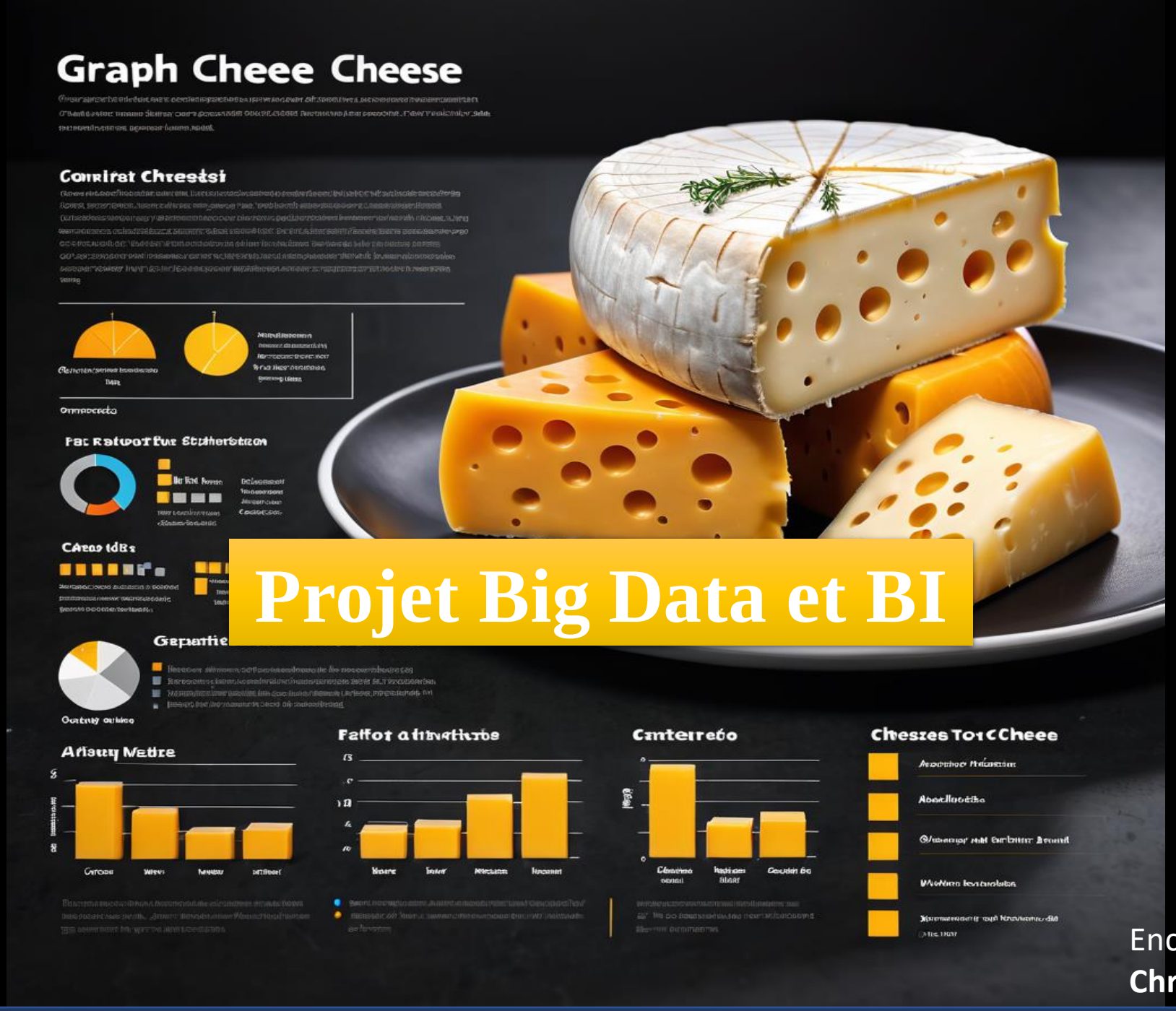
Diginamic Formation

« Dynamisez votre carrière digitale ! »



Projet Big Data et BI

Présenté par:

Camille Hamel

Razan Altuzzar

Randolf Hackman

Nene Aissatou Bah

Luis-Miguel Borderon

Encadré par:

Christophe Germain

Introduction

Notre client (fromagerie) dispose d'un entrepôt de données depuis 2004, représenté par un fichier CSV. Notre mission est de répondre à ses besoins en mettant en place des processus de traitement de données.

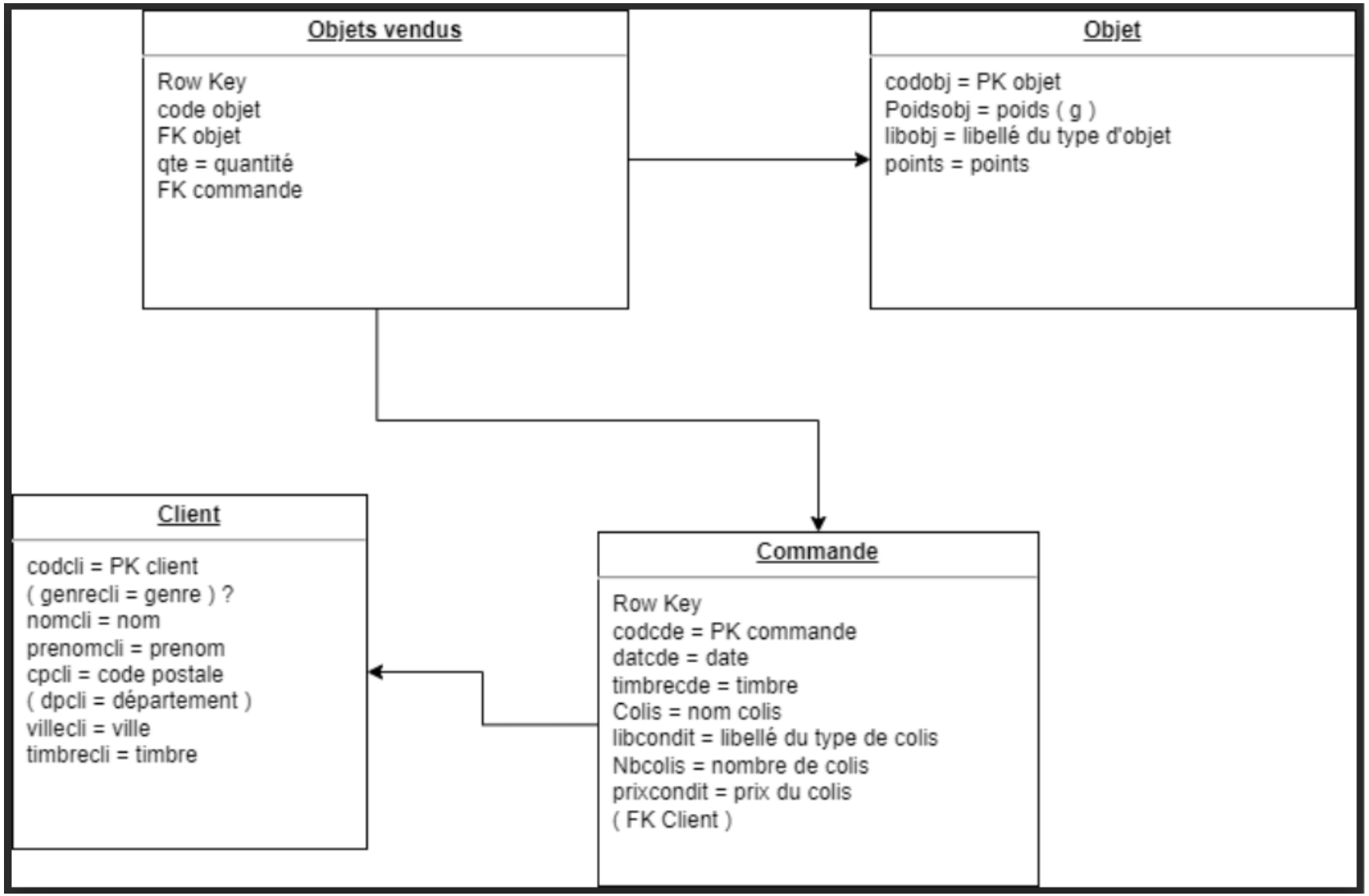
- **Objectif:** Mettre en œuvre un ensemble d'applications Big Data et Power BI

  - **Lot 1** : Filtrer et analyser les données de 2006 à 2010 pour les départements 53, 61 et 28. Identifier les 100 meilleures commandes et exporter les résultats dans un fichier Excel.

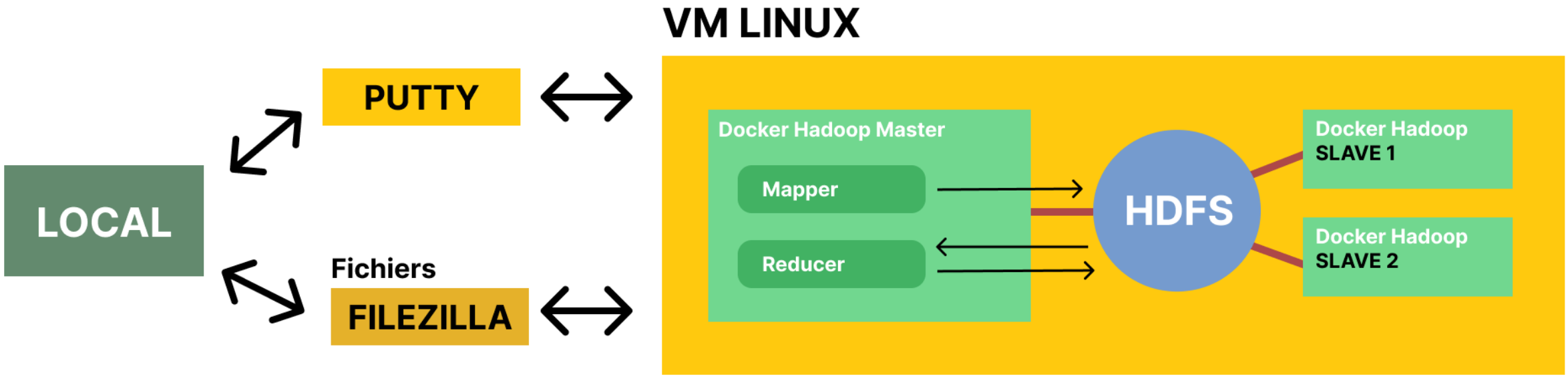
  - **Lot 2** : Répéter le processus pour les années 2011 à 2016, avec les départements 22, 49 et 53. Sélectionner aléatoirement 5% des 100 meilleures commandes, produire un fichier Excel et un document PDF avec un graphique (PIE) par ville.

  - **Lot 3** : Mettre en place une base de données NoSQL HBASE, un moteur de recherche avec Power BI et créer un tableau de bord interactif pour visualiser les résultats des Lots 1 et 2.

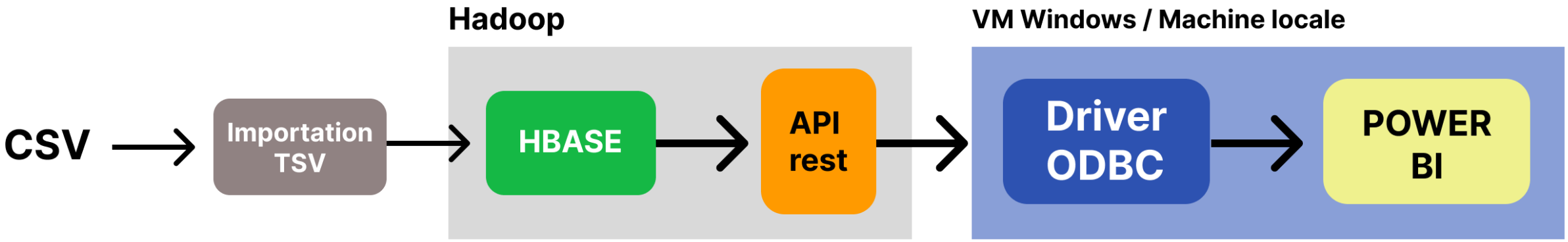
Diagramme UML



Architecture technique HADOOP

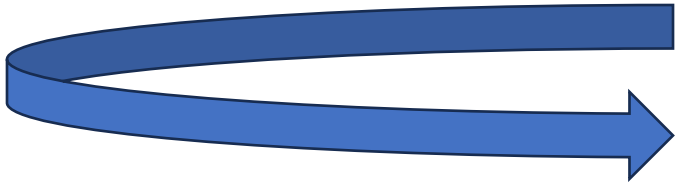


Architecture technique Hbase (vers Power BI)



Valeurs manquantes et variables cibles

```
Tailleobj    86719
genrecli       725
prenomcli      324
points         262
cheqcli         43
timbrecde        9
Colis            8
Nbcolis          8
timbrecli        4
qte              3
datcde           2
dtype: int64
```



Traiter la base sans suppression des na

Création des variables et mise en forme

Types

```
int64      10
object      8
float64     7
dtype: int64
```

Int type

	count	mean	std	min	25%	50%	75%	max
codcli	135277.0	19724.126097	12335.239240	27.0	8068.0	19921.0	30915.0	41291.0
cpcli	135277.0	48183.938807	20452.967226	1120.0	28800.0	50690.0	61350.0	95870.0
codcde	135277.0	56835.146832	20267.018358	478.0	42030.0	58649.0	72837.0	90048.0
barchive	135277.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0
bstock	135277.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0
codobj	135277.0	68.614391	34.233734	20.0	42.0	65.0	84.0	168.0
Poidsobj	135277.0	150.369139	247.473465	0.0	0.0	60.0	250.0	9300.0
indispobj	135277.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
prixcond	135277.0	51.438094	112.704303	0.0	0.0	0.0	34.0	655.0
puobj	135277.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0

Construction du mapper

1- Création de la variable année

```
data['datcde'] = pd.to_datetime(data['datcde'], errors='coerce')
data['annee'] = data['datcde'].dt.year
```

2- Filtrage selon le département

```
df_mapper_result = data[(data['annee']>=2006) & (data['annee']<=2010) & data['cpcli'].isin(["53","61","28"])]
```

3- Selection des variables selon l'objectif

```
print(df_mapper_result.iloc[ind,8],",",df_mapper_result.iloc[ind,7],',',df_mapper_result.iloc[ind,6],',',df_m
```

Construction du reducer

1- Regroupement des quantités demandées pour chaque commande :

```
data=pd.DataFrame(columns=['codcde','villecli','cpcli','qte','timbrecde'])
```

```
cmd_sum = data.groupby(["codcde"])["qte"].sum()
s=cmd_sum.sort_values(ascending=False)
```

2- Intégration des informations demandées :

```
s value :
 codcde
25862     42
42997     33
25861     30
24701     25
26620     20
          ..
```

```
df_merge = pd.merge(s, data, on='codcde', how='left')
```

Résultats

	codcde	villecli	cpcli	timbrecde	qte_commandes
0	25862	ARROU	28	9.85	42
4	42997	SAINT GEORGES DES GROSEILLERS	61	6.4	33
14	25861	ARROU	28	7.85	30
19	24701	TOURNAI SUR DIVE	61	5.1	25
20	26620	BOUCE	61	5.8	20
..	...	...	...	...	...
293	27431	ATHIS VAL DE ROUVRE	61	6.4	9
294	48729	THUBOEUF	53	4.6	9
296	39106	SAINT PIERRE D'ENTREMONT	61	6.4	9
300	45743	ECOUCHE LES VALLEES	61	5.9	9
304	45727	TINCHEBRAY	61	6.5	9

LOT 2 - Objectifs

- Filtrer sur les années 2011 à 2016 et les départements 22, 49 et 53
- Ne garder que les commandes avec un timbrecli à 0
- Obtenir un extract des 100 meilleures commandes avec la somme et la moyenne des quantités d'articles commandées
- Échantilloner ces commandes en en prenant 5%
- Faire un extract excel
- Réaliser un graphique "camembert" en pdf présentant la répartition par ville

LOT 2 – Solution Technique

- Solution mapreduce sur Hadoop
- Mapper :
  - simple lecture de csv avec filtre par colonne
- Reducer :
  - Pandas pour les agrégations, calculs et export excel
  - Matplotlib pour le graphique camembert

- Lecture CSV simple

- Filtre colonnes:
  - Cpcli (col 5)
  - Datecde (col 8)
  - Timbrecli (col 9)

- Export données filtrées en CSV :
  - Codcde
  - Villecli
  - qte
  - Timbrecli
  - Timbrecde

```
57914,RENNES EN GRENOUILLES,1,0,1.5
57914,RENNES EN GRENOUILLES,1,0,1.5
57920,CHEMAZE,1,0,5.6
57920,CHEMAZE,1,0,5.6
57922,JAVRON LES CHAPELLES,1,0,5.6
57922,JAVRON LES CHAPELLES,1,0,5.6
57922,JAVRON LES CHAPELLES,1,0,5.6
57927,MONTIGNE LE BRILLANT,1,0,4.6
57927,MONTIGNE LE BRILLANT,1,0,4.6
57931,OREE D ANJOU,1,0,2.5
57938,CHATEAU GONTIER SUR
MAYENNE,4,0,6.5
```

LOT 2 – Reducer 1/2

- Pandas
  - Lecture de l'entrée standard via read_csv
  - Nommer les colonnes pour une manipulation plus aisée
  - Créer un dataframe trié avec la colonne
  - somme_qte groupée par commande
  - Créer un 2e dataframe avec la colonne moyenne_qte groupée par commande
  - Fusionner les deux
  - Prendre le top 100
  - Sampler 5%
  - Faire un export excel simplement avec pandas.to_excel()

LOT 2 – Reducer 2/2

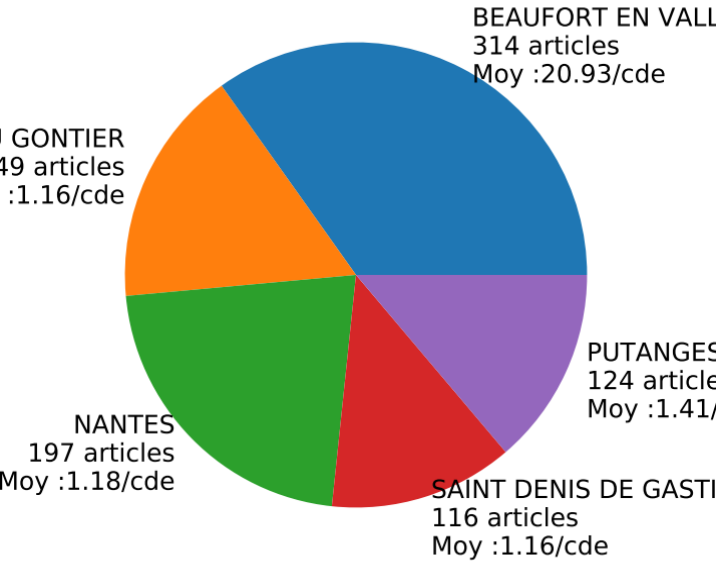
- Répétition des étapes précédentes en groupant par ville
- Export des données par ville
  - 1 liste de libellés
  - 1 liste de valeurs
- Utilisation de MatPlotLib pour la création d'un camembert
- Export PDF

- Exécution de la chaine mapreduce via hadoop streaming

- Récupération des résultats dans le /datavolume1/

```
root@hadoop-master:~# hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-stre
aming-2.7.2.jar -file /root/mapper_lot2_pourri.py -mapper "python3 mapper_lot2_p
ourri.py" -file /root/reducer_lot2.py -reducer "python3 reducer_lot2.py" -input
/user/root/datafromage.csv -output /user/root/sortiefromage
24/02/07 08:37:27 WARN streaming.StreamJob: -file option is deprecated, please u
se generic option -files instead.
packageJobJar: [/root/mapper_lot2_pourri.py, /root/reducer_lot2.py, /tmp/hadoop-
unjar7725706206068163615/] [] /tmp/streamjob2807494938223402713.jar tmpDir=null
24/02/07 08:37:28 INFO client.RMProxy: Connecting to ResourceManager at hadoop-m
aster/172.18.0.2:8032
24/02/07 08:37:28 INFO client.RMProxy: Connecting to ResourceManager at hadoop-m
aster/172.18.0.2:8032
24/02/07 08:37:28 INFO mapred.FileInputFormat: Total input paths to process : 1
24/02/07 08:37:28 INFO mapreduce.JobSubmitter: number of splits:2
24/02/07 08:37:29 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_17
07294768314_0001
24/02/07 08:37:29 INFO impl.YarnClientImpl: Submitted application application_17
07294768314_0001
24/02/07 08:37:29 INFO mapreduce.Job: The url to track the job: http://hadoop-ma
ster:8088/proxy/application_1707294768314_0001/
24/02/07 08:37:29 INFO mapreduce.Job: Running job: job_1707294768314_0001
24/02/07 08:37:35 INFO mapreduce.Job: Job job_1707294768314_0001 running in uber mode : false
24/02/07 08:37:35 INFO mapreduce.Job:  map 0% reduce 0%
24/02/07 08:37:41 INFO mapreduce.Job:  map 100% reduce 0%
24/02/07 08:37:47 INFO mapreduce.Job:  map 100% reduce 100%
24/02/07 08:37:47 INFO mapreduce.Job: Job job_1707294768314_0001 completed successfully
24/02/07 08:37:47 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=1728748
                FILE: Number of bytes written=3819562
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=26538154
                HDFS: Number of bytes written=3995
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=6953
                Total time spent by all reduces in occupied slots (ms)=3706
                Total time spent by all map tasks (ms)=6953
                Total time spent by all reduce tasks (ms)=3706
                Total vcore-milliseconds taken by all map tasks=6953
                Total vcore-milliseconds taken by all reduce tasks=3706
```

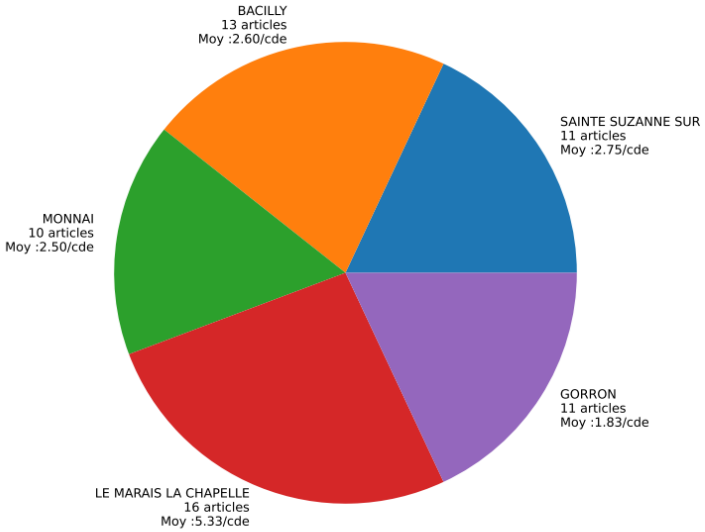
LOT 2 – Résultats



Matplotlib pie chart
VM

Sortie XLSX

	ville	sum_commandes	moy_commandes
12	BEAUFORT EN VALLEE	314	20,93333333
44	CHATEAU GONTIER	149	1,1640625
27	NANTES	197	1,179640719
74	SAINT DENIS DE GASTINES	116	1,16
66	PUTANGES LE LAC	124	1,409090909



Matplotlib pie chart
local

Procedure d’import csv (1/3)

- **Démarrer les dockers master & slave**

  ./start_docker_digi.sh

  ./lance_srv_slaves.sh

```
[root@node175820-env-1839015-etudiant1 ~]# ./start_docker_digi.sh
hadoop-master
hadoop-slave1
hadoop-slave2
[root@node175820-env-1839015-etudiant1 ~]# ./lance_srv_slaves.sh
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to
the list of known hosts.
hadoop-master: zookeeper running as process 4942. Stop it first.
running master, logging to /usr/local/hbase/logs/hbase--master-hadoop-slave1.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was rem
oved in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was
removed in 8.0
: running regionserver, logging to /usr/local/hbase/logs/hbase--regionserver-had
oop-slave1.out
: OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was r
emoved in 8.0
: OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support wa
s removed in 8.0
thrift running as process 601. Stop it first.
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to
the list of known hosts.
hadoop-master: zookeeper running as process 4942. Stop it first.
running master, logging to /usr/local/hbase/logs/hbase--master-hadoop-slave2.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was rem
oved in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was
removed in 8.0
: running regionserver, logging to /usr/local/hbase/logs/hbase--regionserver-had
oop-slave2.out
: OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was r
emoved in 8.0
: OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support wa
s removed in 8.0
thrift running as process 605. Stop it first.
```

- Ces commandes permettent de configurer et lancer un environnement Docker

Copier le csv depuis la VM vers le docker

docker cp ./dataw_fro03.csv hadoop-master:/root/datafromage.csv

- **Rentrer sur le docker hadoop master**
  - ./bash_hadoop_master.sh

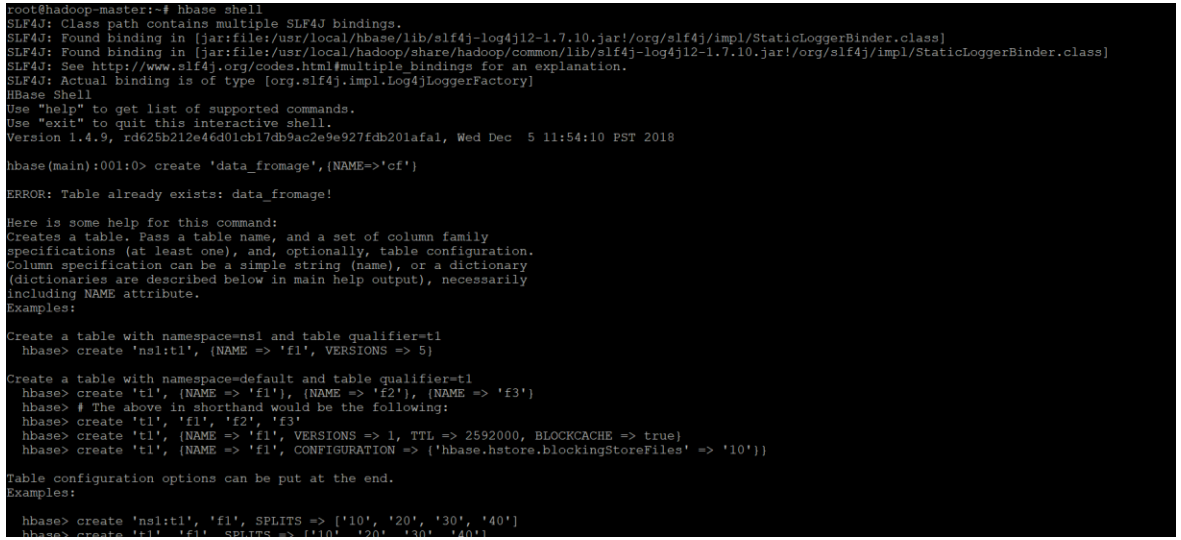
```
[root@node175820-env-1839015-etudiant1 ~]# docker cp ./dataw_fro03.csv hadoop-ma
ster:/root/datafromage.csv
Successfully copied 26.5MB to hadoop-master:/root/datafromage.csv
[root@node175820-env-1839015-etudiant1 ~]# ./bash_hadoop_master.sh
root@hadoop-master:~# ./start-hadoop.sh


Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to
the list of known hosts.
hadoop-master: namenode running as process 294. Stop it first.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to
the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to
the list of known hosts.
hadoop-slave2: datanode running as process 893. Stop it first.
hadoop-slave1: datanode running as process 894. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 512. Stop it first.


starting yarn daemons
resourcemanager running as process 557. Stop it first.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to
the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to
the list of known hosts.
hadoop-slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root
-nodemanager-hadoop-slave2.out
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root
-nodemanager-hadoop-slave1.out
```

- Ces commandes sont utilisées pour transférer des données vers le conteneur Docker, démarrer les services Hadoop, et exécuter des scripts

Procedure d’import csv (2/3)

•Lancer Happybase pour la creation de la table sur hbase (Commande utiliser)

- •hbase shell
- •create 'data_fromage',{NAME=>'cf'}
- •exit



•Préparer son fichier pour l'import

- Script pour ajouter le numéro de ligne au début de chaque ligne du csv
- awk -v OFS=',' '{print NR,$0}' /root/datafromage.csv > /root/datafromage_prepare.csv
- Envoyer le fichier sur le hdfs
  - hadoop fs -copyFromLocal /root/datafromage_prepare.csv /user/root/

Ces commandes préparent et copient le fichier CSV vers HDFS pour son utilisation dans un environnement Hadoop.

Procedure d’import csv (3/3)

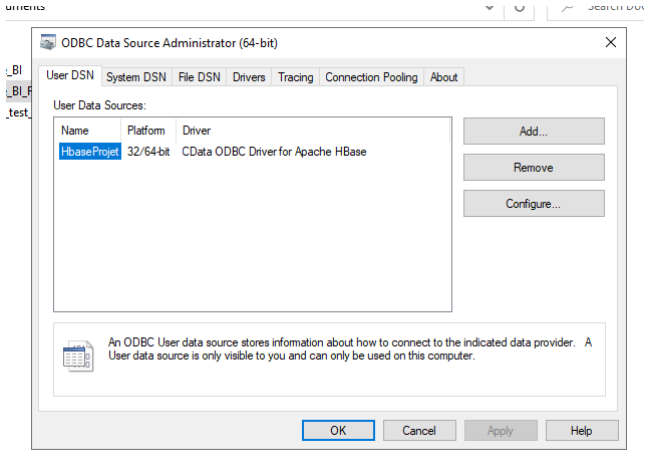
Importer son fichier sur hbase

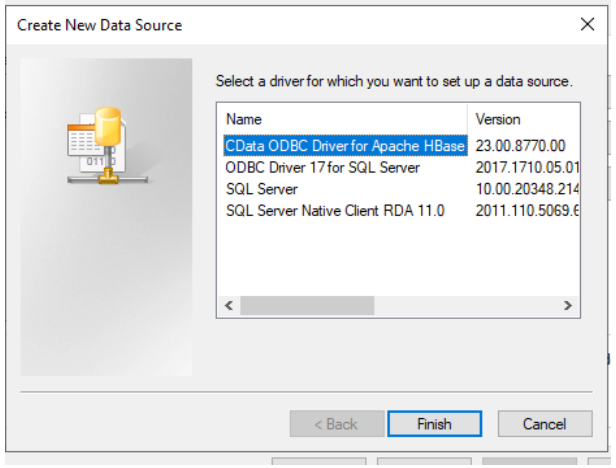
hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=',' -Dimporttsv.columns='HBASE_ROW_KEY,cf:codcli,cf:genrecli,cf:nomcli,cf:prenomcli,cf:cpcli,cf:villecli,cf:codcde,cf:datcde,cf:timbrecli,cf:timbrecde,cf:Nbcolis,cf:cheqcli,cf:barchive,cf:bstock,cf:codobj,cf:qte,cf:Colis,cf:libobj,cf:Tailleobj,cf:Poidsobj,cf:points,cf:indispobj,cf:libcondit,cf:prixcond,cf:puobj' data_fromage /user/root/datafromage_prepare.csv

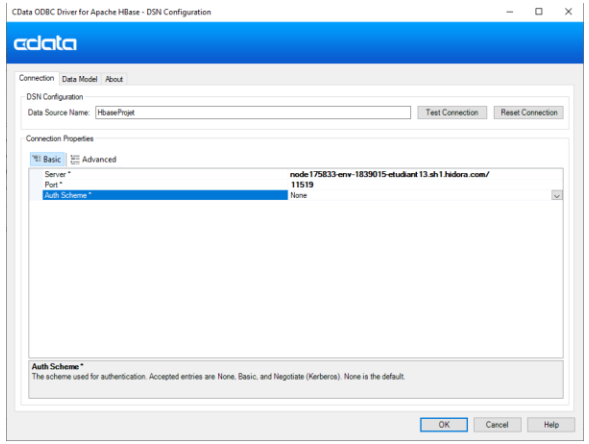
```
root@hadoop-master:~# hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimport
tsv.separator=',' -Dimporttsv.columns='HBASE_ROW_KEY,cf:codcli,cf:genrecli,cf:no
mcli,cf:prenomcli,cf:cpcli,cf:villecli,cf:codcde,cf:datcde,cf:timbrecli,cf:timbr
ecde,cf:Nbcolis,cf:cheqcli,cf:barchive,cf:bstock,cf:codobj,cf:qte,cf:Colis,cf:li
bobj,cf:Tailleobj,cf:Poidsobj,cf:points,cf:indispobj,cf:libcondit,cf:prixcond,cf
:puobj' data_fromage /user/root/datafromage_prepare.csv
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.10.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4
j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2024-02-07 08:52:06,898 INFO  [main] zookeeper.RecoverableZooKeeper: Process ide
ntifier=hconnection-0x22ff4249 connecting to ZooKeeper ensemble=hadoop-master:21
81
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: Client environment:zoo
keeper.version=3.4.10-39d3a4f269333c922ed3db283be479f9deacaa0f, built on 03/23/2
017 10:13 GMT
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: Client environment:hos
t.name=hadoop-master
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.version=1.8.0_292
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.vendor=Private Build
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.home=/usr/lib/jvm/java-8-openjdk-amd64/jre
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: .9.jar:/usr/local/hado
op/share/hadoop/mapreduce/lib/jackson-mapper-asl-1.9.13.jar:/usr/local/hadoop/sh
are/hadoop/mapreduce/lib/guice-servlet-3.0.jar:/usr/local/hadoop/share/hadoop/ma
preduce/lib/avro-1.7.4.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapre
duce-client-jobclient-2.7.2-tests.jar:/usr/local/hadoop/share/hadoop/mapreduce/h
adoop-mapreduce-client-app-2.7.2.jar:/usr/local/hadoop/share/hadoop/mapreduce/ha
doop-mapreduce-client-hs-plugins-2.7.2.jar:/usr/local/hadoop/share/hadoop/mapred
uce/hadoop-mapreduce-client-hs-2.7.2.jar:/usr/local/hadoop/share/hadoop/mapreduc
e/hadoop-mapreduce-examples-2.7.2.jar:/usr/local/hadoop/share/hadoop/mapreduce/h
adoop-mapreduce-client-shuffle-2.7.2.jar:/usr/local/hadoop/share/hadoop/mapreduc
e/hadoop-mapreduce-client-core-2.7.2.jar:/usr/local/hadoop/share/hadoop/mapreduc
e/hadoop-mapreduce-client-jobclient-2.7.2.jar:/usr/local/hadoop/share/hadoop/map
reduce/hadoop-mapreduce-client-common-2.7.2.jar:/usr/local/hadoop/contrib/capaci
ty-scheduler/*.jar
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.library.path=/usr/local/hadoop/lib/native
2024-02-07 08:52:06,903 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
```

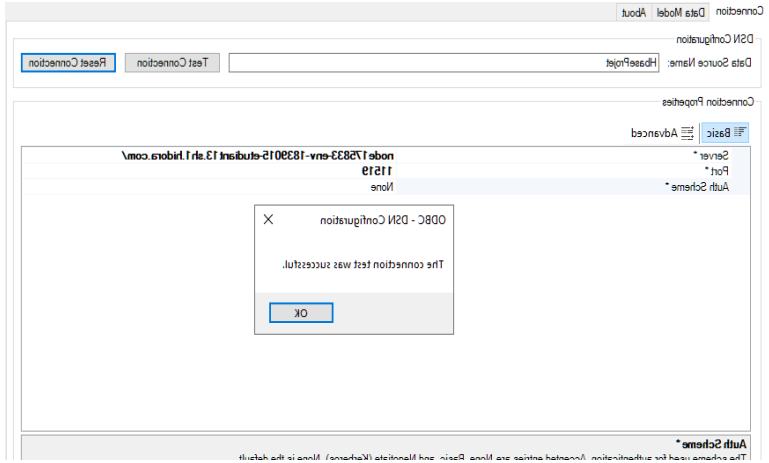
Creation de connexion ODBC

ODBC connecteur permettre PowerBI et HBASE de communiquer pour accéder les données stockées dans la table HBASE ('data_fromage')



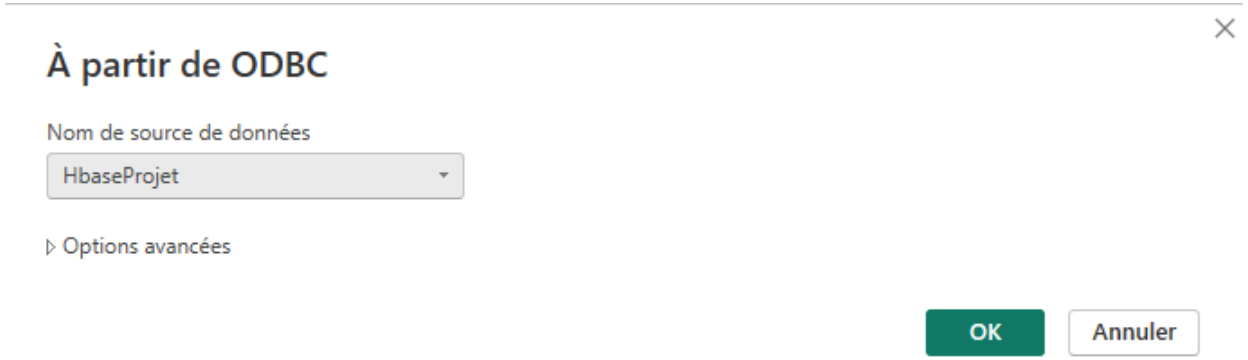


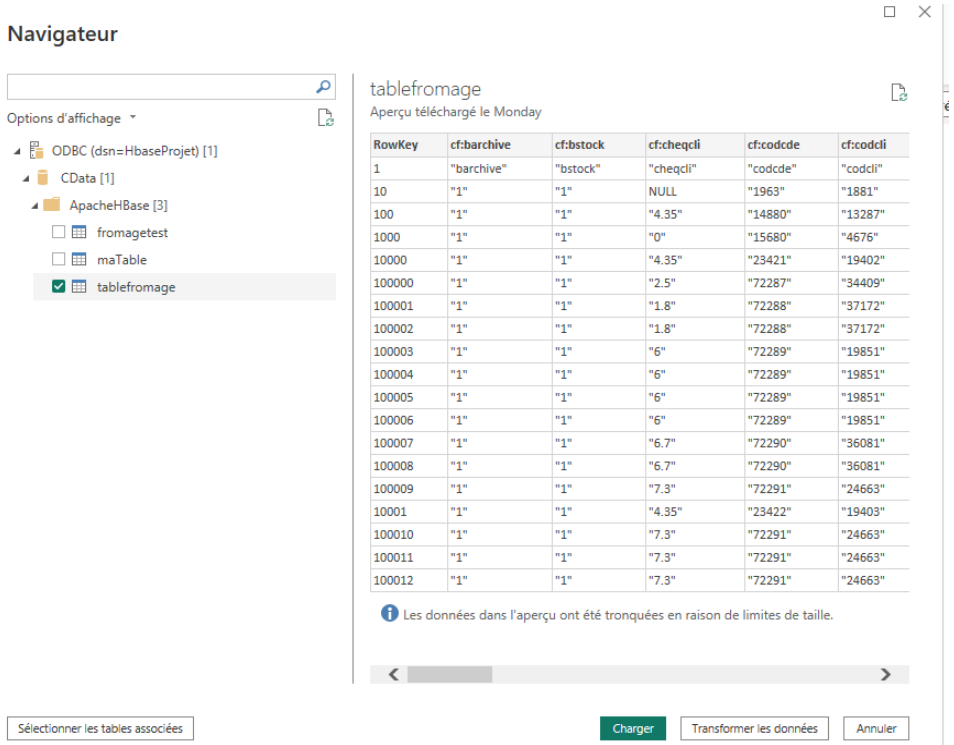




Connexion ODBC en POWERBI

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





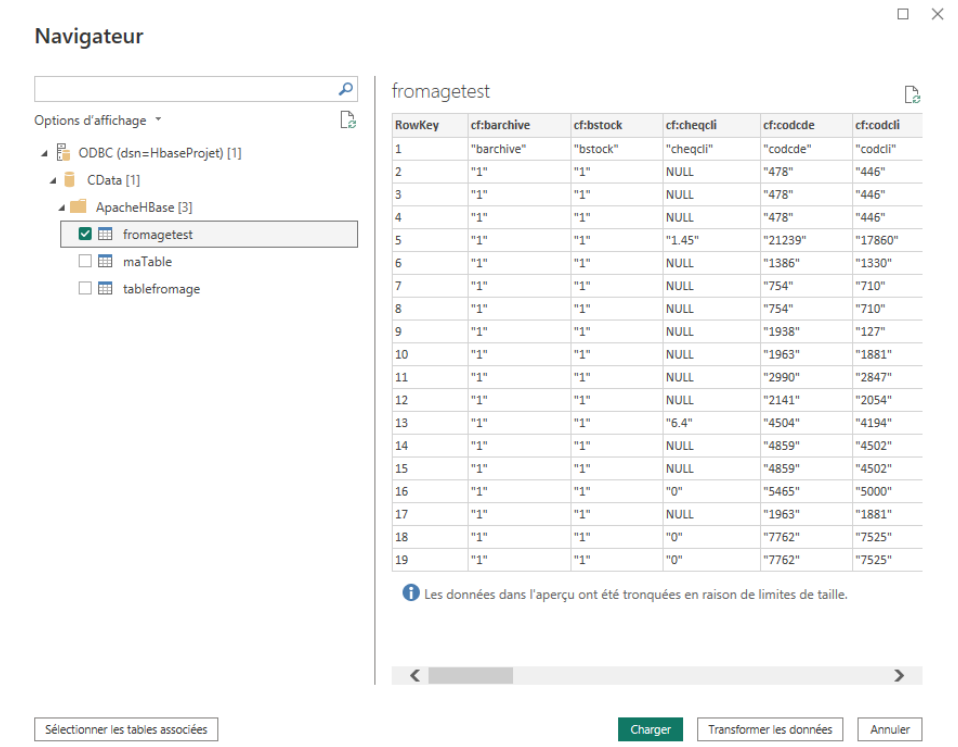
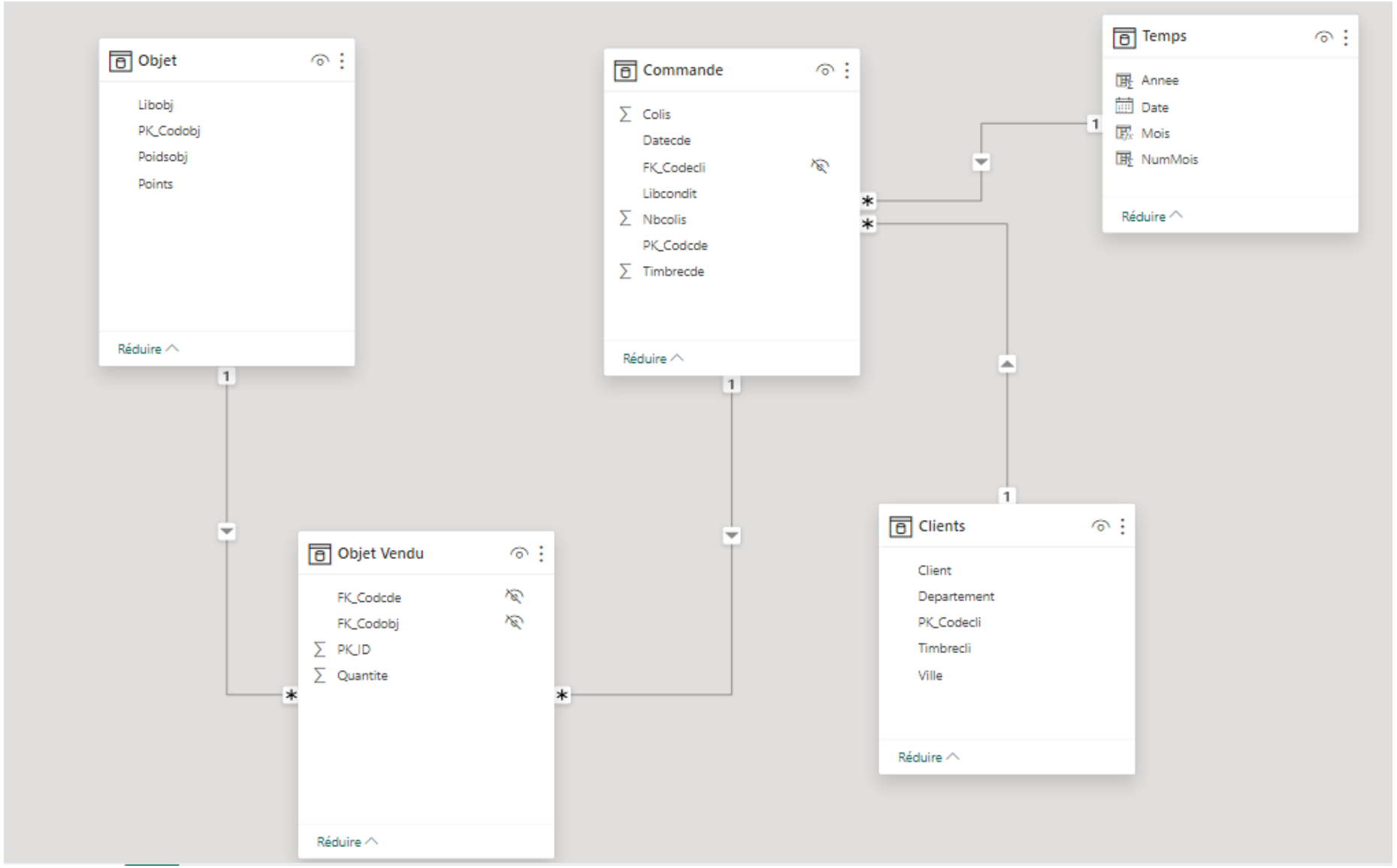
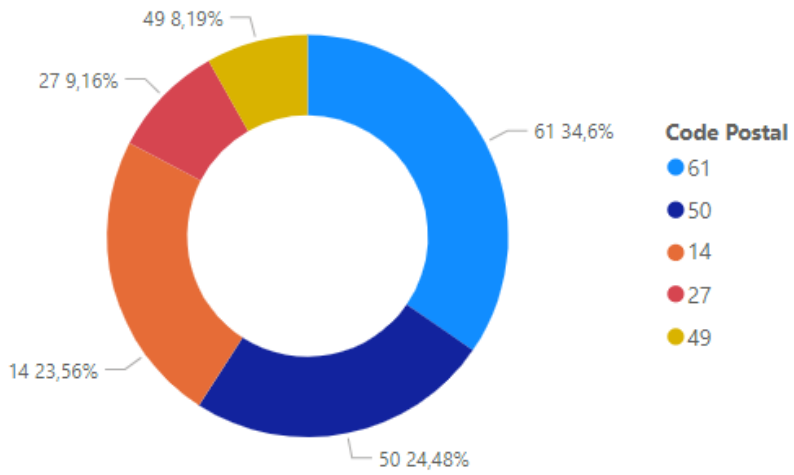


Diagramme modélisation



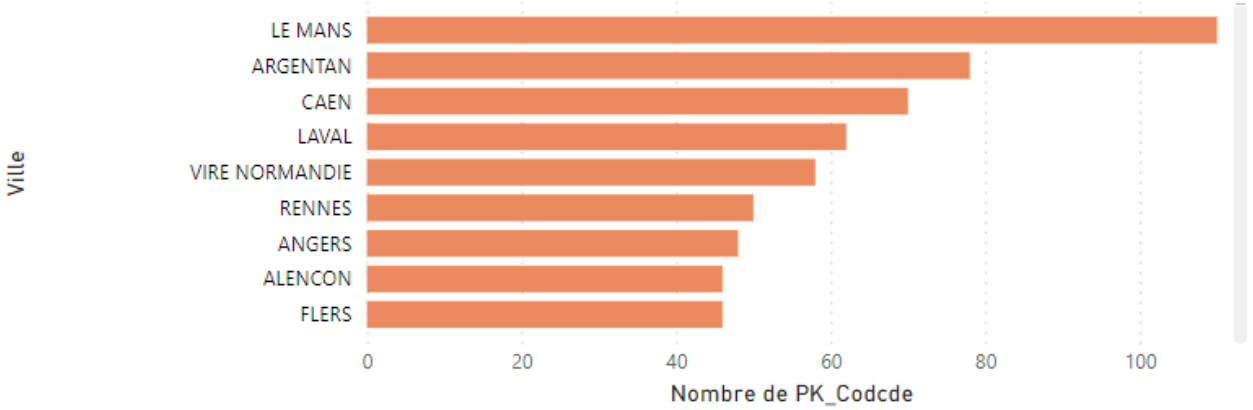
Dashboard(1/2)

Proportion des Commandes par Département



COMMANDES

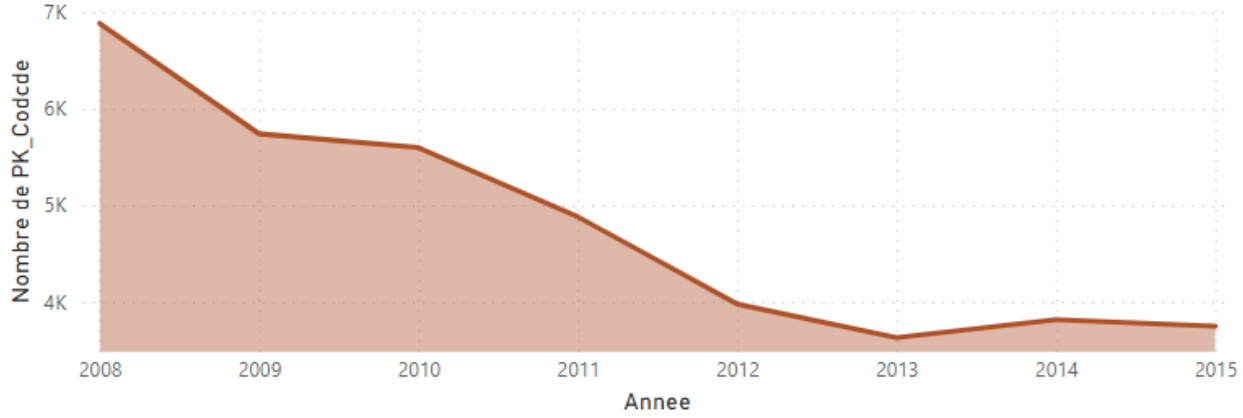
Nombre de commandes par Ville



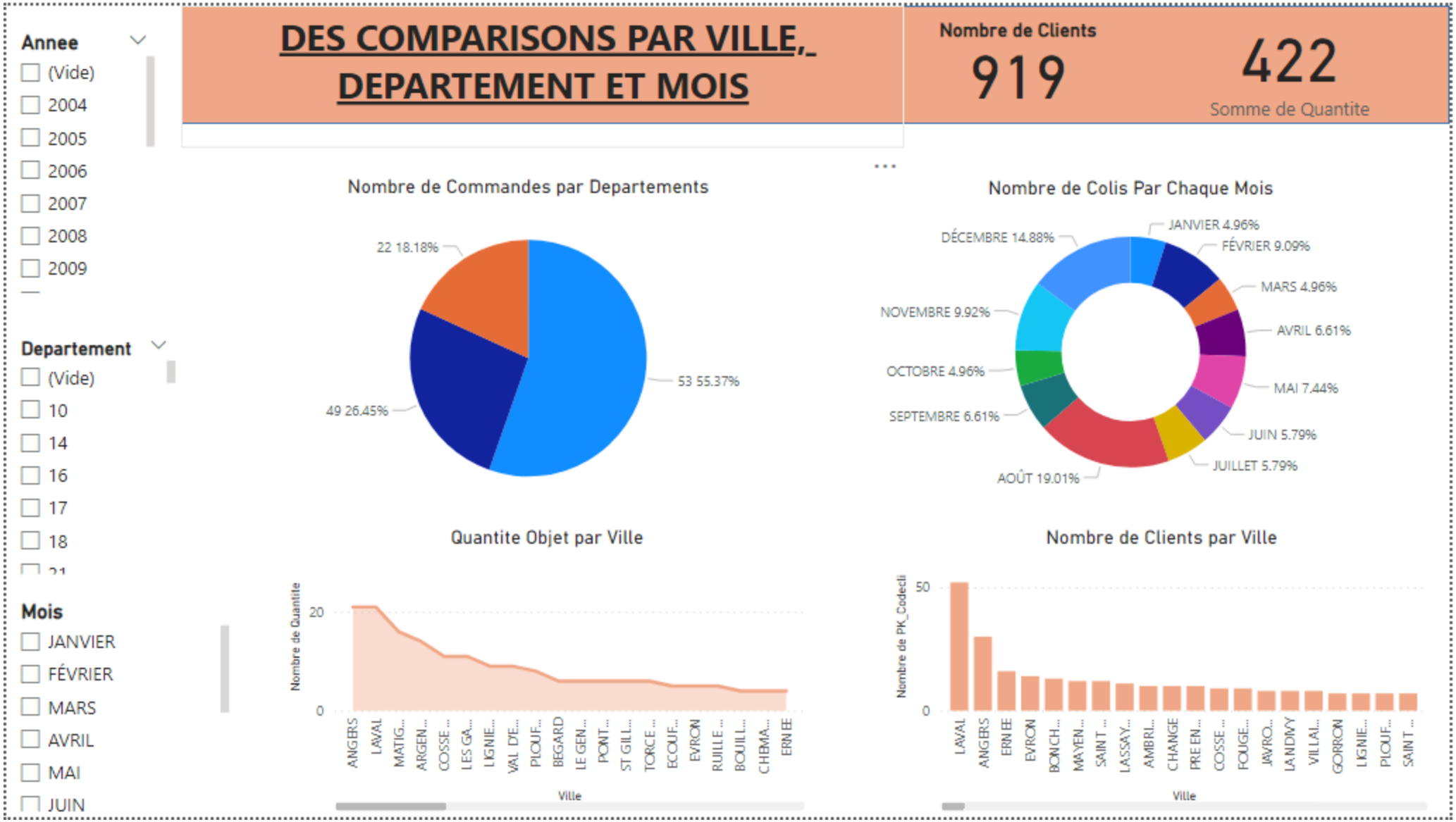
Top 10 des meilleures clients

COISEL Jacky	ARGENTAN	61
Client	Ville	Code Postal
FAUDET Michel	COURGEON	61
Client	Ville	Code Postal
FONTENEAU Thierry	CHANGE	53
Client	Ville	Code Postal
FREULON Alain	BRIOSNE LES SABL...	72
Client	Ville	Code Postal

Evolution des Commandes par Année



Dashboard(2/2)



Conclusion

- L'utilisation des technologies tels que Hadoop,Hbase et Power BI ont simplifié le traitement des données pour la fromagerie cliente. Power BI a facilité l'analyse et la visualisation des données, tandis que Hadoop a garanti une gestion efficace des ensembles de données volumineux. En parallèle, HBASE a assuré un stockage robuste des données.
- L'intégration de Spark et des machines virtuelles à jour auraient permis d'avoir des résultats plus significatifs.
- Le transfert de données de Hbase vers une base de données relationnelle(datamart) au préalable avant de requêter sur Power Bi serait plus performant
- Le projet nous a permis de développer nos compétences sur le traitement et l'analyse de données.

Téléchargement du projet

https://github.com/schwaxpl/Projet_datafromage_groupe3

Questions ?

Merci !