

# Projet Big Data

Réalisé par le **Groupe 3** composé de :

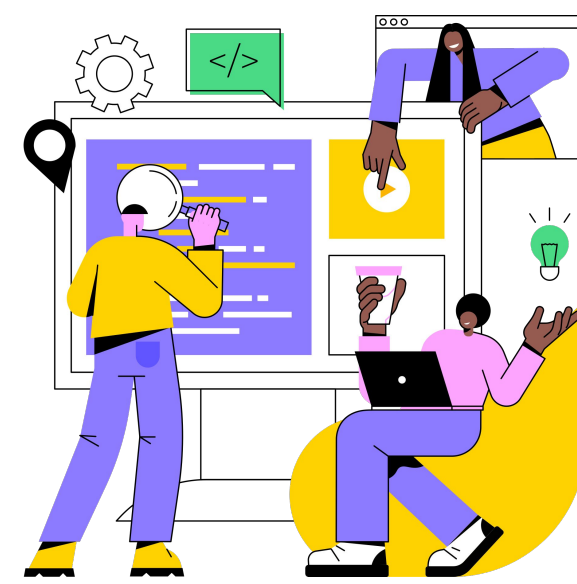
**Issam Harchi**  
**Roukayatou Omorou**  
**Nacer MESSAOUI**  
**Saba AZIRI**



[https://github.com/Rouk-Peace/ProjetDigi\\_Hadoop](https://github.com/Rouk-Peace/ProjetDigi_Hadoop)

# Contexte

Equipe data ingénieur(e)s/analystes



Fromagerie Digi



Infos clients, produits commandés, points de fidélité, ...

Besoins: améliorer ses stratégies de fidélisation client et de ciblage des offres

**Analyse du besoin client**  
**Etude des données**  
Source de données et format  
Volume de données  
Type de données

**Besoins techniques**

Hadoop HDFS HBase  
Power BI  
Python

Gestion de projet agile  
Répartition des tâches

**Partie HDFS**  
**MapReduce**  
Rouky & Issam

**Partie Hbase**  
**Power BI**  
Saba & Nacer

**Livrables**

**HDFS MapReduce:**  
Fichier excel des Top 10 clients en fonction des filtres  
PDF de la répartition des objets commandés / client

**HBase PowerBI**  
Visualisation de la fidélité des clients  
viz nombre d'objets commandés par année, avec top 5 objets  
Viz géographique du prog de fidélité

# Détails de la base de données Nosql

## Base principal

Nbre lignes : 135 277

Nbre colonnes : 25

## Base test

Nbre lignes : 1000

Nbre colonnes : 25

## Fichier csv:

séparateur = ,

guillemets = "

format de date = YYYY-MM-DD

HH:MM:SS

Colonne	Description	Type
codcli	Code client	Numérique
genrecli	Genre du client	Texte
nomcli	Nom du client	Texte
prenomcli	Prénom du client	Texte
cpcli	Code postal du client	Numérique
villecli	Ville du client	Texte
codcde	Code de la commande	Numérique
datcde	Date de la commande	Datetime
timbrecli	Timbre du client	Numérique
timbrecde	Timbre de la commande	Numérique
Nbcolis	Nombre de colis	Numérique
cheqcli	Chèque du client	Numérique
barchive	Indicateur d'archivage	Numérique ou Booléen
bstock	Indicateur de stock	Numérique ou Booléen
codobj	Code de l'objet	Numérique
qte	Quantité	Numérique
Colis	Colis	Texte
libobj	Libellé de l'objet	Texte
Tailleobj	Taille de l'objet	Texte
Poidsobj	Poids de l'objet	Numérique
points	Points	Numérique
indispobj	Disponibilité de l'objet	Numérique ou Booléen
libcondit	Libellé de la condition	Texte
prixcond	Prix de la condition	Numérique
puobj	Prix unitaire de l'objet	Numérique

## 11 colonnes avec valeurs nulles

Colonne	Valeurs nulles
genrecli	725
prenomcli	324
datcde	2
timbrecli	4
timbrecde	9
Nbcolis	8
cheqcli	43
qte	3
Colis	8
Tailleobj	86719
points	262

# HDFS MapReduce

## Objectif

Mettre en place une solution de traitement de données basée sur Hadoop HDFS, capable de filtrer et analyser les commandes clients sur une période donnée, afin d'identifier les 10 clients les plus fidèles. Cette analyse permet de visualiser la répartition des produits commandés par ces clients et d'exporter les résultats pour une prise de décision data driven.

### Mapper :

- Traite les données (format csv, Null, type de donnée, format de date)

- Filtre les données selon les critères :années (2008 -2012), départements (53, 61,75 et 28)

- Envoie les lignes (clé/valeur) au Reducer.

### Reducer :

- Calcule la fidélité des clients

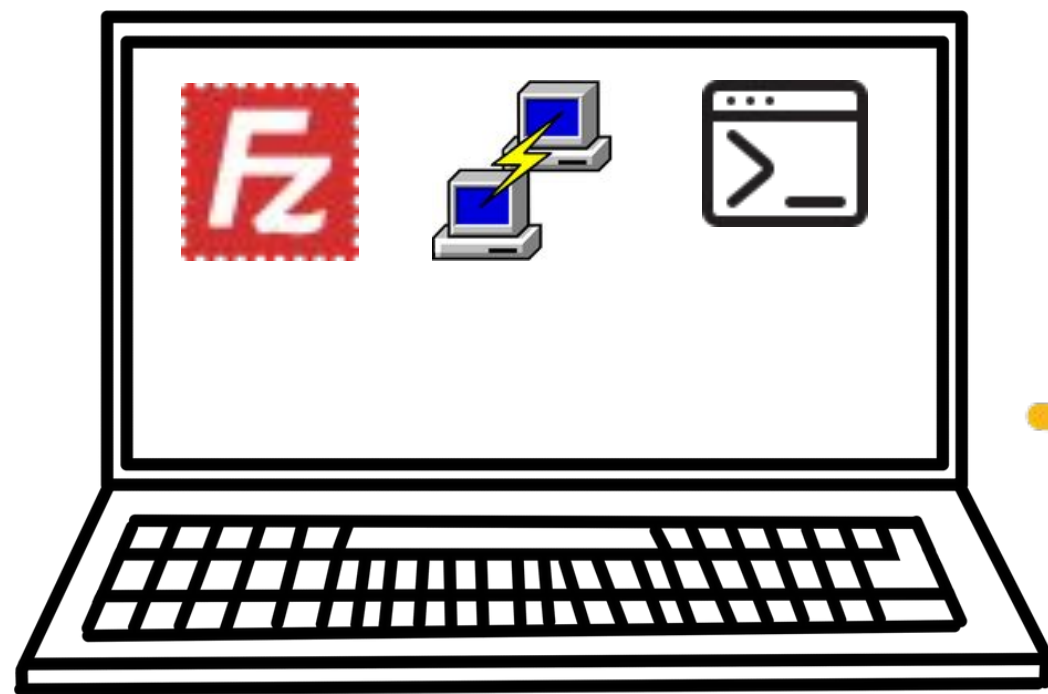
- Trie pour trouver les 10 plus fidèles.

- Crée les graphes de répartition des produits commandés/client

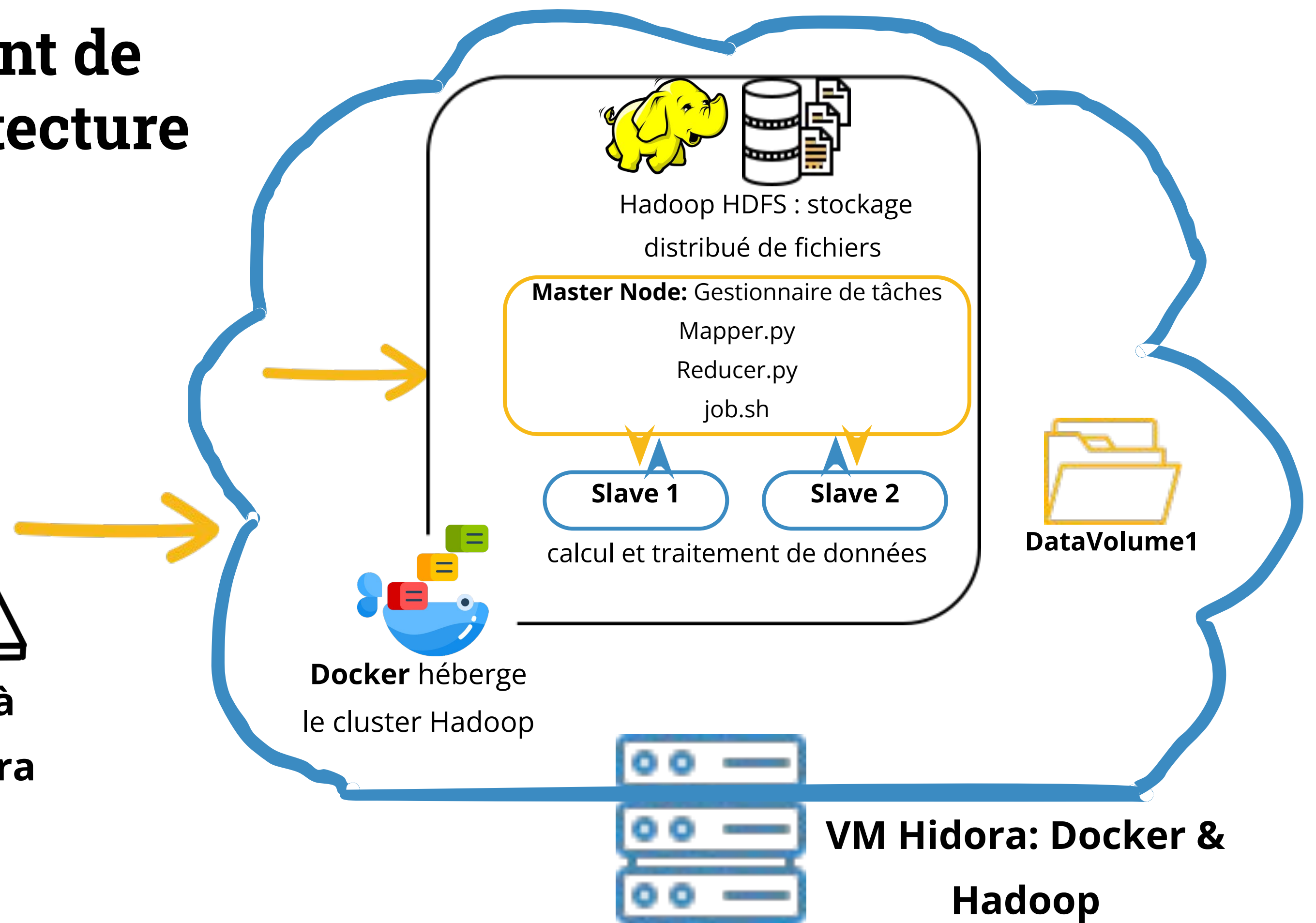
### job.sh

- Hadoopjar streaming

# Environnement de travail et architecture



Interface de connexion à distance avec la VM Hidora



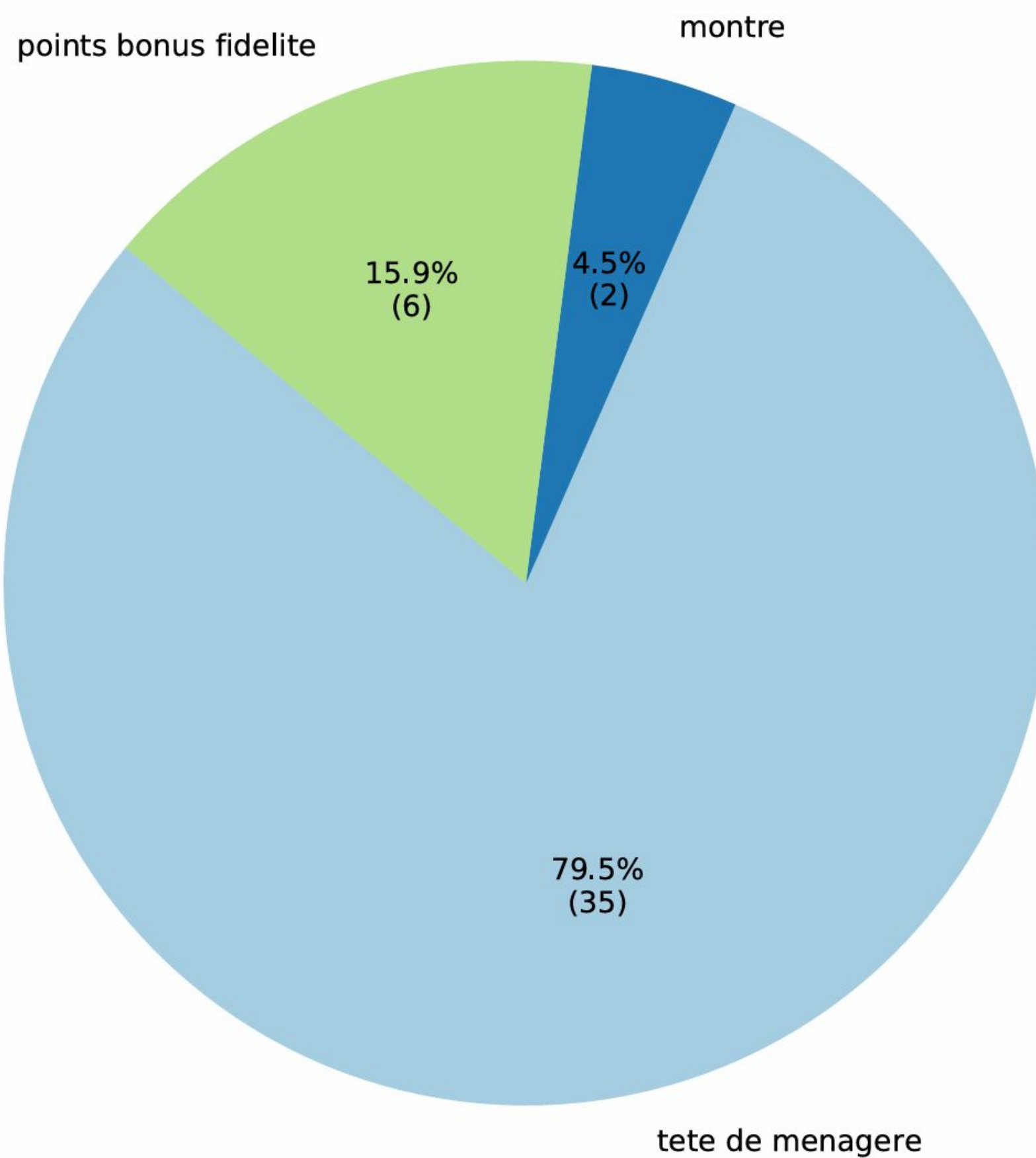
# Démo

1. Ouvrir Putty et filezilla et se connecter avec ses identifiants env à la VM
2. Copier le dossier job\_projetfromagerie( dataw\_fro03.csv, job\_projet.sh, mapper.py, reducer.py) dans filezilla
3. Dans le terminal putty, **faire ls** pour vérifier si le dossier à été bien copié dans la VM
4. **./start\_docker\_digi.sh**
5. **cd jobprojetfromagerie** pour se déplacer dan le repertoire
6. **ls** pour verifier les fichiers du repertoire
7. **docker cp job\_projet.sh hadoop-master:/root**
8. **docker cp mapper\_projet.sh hadoop-master:/root**
9. **docker cp reducer\_projet.sh hadoop-master:/root**
10. **docker cp dataw\_fro03.csv hadoop-master:/root**
11. **docker cp dataw\_fro03\_mini\_1000.csv hadoop-master:/root**
12. **cd ..** pour sortir du repertoire
13. **./lance\_srv\_slaves.sh**
14. **./bash\_hadoop\_master.sh**
15. **ls**
16. **chmod 777 job\_projet.sh**
17. **ls**
18. **./job\_projet.sh**
19. Ensuite lancer le service web sur le port 9070 (**<http://node182955-env-1839015-etudiant-l10.sh1.hidora.com:11529/>**)
20. Lancer le web yarn sur le port 8088 (**<http://node182955-env-1839015-etudiant-l10.sh1.hidora.com:11531/>**)
21. Allez sur **filezilla** pour vérifier l'exportation des fichiers : **/var/lib/docker/volumes/digi01/\_data**



Nom	Prénom	Département	Ville	Objet	Quantité	Fidélité totale
verrier	paulette	61	athis val de rouvre	tete de menagere	35	3800
verrier	paulette	61	athis val de rouvre	montre	2	
verrier	paulette	61	athis val de rouvre	points bonus fidelite	7	
verrier	paulette	61	athis val de rouvre	carte publicitaire	5	
verrier	paulette	61	athis val de rouvre	flyer	1	
dallet	nathalie	61	st germain du corbeis	serviette + gant	3	3220
dallet	nathalie	61	st germain du corbeis	points bonus fidelite	6	
dallet	nathalie	61	st germain du corbeis	carte publicitaire	4	
dallet	nathalie	61	st germain du corbeis	cle usb	1	
dallet	nathalie	61	st germain du corbeis	points flyer	1	
dallet	nathalie	61	st germain du corbeis	collecteur	10	
dallet	nathalie	61	st germain du corbeis	montre	9	
dallet	nathalie	61	st germain du corbeis	drap de bain	3	
dallet	nathalie	61	st germain du corbeis	flyer	1	
dallet	nathalie	61	st germain du corbeis	t-shirt blanc	20	
dallet	nathalie	61	st germain du corbeis	convertisseur	3	
jousseau	claud	53	crennes sur fraubee	tete de menagere	18	3110
jousseau	claud	53	crennes sur fraubee	points bonus fidelite	6	
jousseau	claud	53	crennes sur fraubee	carte publicitaire	4	
jousseau	claud	53	crennes sur fraubee	points flyer	1	
jousseau	claud	53	crennes sur fraubee	pelle a tarte	5	
jousseau	claud	53	crennes sur fraubee	couver	4	

## Répartition des objets commandés pour Verrier Paulette





# Hadoop & Hbase (Étapes)

## 1. Préparation et Connexion

└─  ssh -p 11453


root@node182946-env-1839015-etudiant-l01.sh1.hidora.com (Saba)

└─  ls

└─  cat ./start\_docker\_digi.sh

## 2. Configuration des Conteneurs Docker

└─  ./start\_docker\_digi.sh

└─  ./lance\_srv\_slaves.sh

└─  ./bash\_hadoop\_master.sh

└─  ./start-hadoop.sh

## 3. Configuration et Vérification des Services Hadoop

└─  Obtenir URL API Web Hadoop :

http://node182946-env-1839015-etudiant-l01.sh1.hidora.com:11452/  
(Saba)

└─  Vérifier Services : jps

## 4. Configuration du Service ODBC pour HBase

└─  ODBC Configuration

- Serveur : env-1839015-Etudiant-l09.sh1.hidora.com (Nacer)

- Port : 11521

- Utilisateur : root

- Mot de passe : (Aucun)

└─  URL Public pour ODBC :

http://env-1839015-etudiant-l09.sh1.hidora.com:9091 (Nacer)

## 5. Configuration et Lancement des Services Thrift pour HBase

└─ Lancer les Services Thrift : ./services\_hbase\_thrift.sh

└─ Vérifier les Services Thrift : jps

## 6. Gestion des Tables HBase

└─  hbase shell


└─  create 'BigFromagerie', 'cf'

└─  list

└─  describe 'BigFromagerie'

└─  put 'BigFromagerie', '1', 'cf:codcli', '001'

└─  scan 'BigFromagerie'

└─  disable 'BigFromagerie'

└─  drop 'BigFromagerie'


## 7. Importer les Données

└─  docker cp ./dataw\_fro03.csv

hadoop-master:/root/BigFromagerie.csv`

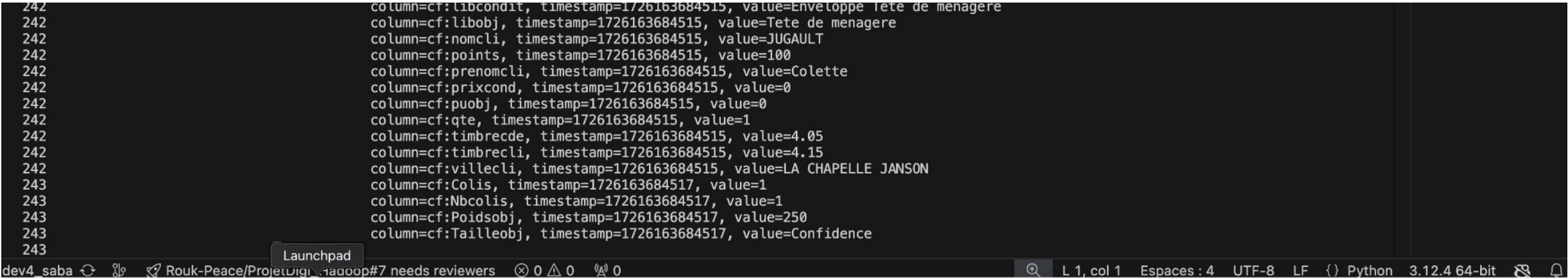
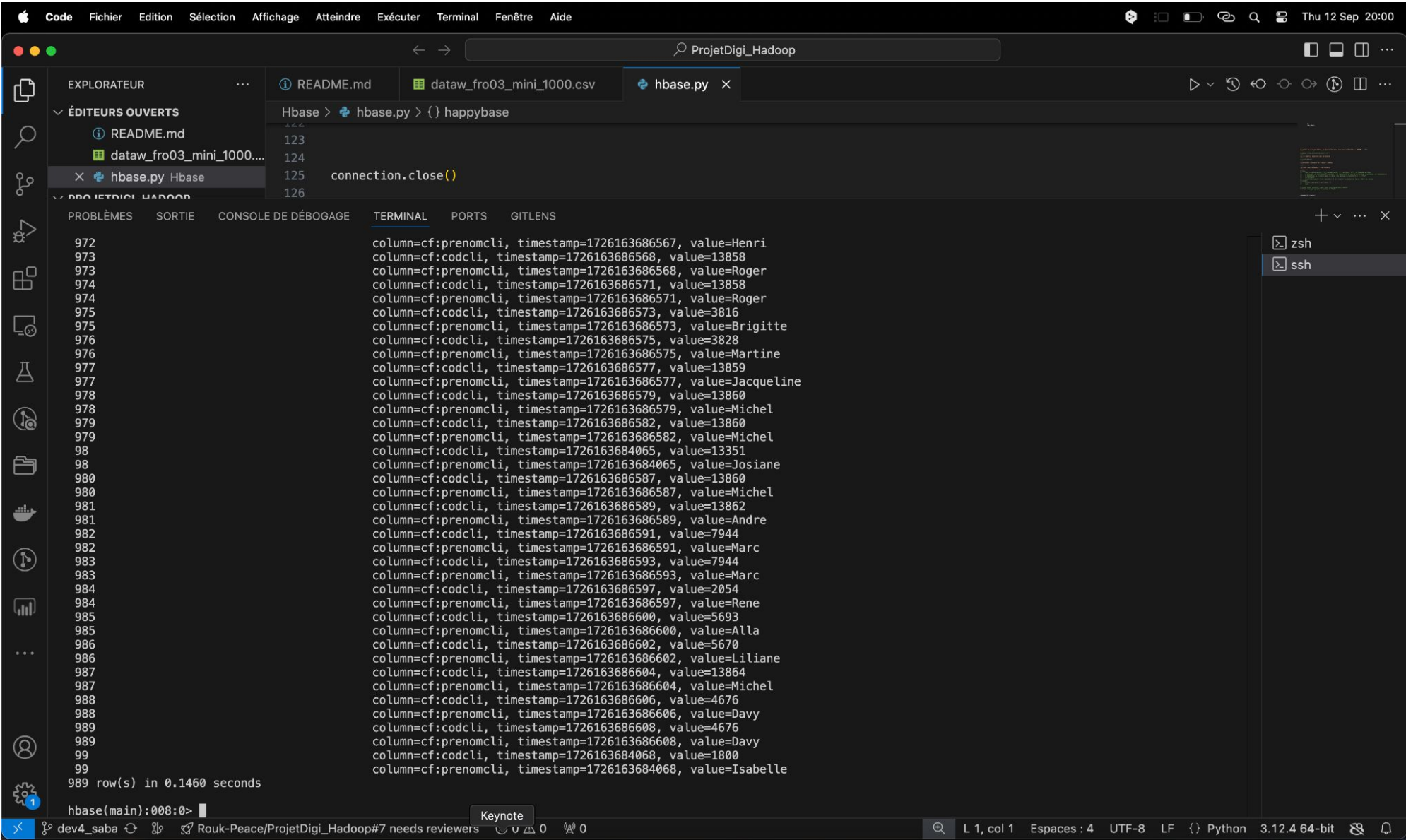
└─  ./bash\_hadoop\_master.sh`

└─  Exécuter le Script Python : python3 hbase\_finale.py

└─  hdfs dfs -put /root/BigFromagerie.csv

/user/root/input

# Hadoop & Hbase (cmd)

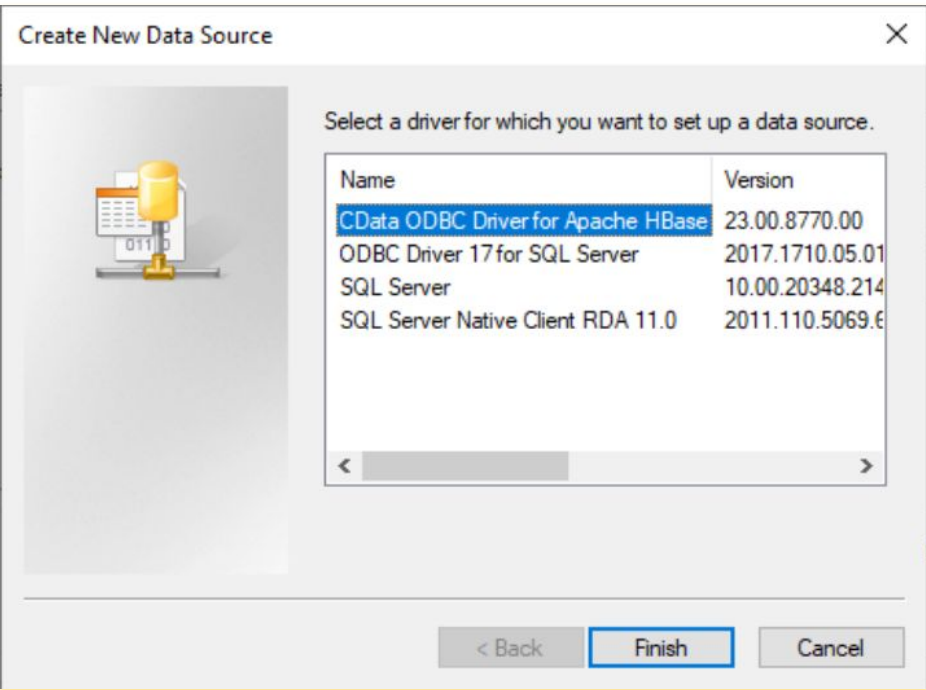




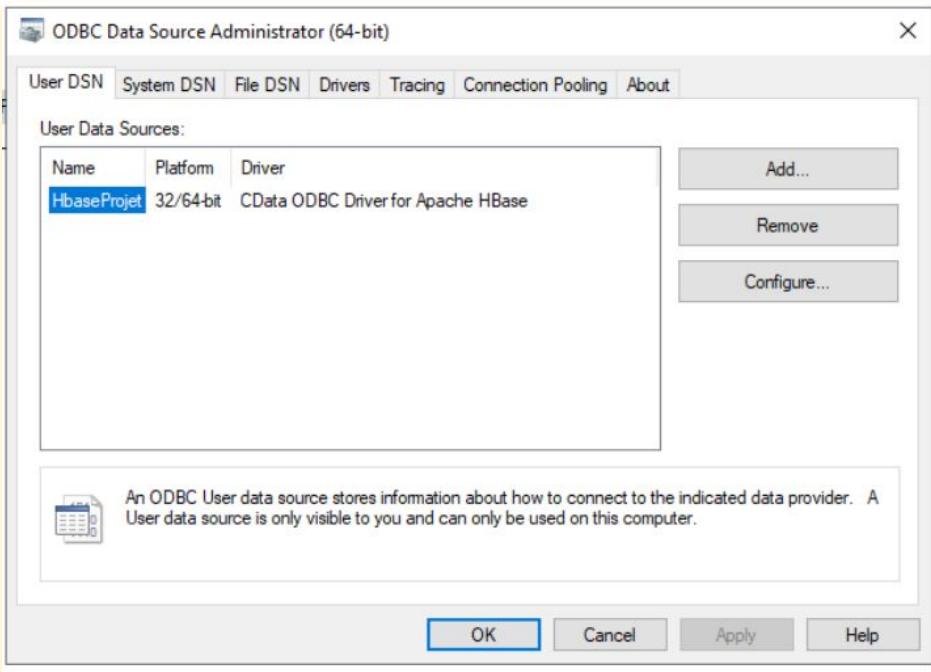
# Hbase & Odbc

**ODBC est un connecteur permettre PowerBI et HBASE de communiquer**  
**Pour accéder les données stockées dans la table Hbase**

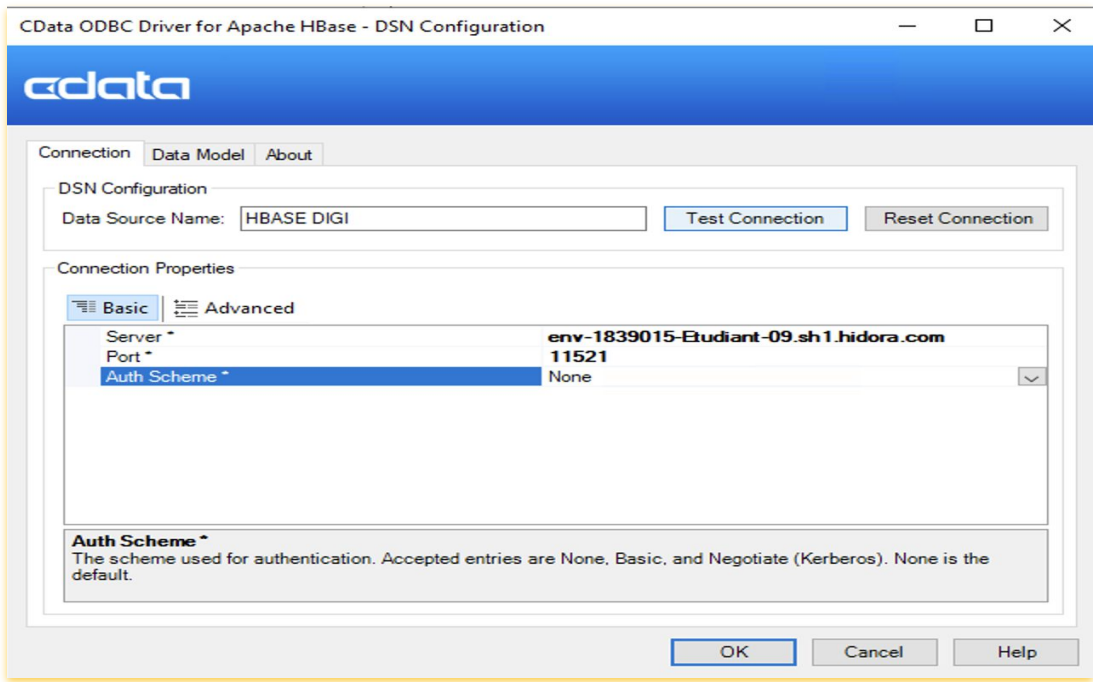
2.Sélectionner Cdata  
OBDC Driver



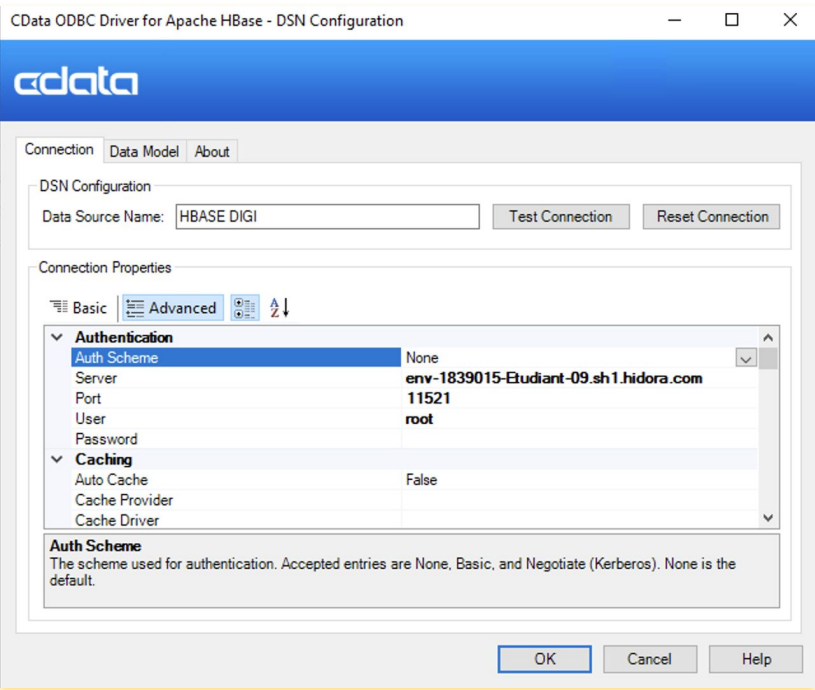
3. Procéder à la configuration  
HbaseProjet



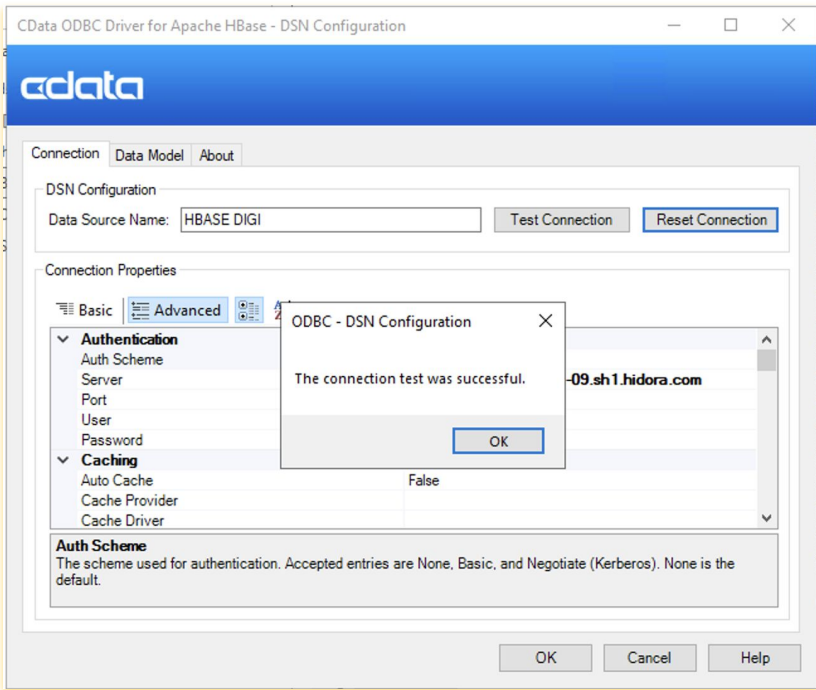
4. Configuration Dsn Basic



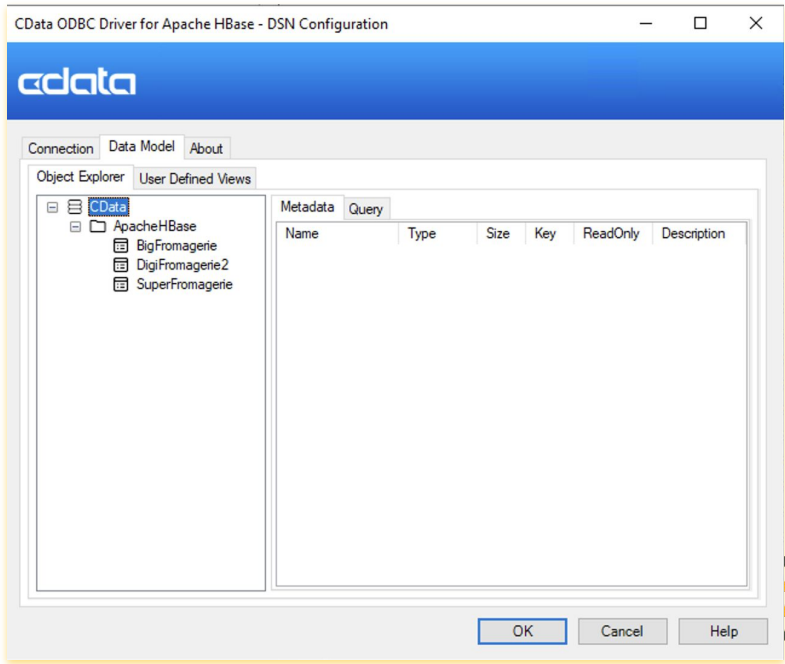
4. Configurer Dsn Basic



5. Configurer Data Model : HBASE DIGI



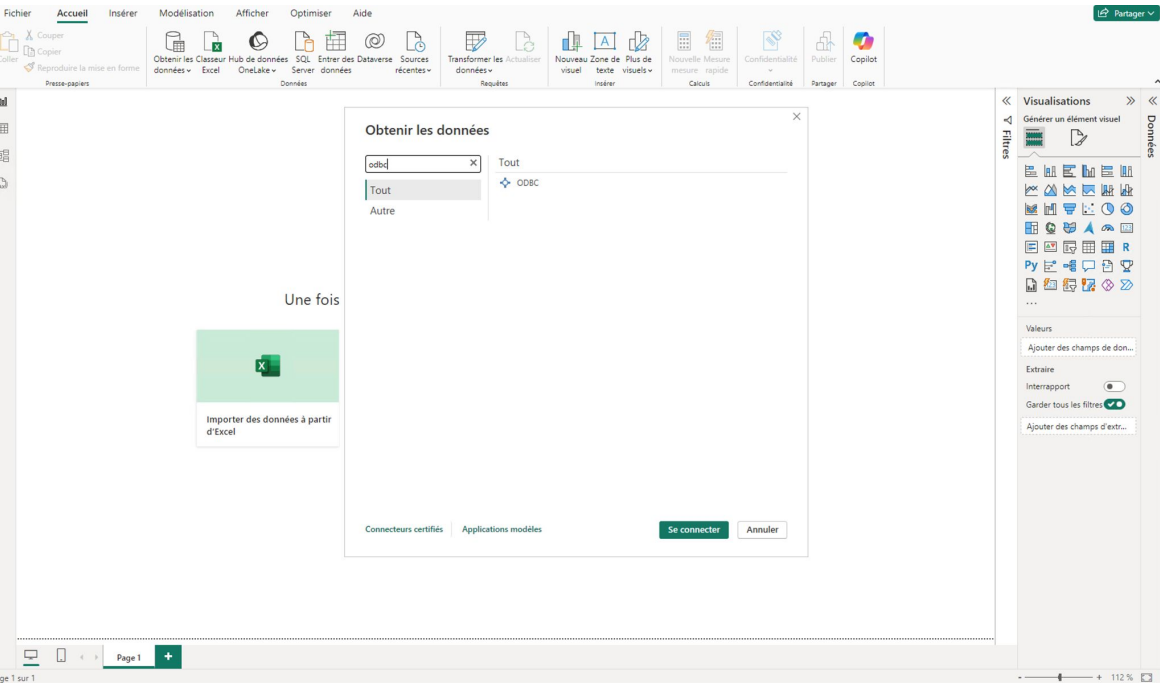
6. Vérification des tables



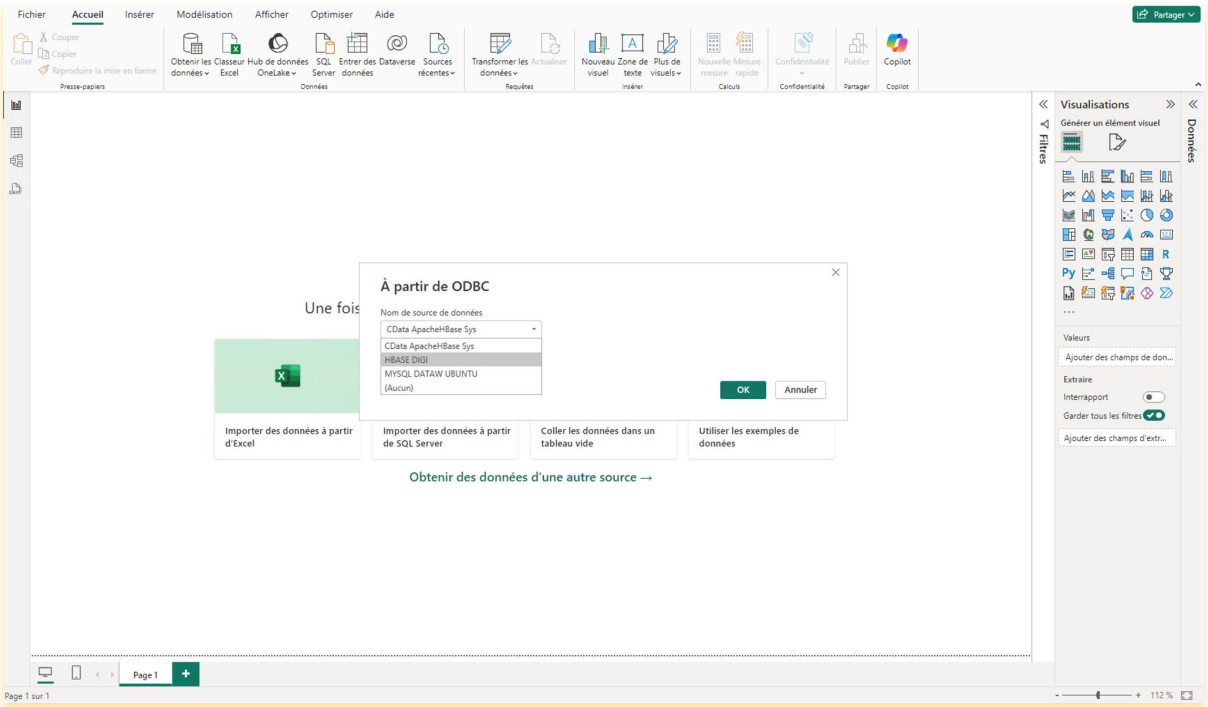
# Hbase et Power BI (driver ODBC)

En ouvrant PowerBI, nous allons exploiter les données dans la table HBASE en utilisant l'API ODBC.

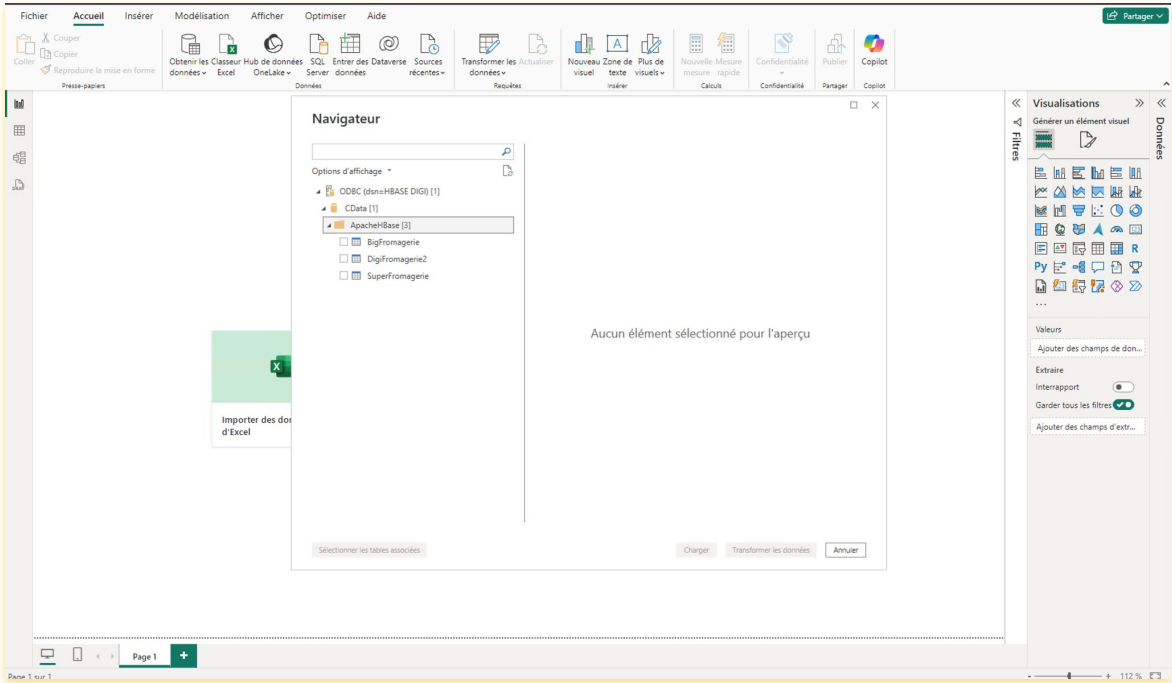
1.Ouvrir Power BI  
Obtenir les données  
Saisir ODBC dans la barre de recherche



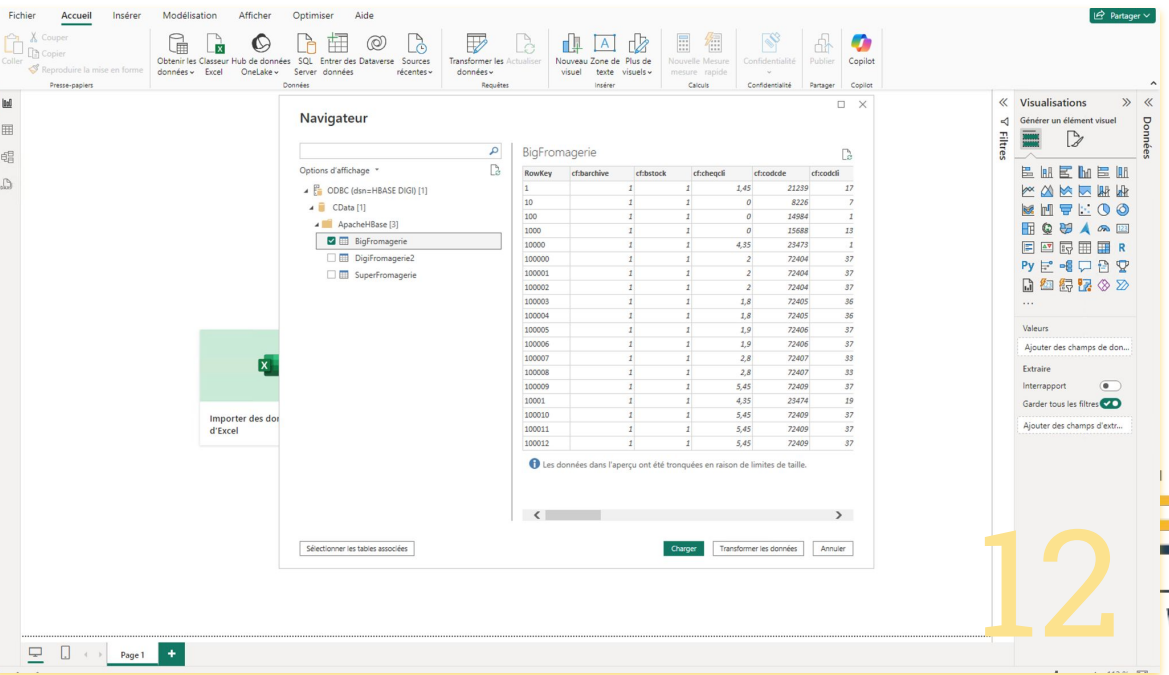
2.Sélectionner HBASE DIGI



3.Les bases de données Hbase  
s'affichent



4.Sélectionner la base que vous voulez  
charger et cliquer sur Transformer les  
données





# Power BI : Nettoyage données

- Renommer et structurer les données
- Renommer les colonnes et les tables
  - Réorganiser les colonnes
- Supprimer les colonnes ou les lignes inutiles
  - Gestion des doublons
- Modifier les erreurs : valeurs aberrantes (Date)
- Remplacer les valeurs négatifs par 0 (Points)
  - Ajouter une table Date et Département
  - Créer les tables de faits et dimensions
    - Faire le modèle en flocon

Nombre de lignes après nettoyage : 134 821

The screenshot shows the Power BI Query Editor interface. The formula bar contains the DAX formula: `= Table.SelectRows("#Type modifié3", each [CodeClient] = 8695)`. Below the formula bar, a table is displayed with columns: `CodeClient`, `CodeDepartement`, and `Client`. The table has two rows of data.

	CodeClient	CodeDepartement	Client
1	8695	35	Paulette Cannecu
2	8695	35	Paulette Canneau

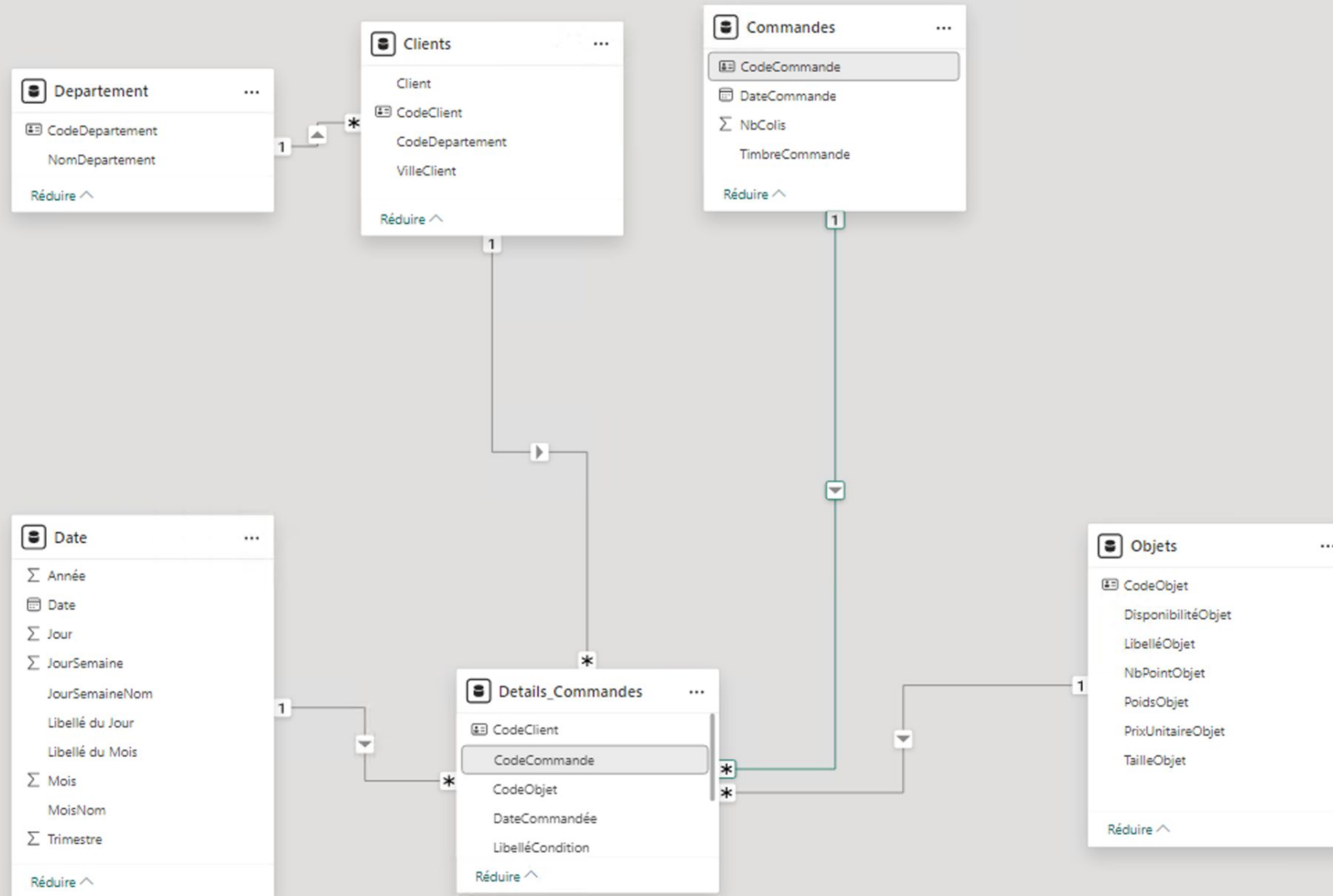
The screenshot shows a data table in the Power BI Query Editor. The table has columns: `Stock`, `CodeCommande`, `CodeClient`, `DateCommande`, `NbColis`, and `TimbreCommande`. The table contains 7 rows of data.

Stock	CodeCommande	CodeClient	DateCommande	NbColis	TimbreCommande
1	69938	3490	13/11/1015	1	8
1	80746	39073	30/12/1899	1	2,75
1	87671	40679	30/12/1899	1	2,1
1	53937	1247	30/12/1899	1	4,6
1	2990	2847	17/01/2005	1	4
1	22030	6734	25/02/2005	1	1,4
1	4504	4194	03/03/2005	1	6,4

The screenshot shows the Power BI Query Editor interface. The formula bar contains the DAX formula: `= Table.ReplaceValue("#Texte en majuscules", "ST ", "SAINT ", Replacer.ReplaceText, {"Ville"})`. Below the formula bar, a table is displayed with columns: `CodePostal`, `CodeDepartement`, and `Ville`. The table contains 17 rows of data.

	CodePostal	CodeDepartement	Ville
1	1120		1 PIZAY
2	1430		1 CONDAMINE
3	1474		1 PG OOSTHUIZEN
4	2000		2 CHAVIGNON
5	2000		2 LAON
6	2100		2 NEUVILLE SAINT AMAND
7	2100		2 SAINT QUENTIN
8	2100		2 SAINT QUENTIN
9	2110		2 BEAUREVOIR
10	2110		2 BECQUIGNY
11	2110		2 PREMONT
12	2120		2 GUISE
13	2120		2 LANDIFAY ET BERTAIGNEMO...
14	2130		2 FERRE EN TARDENOIS
15	2130		2 VILLERS SUR FERRE
16	2140		2 VERVINS
17	2190		2 AMIFONTAINE

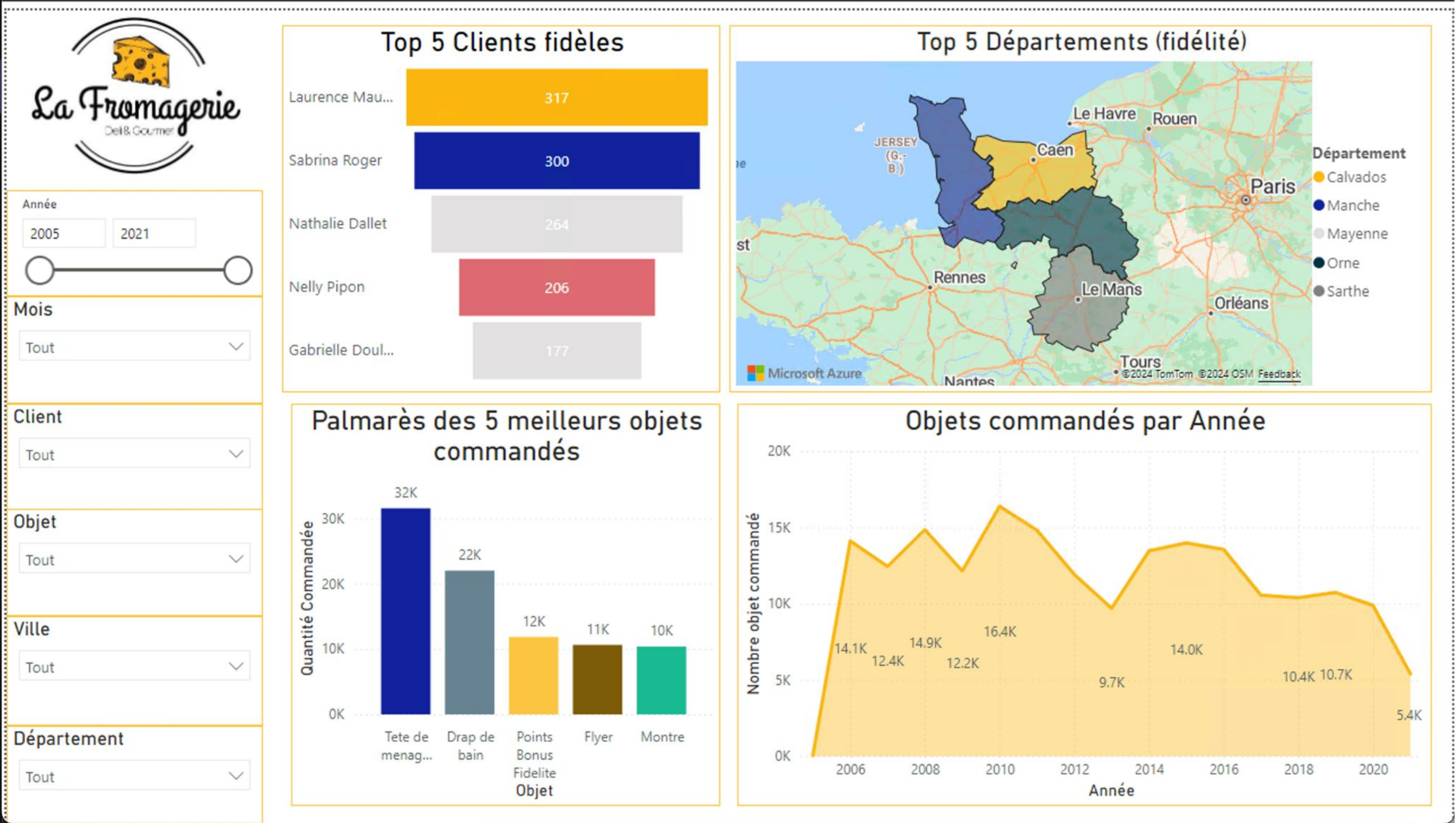
# Power BI : Vue du modèle





# Power BI : Dashboard

- Voir la fidélité des clients sur selon l'intervalle de dates,
- Voir le nombre d'objets commandés par année et les 5 meilleurs (Palmarès),
- Voir les départements les plus représentatifs du programme de fidélité de la fromagerie (Palmarès).



# CONCLUSION

Le projet de traitement des données de la fromagerie à l'aide de Hadoop a permis de mettre en place un processus efficace de traitement de données massives via un workflow de type MapReduce.

Ce processus a non seulement permis de nettoyer et filtrer les données, mais aussi d'extraire des informations pertinentes sur la fidélité des clients, les objets commandés, et les départements les plus actifs. Les résultats obtenus offrent une vue claire de la relation entre les commandes et les clients, ainsi que des indicateurs importants pour le suivi des activités de fidélité.

## Perspectives d'amélioration

- **Connexion à une base de données NoSQL** : Une prochaine étape serait d'intégrer HBase pour un stockage persistant et plus rapide des données, facilitant l'accès direct aux informations extraites.
- **Automatisation des Jobs Hadoop** : Programmer les jobs Hadoop à intervalles réguliers, ou sur événement, permettrait de traiter les nouvelles données de manière continue, sans intervention manuelle.
- **Mise en place de Data Streaming** : Pour améliorer encore l'efficacité et la rapidité du traitement des données, la mise en œuvre du streaming avec Apache Kafka ou Flink pourrait être envisagée. Cela permettrait de traiter les données en temps réel et d'alimenter des tableaux de bord dynamiques avec Power BI ou ELK.
- **Amélioration du tableau de bord** : Enrichir les visualisations des données avec Power BI ou d'autres outils de reporting pourrait offrir une vue plus interactive et détaillée sur l'évolution des comportements clients et des ventes.