure 1, 37 : table externe vers immatriculation mongoDb 20](#_Toc132242478)

[Figure 1, 38 : table externe vers client mongoDb 21](#_Toc132242479)

[Figure 1, 39 : table externe vers marketing mongoDb 22](#_Toc132242480)

[Figure 1, 40 : les tables externes vers mongo 22](#_Toc132242481)

[Figure 1, 41 : table externe vers le fichier immatriculation 23](#_Toc132242482)

[Figure 1, 42 : table externe vers le fichier client 23](#_Toc132242483)

[Figure 1, 43 : table externe vers le fichier marketing 24](#_Toc132242484)

[Figure 1, 44 : table externe vers le fichiers catalogue 25](#_Toc132242485)

**Liste des acronymes**

HDFS. *Hadoop Distributed File System*

KVstore. Key Value store

nosql. Not Only SQL

sql. *Structured Query Language*

YARN. Yet Another Resource Negotiator

**Plan du document**

[1. Introduction générale 3](#_Toc132121808)

[2. Présentation du projet 4](#_Toc132121809)

[3. Répartition du travail en membre du groupe 4](#_Toc132121810)

[4. Architecture du data lake 4](#_Toc132121811)

[5. Construction du data lake par étape 4](#_Toc132121812)

[6. Hadoop Map Reduce 4](#_Toc132121813)

[7. Visualisation de données avec des outils de DataViz (si concerné) 4](#_Toc132121814)

[8. Analyse de données avec des outils de machine learning (R, …) 5](#_Toc132121815)

[9. Conclusion générale 5](#_Toc132121816)

[10. Références et Bibliographie 5](#_Toc132121817)

[11. Annexes 5](#_Toc132121818)

[11.1 Vidéo de présentation 5](#_Toc132121819)

[11.2 Dossier contenant les scripts et programmes de construction du lac de données 5](#_Toc132121820)

[11.3 Dossier contenant les scripts et programmes Hadoop Map Reduce 5](#_Toc132121821)

[11.4 Dossier contenant les scripts et programmes de visualisation de données 5](#_Toc132121822)

[11.5 Dossier contenant les scripts et programmes d’analyse de données 5](#_Toc132121823)

1. Introduction générale

**« Explorez l'océan de données pour découvrir des perles d'informations précieuses »**

Pour avoir la possibilité d’exploiter et de découvert l’importance d’analyser l’informations collecter et stocker des êtres humains, animaux, objets… On a essayé de réaliser un projet de big data avec des milliers des informations bruts.

Dans ce sens on a suivi plusieurs étapes pour atteindre l’objectif de prédire le future et les raisons des événements dans les différents aspects à partir d’historique. On doit nettoyer les données brutes pour construire un Lake de données qui présenter un point de rencontre entre les différents systèmes de stockage des données. Puis on doit préparer et adapter le fichier CO2.csv pour intégrer le contenu dans la table catalogue avec un programme Map/Reduce afin d’accomplir nos données. Afin de construire un modelé en R qui permet de prédire les catégories de véhicules à proposer aux clients de la concession automobile.

Ce rapport va contient en générale la description des trois chapitres, le premier chapitre qui va décrit la partie de préparation des données et de construction du Lake, puis la partie de Hadoop par laquelle on complimenter notre data et finalement la partie ou on prédire la catégorie compatible avec chaque client.

2. Présentation du projet

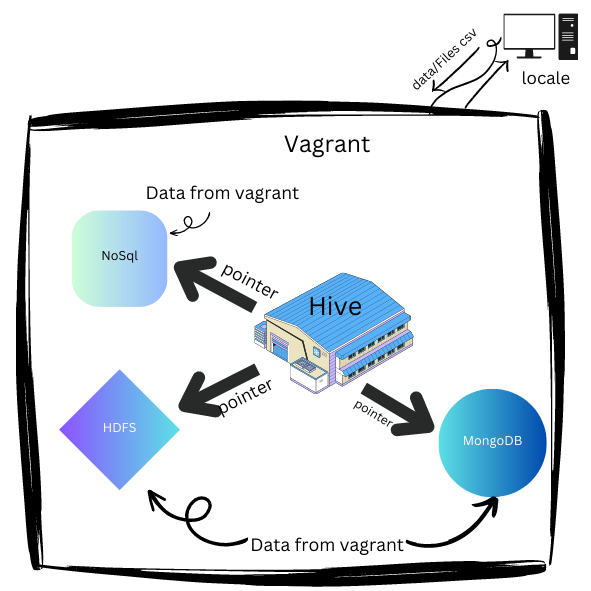
En équipe on a réalisé le projet big data qu’est en relation avec trois modules qu’on a dans master 2. La partie de l’architecture qui contient un ensemble des bases de données avec différents types et un Lake qui permet de l’accès à toutes les bases de données. Ensuit la partie Hadoop qui nous permet de préparer et compléter les données et les adapter. Et finalement avoir une base de données prêt par laquelle on peut construire un modèle qui prédit les catégories des automobiles pour chaque client.

3. Répartition du travail en membre du groupe

Dans le cadre de réalisation de notre projet big data on a construit une équipe de trois personnes. On a devisé le travail comme suit ; la première partie de l’architecture faite par Wiam LAQSSIR, la deuxième partie de Hadoop réaliser par Achraf ABOUALAOUD et la dernière partie de data mining résolu par Badreddine TIRGANI

4. Architecture du data lake

On a réalisé une architecture comme suit :



5. Construction du data Lake par étape

Dans le cadre de création de l’architecture en haut on utiliser la machine virtuelle en vagrant et une liste des commandes qu’on va voir dans la suite.

* Préparation de la machine virtuelle

On a pris une copie de la machine virtuelle vagrant en GitHub avec la commande clone.

A picture containing text, screenshot, font

Description automatically generated

Figure 1, 1 : clonner la machine virtuelle en github

Puis on a lancer notre machine virtuelle afin qu’on puisse accéder les composants (KVstore,mongoDB…):

A screen shot of a computer

Description automatically generated with medium confidence

Figure 1, 2 : lancer vagrant

On a réinstallé les composants (Hadoop, kvstore…) puisqu’ils ne sont pas bien fonctionnés :

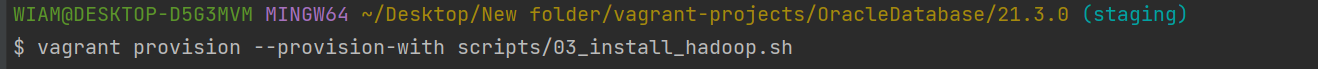


Figure 1, 3 : installer les composants

pour Kvstore : vagrant provision --provision-with scripts/05\_install\_kvstore.sh.

pour Hive: vagrant provision --provision-with scripts/06\_install\_hive.sh.

pour mongodb vagrant provision --provision-with scripts/07\_install\_mongodb.sh, mais on a la remplace par: vagrant provision --provision-with scripts/99\_downgrade\_mongodb.sh

On a copie les fichiers csv dans vagrant vagrant-projects/OracleDatabase/21.3.0

A screenshot of a computer program

Description automatically generated with medium confidence

Figure 1, 4 : les fichiers CSV

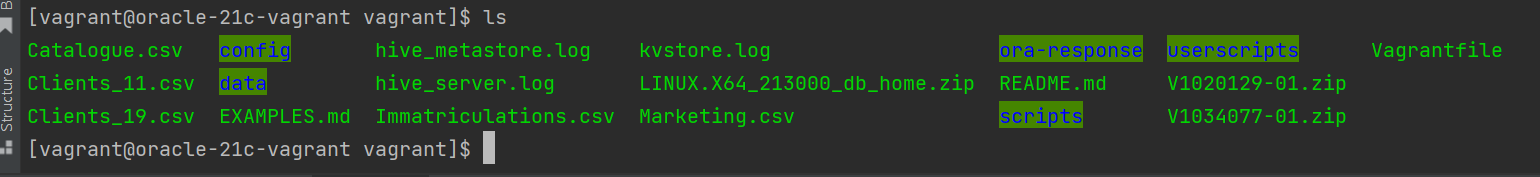


Figure 1, 5 : l’affichage des fichier csv en vagrant

Accéder à la machine virtuelle vagrant (vagrant ssh) :

A screenshot of a computer program

Description automatically generated with medium confidence

Figure 1, 6 : accéder à vagrant

* La partie Not Only SQL (nosql) Key Value store (KVstore):

Le Hadoop Distributed File System (HDFS) et Yet Another Resource Negotiator (YARN) doivent être déjà lancer:

A screenshot of a computer program

Description automatically generated with medium confidence

Figure 1, 7 : lacement de HDFS

Lancer le KVstore :

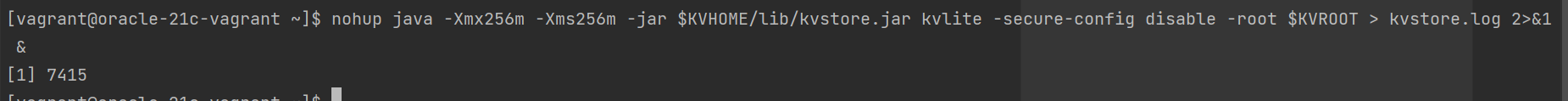


Figure 1, 8 : KVSTORE

Lancer le sigle de Structured Query Language (sql) shell :

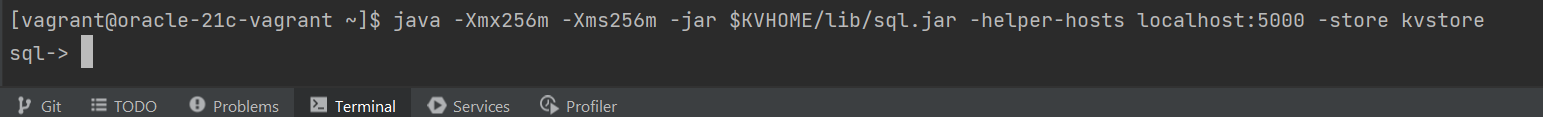


Figure 1, 9 : SQL shell

[1]Creation des tables :

CREATE table nosql\_client11(

AGE INTEGER,

SEXE ENUM(M,F,vide) DEFAULT vide,

TAUX INTEGER,

SITUATION\_FAMILIALE ENUM (Celibataire, Divorcee,En\_Couple, Marie,Seul,Seule,vide) DEFAULT vide,

NB\_ENFANTS\_ACHARGE INTEGER,

DEUXIEME\_VOITURE BOOLEAN,

IMMATRICULATION STRING,

PRIMARY KEY(AGE)

);

Figure 1, 10 : script du table client NOsql

CREATE table nosql\_marketing(

AGE INTEGER,

SEXE ENUM(M,F,vide) DEFAULT vide,

TAUX INTEGER,

SITUATION\_FAMILIALE ENUM (vide,Celibataire, Divorcee,En\_Couple, Marie,Seul,Seule) DEFAULT vide,

NB\_ENFANTS\_ACHARGE INTEGER,

DEUXIEME\_VOITURE BOOLEAN,

PRIMARY KEY(AGE)

);

Figure 1, 11 : table marketing NOsql

CREATE table nosql\_immatriculation(

IMMATRICULATION STRING,

MARQUE STRING,

NOM STRING,

PUISSANCE INTEGER,

LONGEUR ENUM (courte, moyenne, longue, tres\_longue),

NBPLACES INTEGER,

NBPORTES INTEGER,

COULEUR ENUM (blanc, bleu, gris, noir, rouge),

OCCASION BOOLEAN,

PRIX INTEGER,

PRIMARY KEY(IMMATRICULATION)

);

Figure 1, 12 : script du table immatriculation NOsql

Le problème est on doit éviter les espaces dans l’enums et les accents et sur tout le mot clé ‘null’ donc on a le remplacer par ‘vide’.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 1, 13 : l’erreur de l’espace et l’accent

A picture containing text, screenshot, font

Description automatically generated

Figure 1, 14 : création du table client

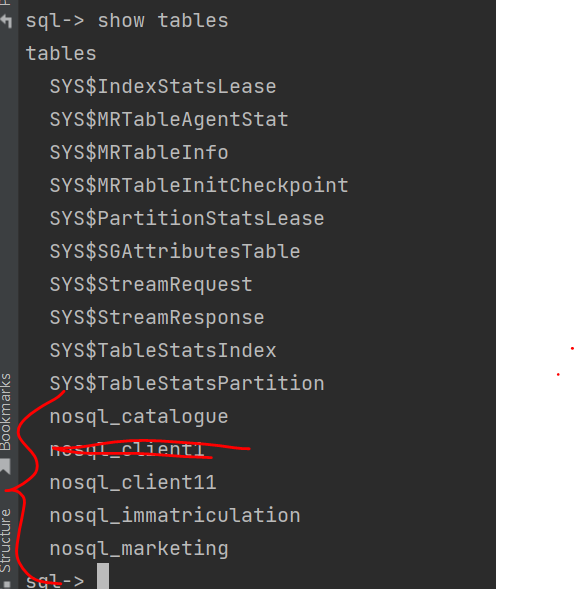


Figure 1, 15 : l’affichage des tables NoSql

Le remplissage des quatre tables par la commande suivant :

[2]sql-> import -table nosql\_catalogue -file CatalogueSave.csv CSV

Avant la remplissage on a remplacé les espaces (En couple 🡺 En\_Couple ) et les accents

A picture containing text, screenshot, font, information

Description automatically generated

Figure 1, 16 : l’affichage de la table nosql catalogue

A screen shot of a computer code

Description automatically generated with low confidence

Figure 1, 17 : l’affichage de la table nosql marketing

A picture containing text, screenshot, font, information

Description automatically generated

Figure 1, 18 : l’affichage de la table nosql immatriculation

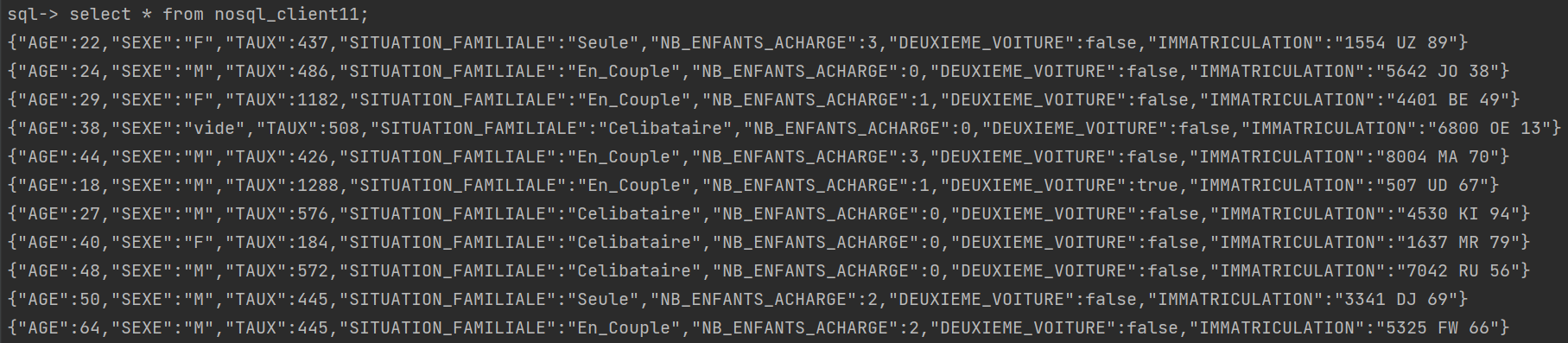


Figure 1, 19 : l’affichage de la table nosql client

* La partie MongoDB: [3]

On a connecter a mongodb avec ‘mongo’ après qu’on a lancer mongoDB

A screen shot of a computer

Description automatically generated with low confidence

Figure 1, 20 : lancement de mongoDB

Remplissage de Collection immatricule (Le fichier contient les noms des colonnes) on a ajouté l’entete parce que les deux colonnes (nombre de place et nombre de porte sont identiques):

A picture containing text, screenshot, font

Description automatically generated

Figure 1, 21 : remplissage de collection immatriculation

A screen shot of a computer

Description automatically generated with low confidence

Figure 1, 22 : l’affichage de la collection immatriculation

Remplissage de collection ‘Client’

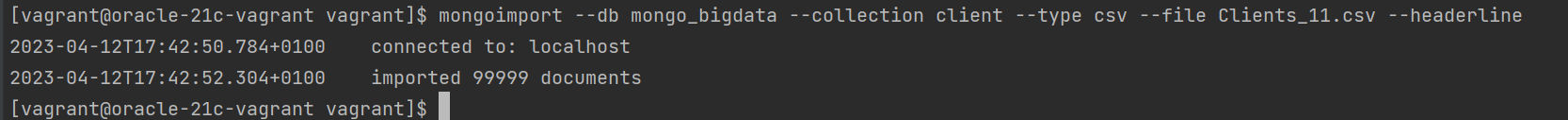


Figure 1, 23 : remplissage de la collection client

A picture containing text, screenshot, font, information

Description automatically generated

Figure 1, 24 : affichage de la collection client

Remplissage de collection ‘Catalogue’

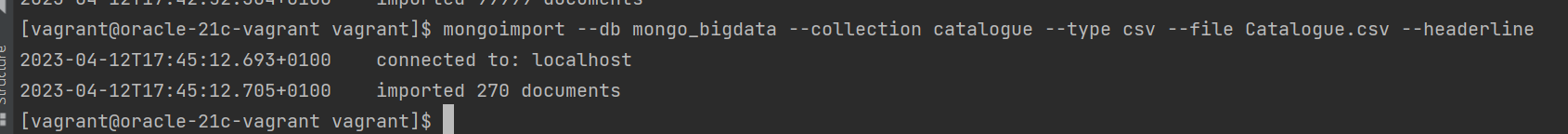


Figure 1, 25 : remplissage de la collection catalogue

A picture containing text, screenshot, font

Description automatically generated

Figure 1, 26 : l’affichage de la collection catalogue

La Collection ‘Marketing’ :

[4] mongoimport --db mongo\_bigdata --collection marketing --type csv --file Marketing.csv --headerline

A picture containing text, screenshot, font, information

Description automatically generated

Figure 1, 27 : l’affichage de la collection marketing

* HDFS:

Pour chaque fichier csv on crée un dossier :

hdfs dfs -mkdir /data\_marketing

hdfs dfs -mkdir /data\_client/hdfs\_client.csv

hdfs dfs -mkdir /data\_immatriculation/hdfs\_immatriculation.csv

hdfs dfs -mkdir /data\_catalogue/hdfs\_catalogue.csv

Pour remplire le fichier en HDFS on utiliser la commande suivant :

hdfs dfs -put Marketing.csv /data\_marketing/hdfs\_marketing.csv

hdfs dfs -put Client.csv /data\_client/hdfs\_client.csv

hdfs dfs -put Immatriculation.csv /data\_immatriculation/hdfs\_immatriculation.csv

hdfs dfs -put Catalogue.csv /data\_catalogue/hdfs\_catalogue.csv

Exemple de fichier marketing puisque le fichier le plus court :

A screen shot of a computer

Description automatically generated with medium confidence

Figure 1, 28 : l’affichage de fichier marketing.csv en hdfs

A screen shot of a computer program

Description automatically generated with low confidence

Figure 1, 29 : l’affichage de tous les fichiers de projet big data

* KVStore:

Lancer la Hive

A screen shot of a computer

Description automatically generated with low confidence

Figure 1, 30 : Hive

Connecter à hive par cette commande :

A screen shot of a computer

Description automatically generated with medium confidence

Figure 1, 31 : connexion a beeline

*Tables externes pointant vers des tables nosql kvstore :*

CREATE EXTERNAL TABLE nosql\_H\_IMMATRICULATION (

IMMATRICULATION string,

MARQUE string,

NOM string,

PUISSANCE int,

LONGEUR string,

NBPLACES int,

NBPORTES int,

COULEUR string,

OCCASION boolean,

PRIX int

)

STORED BY 'oracle.kv.hadoop.hive.table.TableStorageHandler'

TBLPROPERTIES (

"oracle.kv.kvstore" = "kvstore",

"oracle.kv.hosts" = "localhost:5000",

"oracle.kv.hadoop.hosts" = "localhost/0.0.0.0",

"oracle.kv.tableName" = "nosql\_immatriculation");

Figure 1, 32 : table externe versimmatriculation nosql



A screenshot of a computer

Description automatically generated

Exemple de requête :



A screen shot of a computer

Description automatically generated with low confidence

CREATE EXTERNAL TABLE nosql\_H\_CLIENT (

AGE int,

SEXE string,

TAUX int,

SITUATION\_FAMILIALE string,

NB\_ENFANTS\_ACHARGE int,

DEUXIEME\_VOITURE boolean,

IMMATRICULATION string

)

STORED BY 'oracle.kv.hadoop.hive.table.TableStorageHandler'

TBLPROPERTIES (

"oracle.kv.kvstore" = "kvstore",

"oracle.kv.hosts" = "localhost:5000",

"oracle.kv.hadoop.hosts" = "localhost/0.0.0.0",

"oracle.kv.tableName" = "nosql\_client11");

Figure 1, 33 : table externe vers client nosql



A screenshot of a computer

Description automatically generated with medium confidence

CREATE EXTERNAL TABLE nosql\_H\_MARKETING (

AGE int,

SEXE string,

TAUX int,

SITUATION\_FAMILIALE string,

NB\_ENFANTS\_ACHARGE int,

DEUXIEME\_VOITURE boolean

)

STORED BY 'oracle.kv.hadoop.hive.table.TableStorageHandler'

TBLPROPERTIES (

"oracle.kv.kvstore" = "kvstore",

"oracle.kv.hosts" = "localhost:5000",

"oracle.kv.hadoop.hosts" = "localhost/0.0.0.0",

"oracle.kv.tableName" = "nosql\_marketing");

Figure 1, 34 : table externe vers marketing nosql

Select \* from nosql\_h\_marketing;

A picture containing screenshot, text, space, astronomy

Description automatically generated

CREATE EXTERNAL TABLE nosql\_H\_CATALOGUE (

MARQUE string,

NOM string,

PUISSANCE int,

LONGEUR string,

NBPLACES int,

NBPORTES int,

COULEUR string,

OCCASION boolean,

PRIX int

)

STORED BY 'oracle.kv.hadoop.hive.table.TableStorageHandler'

TBLPROPERTIES (

"oracle.kv.kvstore" = "kvstore",

"oracle.kv.hosts" = "localhost:5000",

"oracle.kv.hadoop.hosts" = "localhost/0.0.0.0",

"oracle.kv.tableName" = "nosql\_catalogue");

Select \* from nosql\_H\_CATALOGUE;

Figure 1, 35 : table externe vers catalogue nosql

A screenshot of a computer program

Description automatically generated with medium confidence

*Tables externes pointant vers les collections MongoDB :*

Table externe Catalogue pointer vers la collection Catalogue en mongoDB(Le probleme de null dans les deux colonnes nbrPortes nbrPlaces)

CREATE EXTERNAL TABLE mongo\_catalogue (

id STRING,

MARQUE STRING,

NOM STRING,

PUISSANCE INT,

longueur STRING,

nbPlaces string,

nbPortes string,

COULEUR string,

OCCASION string,

PRIX int

)

STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'

WITH SERDEPROPERTIES('mongo.columns.mapping'='{"id":"\_id"}')

TBLPROPERTIES('mongo.uri'='mongodb://localhost:27017/mongo\_bigdata.catalogue');

Figure 1, 36 : table externe vers catalogue mongoDb

A screen shot of a computer

Description automatically generated with low confidence

Table externe Immatriculation pointer vers la collection immatriculation en mongoDB (Le problème de null dans les deux colonnes nbrPortes nbrPlaces)

CREATE EXTERNAL TABLE mongo\_immatriculation (

id STRING,

immatriculation STRING,

marque STRING,

nom STRING,

puissance INT,

longueur string,

nbPlaces int,

nbPortes int,

couleur string,

occasion string,

prix int

)

STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'

WITH SERDEPROPERTIES('mongo.columns.mapping'='{"id":"\_id"}')

TBLPROPERTIES('mongo.uri'='mongodb://localhost:27017/mongo\_bigdata.immatriculation');

Figure 1, 37 : table externe vers immatriculation mongoDb

A picture containing screenshot, text, space

Description automatically generated

Table externe mongo\_client pointer vers la collection client :

CREATE EXTERNAL TABLE mongo\_client (

id STRING,

age int,

sexe STRING,

taux int,

situationFamiliale string,

nbEnfantsAcharge int,

deuxieme\_voiture string,

immatriculation string

)

STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'

WITH SERDEPROPERTIES('mongo.columns.mapping'='{"id":"\_id"}')

TBLPROPERTIES('mongo.uri'='mongodb://localhost:27017/mongo\_bigdata.client');

Figure 1, 38 : table externe vers client mongoDb

A screenshot of a computer program

Description automatically generated with medium confidence

Table mongo\_marketing pointer vers la collection marketing en mongoDB :

CREATE EXTERNAL TABLE mongo\_marketing (

id STRING,

AGE INT,

SEXE STRING,

TAUX STRING,

SITUATION\_FAMILIALE STRING,

NB\_ENFANTS\_ACHARGE INT,

DEUXIEME\_VOITURE BOOLEAN)

STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'

WITH SERDEPROPERTIES('mongo.columns.mapping'='{"id":"\_id"}')

TBLPROPERTIES('mongo.uri'='mongodb://localhost:27017/mongo\_bigdata.marketing');

Figure 1, 39 : table externe vers marketing mongoDb

A screenshot of a computer

Description automatically generated with medium confidence

Tous les Int sont considéré null même si sont tous existe dans mongoDB, on a essayé tous les types possible mais toujours même problème.

A screenshot of a computer program

Description automatically generated with medium confidence

Figure 1, 40 : les tables externes vers mongo

*Tables externes pointant vers les fichiers HDFS :*

CREATE EXTERNAL TABLE hdfs\_immatriculation (

immatriculation STRING,

marque STRING,

nom STRING,

puissance INT,

longueur string,

nbPlaces int,

nbPortes int,

couleur string,

occasion string,

prix int

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'hdfs:/data/data\_immatriculation';

Figure 1, 41 : table externe vers le fichier immatriculation

select \* from hdfs\_immatriculation limit 5;

A screen shot of a computer

Description automatically generated with medium confidence

CREATE EXTERNAL TABLE hdfs\_client (

AGE INT,

SEXE STRING,

TAUX INT,

SITUATION\_FAMILIALE STRING,

NB\_ENFANTS\_ACHARGE INT,

DEUXIEME\_VOITURE BOOLEAN,

immatriculation string

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'hdfs:/data/data\_client';

Figure 1, 42 : table externe vers le fichier client

select \* from hdfs\_client limit 5

A picture containing screenshot, space, black, astronomy

Description automatically generated

CREATE EXTERNAL TABLE hdfs\_marketing (

AGE INT,

SEXE STRING,

TAUX INT,

SITUATION\_FAMILIALE STRING,

NB\_ENFANTS\_ACHARGE INT,

DEUXIEME\_VOITURE BOOLEAN

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'hdfs:/data/data\_marketing';

Figure 1, 43 : table externe vers le fichier marketing

select \* from hdfs\_marketing limit 5

A picture containing screenshot, space, black, line

Description automatically generated

CREATE EXTERNAL TABLE hdfs\_catalogue (

MARQUE STRING,

NOM STRING,

PUISSANCE INT,

longueur STRING,

nbPlaces string,

nbPortes string,

COULEUR string,

OCCASION string,

PRIX int

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'hdfs:/data/data\_catalogue';

Figure 1, 44 : table externe vers le fichier catalogue

select \* from hdfs\_catalogue limit 5

A screenshot of a computer program

Description automatically generated with medium confidence

6. Hadoop Map Reduce

Cette partie n’est pas réaliser par Achraf ABOUALAOUD

8. Analyse de données avec des outils de machine learning (R, …)

Vous devez ici décrire chaque étape de la mise en œuvre de l’activité autour des outils de visualisation de données (voir le sujet du projet).

9. Conclusion générale

10. Références et Bibliographie

|  |  |
| --- | --- |
| [1] | «stack overflow,» [En ligne]. Available: https://stackoverflow.com/. |
| [2] | «Oracle,» [En ligne]. Available: https://docs.oracle.com/en/. |
| [3] | «MongoDB,» [En ligne]. Available: https://www.mongodb.com/docs. |
| [4] | «hevodata,» [En ligne]. Available: https://hevodata.com/learn/mongoimport/. |

11. Annexes

11.1 Vidéo de présentation

Partie de l’architecture :

<https://drive.google.com/file/d/10X0XHqE0XeFX2XArY3M2J7AmdiKv_Lj5/view?usp=share_link>

11.2 Dossier contenant les scripts et programmes de construction du lac de données

Voir le dossier (Big\_data\_project\Groupe\_TPT\_21\Livrable\scripts)

11.3 Dossier contenant les scripts et programmes Hadoop Map Reduce

Cette partie n’est pas réaliser par Achraf ABOUALAOUD

11.4 Dossier contenant les scripts et programmes de visualisation de données

11.5 Dossier contenant les scripts et programmes d’analyse de données