PROJET BIG DATA

Tanguy CHAUMETTE

Alexis MURAIL

Lucas BONNEAU

Florence BOUCHART

Abdeljalil AISSASNOU

CONTEXTE ET APPROCHE

- Fromagerie le Bon Mayennais
- Système de cadeaux récompensant la fidélité
- 3 Lots de statistiques à livrer

DESCRIPTION DU FICHIER SOURCE

	codcli	genrecli	nomcli	prenomcli	cpcli	villecli	codcde	datcde	timbrecli	timbrecde	...	qte	Colis	libobj	Tailleobj	Poidsobj	points	indispobj	libcondit	prixcond
0	446	Mme	CHRETIEN	Daniel	14540	BOURGUEBUS	478	2004-10-22 00:00:00	5.0	4.80	...	2.0	1.0	Polo	XL	230	60.0	0	Carton Tete de menagere	0
1	446	Mme	CHRETIEN	Daniel	14540	BOURGUEBUS	478	2004-10-22 00:00:00	5.0	4.80	...	2.0	1.0	T-shirt Blanc	L	170	60.0	0	Carton Tete de menagere	0
2	446	Mme	CHRETIEN	Daniel	14540	BOURGUEBUS	478	2004-10-22 00:00:00	5.0	4.80	...	1.0	1.0	Montre	Homme	30	150.0	0	Carton Tete de menagere	0
3	17860	M.	VERARDO	Anthony	35400	SAINT MALO	21239	2006-10-03 00:00:00	0.0	3.90	...	1.0	1.0	T-shirt Blanc	XL	180	60.0	0	Distingo 500 g	34
4	1330	Mme	ROBERT	Yvonne	61000	ALENCON	1386	2004-11-23 00:00:00	4.0	6.50	...	3.0	1.0	Tete de menagere	Confidence	250	100.0	0	Carton Tete de menagere	0

Démo vidéo

LOT 1

Demande du client :

- Filtres :
  - Années comprises entre 2006 et 2010
  - Départements 53, 61, 28

- 100 meilleures commandes (qté la + grande puis timbrecli le + grand) en affichant
  - la ville
  - la somme des quantités d'articles commandés (la + grande)
  - la valeur de « timbrecde »

- Export dans un fichier Excel

Timbrecli = nombre de timbrecde

APPROCHE POUR RÉALISER LE LOT 1

- Filtrage par années et numéros de départements dans le mapper
- Obtention des 100 meilleures commandes via le reducer en ressortant la valeur de timbrecde.
- Export dans un fichier excel

<u>Répartition du travail en 2 groupes :</u>

- un sur le mapper
- un sur le reducer

Puis assemblage des deux et connexion à HDFS

CODE – LOT 1

```python
import sys
import csv

csv_reader = csv.reader(sys.stdin)
csv_file_path = "C:/Users/abdel/Desktop/dataw_fro03.csv"
with open(csv_file_path, 'r') as file:

    csv_reader = csv.reader(file)
    next(csv_reader, None)

# Parcourez les lignes du CSV
    for row in csv_reader:
        # Extraire les champs du CSV
        datcde, villecli, timbrecde, codcde, cpcli, qte = (
            row[7],
            row[5],
            row[9],
            row[6],
            row[4],
            row[15],
        )
```

ETAPE DU MAPPER

```python
#Annee
if row[7] is not None and row[7] != "NULL":
    datcde= row[7][0:4]
else:
    datcde=0

#Cp
if row[4] is not None:
    cpcli=row[4][0:2]
else:
    cpcli=" "

#Ville
if row[5] is not None:
    villecli=row[5]
else :
    villecli=" "

#Timbre
if row[9] is not None:
    timbrecde=row[9]
else :
    timbrecde = 0.0

#Codcde
if row[6] is not None:
    codcde=row[6]
else :
    codcde = 0.0

#Qte
if row[15] is not None:
    qte=row[15]
else :
    qte = 0

if (int(datcde) >= 2006) and (int(datcde) <= 2010) and (cpcli == "53" or cpcli == "61" or cpcli == "28"):
    print(codcde + '\t', villecli + '\t', timbrecde + '\t', qte)
```

CODE – LOT 1

```python
import numpy as np
import pandas as pd
import sys

# Initialisation de listes pour les colonnes du df
codcde = []
timbrecli = []
timbrecde = []
villecli = []
qte=[]

for line in sys.stdin:
    line = line.strip()
    mapper = line.split('\t')

    codcde.append(mapper[0])
    villecli.append(mapper[1])
    timbrecli.append(mapper[2])
    timbrecde.append(mapper[3])
    qte.append(mapper[4])

# Création du DataFrame
df = pd.DataFrame({
    'Code Commande': codcde,
    'Ville Client': villecli,
    'Quantités': qte,
    'Timbres Commande' : timbrecde,
    'Timbres Client' : timbrecli
})
```

ETAPE DU REDUCER

```python
# Regroupement par Commande, Timbres et Villes
df_group = df.groupby(['Code Commande', 'Timbres Client', 'Timbres Commande','Ville Client']).sum()

# Aplatissement du DataFrame
df_group_flat = df_group.reset_index()
df_group_flat = df_group_flat[['Code Commande','Ville Client','Quantités','Timbres Commande', 'Timbres Client']]

# Renommez les colonnes pour avoir des noms plus explicites
df_group_flat.columns = ['Code Commande', 'Ville Client', 'Somme des Quantités', 'Timbres Commande', 'Timbres Client']

# Classement des résultats dans l'ordre des quantités puis des timbres dans l'ordre décroissant
df_sort = df_group_flat.sort_values(['Somme des Quantités','Timbres Client'], ascending=False)

# Isolement des 100 premiers résultats
df_100 = df_sort.head(100)

# Export csv
df_100.to_csv('/datavolume1/lot1.csv')
```

CODE – LOT 1

EXPORT

	A	B	C	D	E	F	G
1	,Code Commande,Ville Client,Somme des Quantit?s,Timbres Commande,Timbres Client						
2	2690,23617, COULIMER, 9, 6.4, 0						
3	2853,24165, SAINT LANGIS LES MORTAGNE, 9, 4.35, 0						
4	3334,25668, BOITRON, 9, 6.4, 0						
5	3452,25996, LE GENEST SAINT ISLE, 9, 6.4, 0						
6	3915,27431, ATHIS VAL DE ROUVRE, 9, 6.4, 0						
7	4705,29976, ASTILLE, 9, 6.4, 0						
8	4777,30221, SAINT GERMAIN DU CORBEIS, 9, 5.1, 0						
9	5181,31453, NUILLE SUR VICOIN, 9, 6.4, 0						
10	7174,37923, ANDOUILLE, 9, 5.1, 0						
11	8429,42082, MENIL FROGER, 9, 5.5, 0						
12	8502,42326, NORMANDEL, 9, 5.5, 0						
13	8927,43750, ST BOMER LES FORGES, 8, 7, 7						
14	2459,22880, DOMFRONT EN POIRAIE, 8, 6.4, 6.42						
15	3800,27095, SAINT CALAIS DU DESERT, 8, 6.4, 5.13						
16	4259,28550, QUELAINES SAINT GAULT, 8, 6.4, 0						
17	4865,30501, LA HAUTE CHAPELLE, 8, 4.35, 0						
18	6386,35456, DAMIGNY, 8, 5.8, 0						
19	6387,35457, SAINT LANGIS LES MORTAGNE, 8, 5.8, 0						
20	6837,36946, BAZOCHES SUR HOENE, 8, 5.1, 0						

LOT 2

Demande du client :

- Filtres :
  - Années comprises entre 2011 et 2016
  - Départements 22, 49, 53

- 5% aléatoires des 100 meilleures commandes(qté la + grande) sans timbre en affichant
  - La ville
  - La somme des quantités d'articles commandés
  - La moyenne des quantités de chaque commande

- Extraction du PieChart dans un fichier pdf
- Export dans un fichier Excel

Timbrecli = nombre de timbrecde

APPROCHE POUR RÉALISER LE LOT 2

- Mapper : Filtrage
  (même principe que pour le lot 1 avec des années et numéros de départements différents).
- Reducer : Tri et échantillonnage
- Extraction du PieChart dans un fichier pdf
- Export dans un fichier excel

<u>Répartition du travail en 2 groupes :</u>

- Un sur le mapper
- Un sur le reducer

Puis assemblage des deux et connexion à HDFS

CODE – LOT 2

```python
import sys
import csv

csv_reader = csv.reader(sys.stdin)
csv_file_path = "C:/Users/abdel/Desktop/dataw_fro03.csv"
with open(csv_file_path, 'r') as file:

    csv_reader = csv.reader(file)
    next(csv_reader, None)

# Parcourez les lignes du CSV
    for row in csv_reader:
        # Extraire les champs du CSV
        datcde, villecli, timbrecli, codcde, cpcli, qte = (
            row[7],
            row[5],
            row[8],
            row[6],
            row[4],
            row[15],
        )
```

ETAPE DU MAPPER

```python
#Annee
if row[7] is not None and row[7] != "NULL":
    datcde= row[7][0:4]
else:
    datcde=0

#Cp
if row[4] is not None:
    cpcli=row[4][0:2]
else:
    cpcli=" "

#Ville
if row[5] is not None:
    villecli=row[5]
else :
    villecli=" "
#Codcde
if row[6] is not None:
    codcde=row[6]
else :
    codcde=" "

#Timbre
if row[8] is not None and row[8] != "NULL":
    timbrecli=row[8]
else :
    timbrecli = 0.0

#Qte
if row[15] is not None:
    qte=row[15]
else :
    qte = 0

if (int(datcde) >= 2011) and (int(datcde) <= 2016) and (cpcli == "22" or cpcli == "49" or cpcli == "53"):
    print(codcde + '\t' , villecli + '\t', timbrecli + '\t', qte)
```

CODE – LOT 2

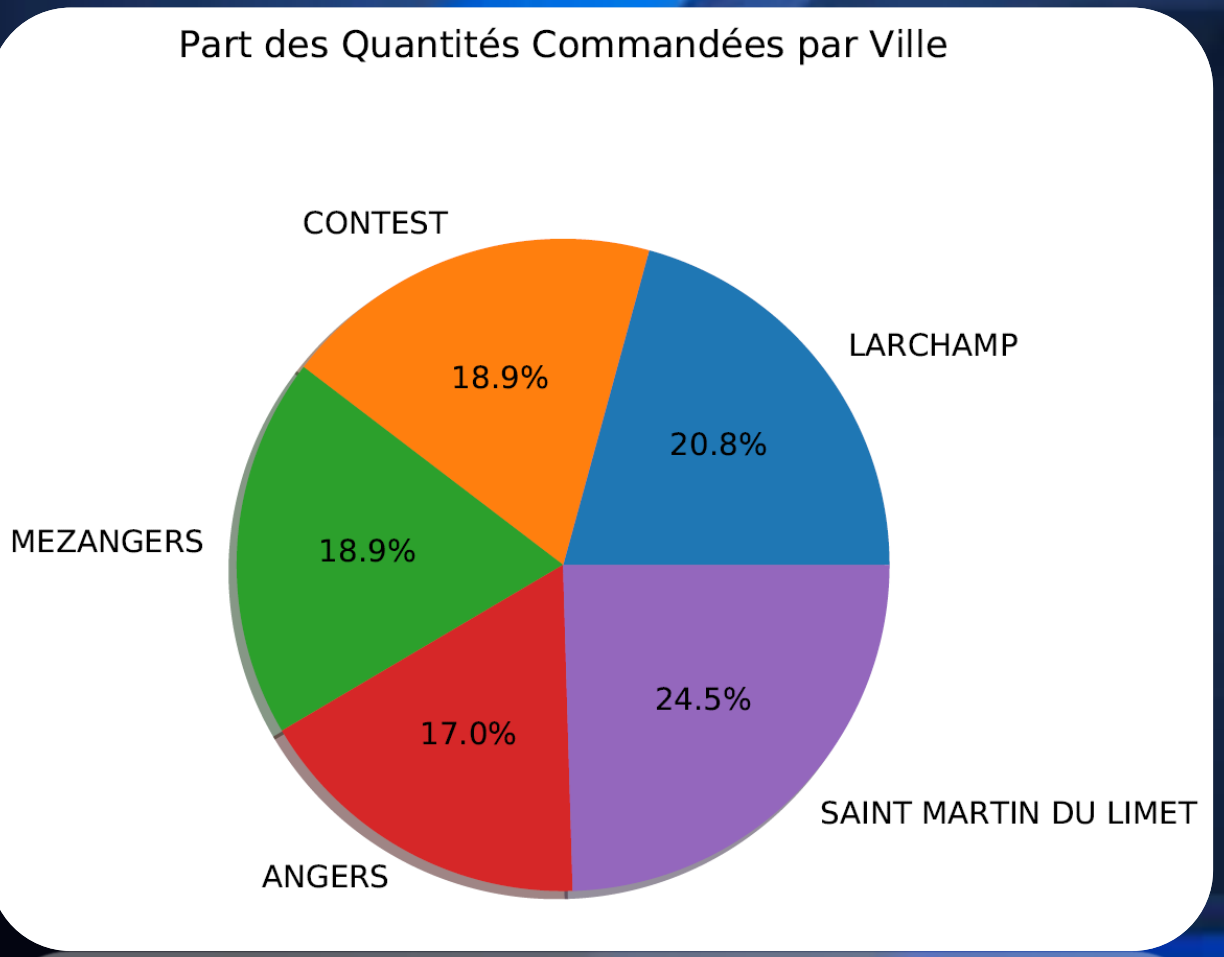
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sys

""" Initialisation de listes pour les colonnes du df"""
codcde = []
timbrecli = []
villecli = []
qte=[]


for line in sys.stdin:
    line = line.strip()
    mapper = line.split('\t')

    codcde.append(mapper[0])
    villecli.append(mapper[1])
    timbrecli.append(float(mapper[2]))
    qte.append(int(mapper[3]))
    print('ok;')

""" Création du DataFram"""
df = pd.DataFrame({
    'Code Commande': codcde,
    'Ville Client': villecli,
    'Quantites': qte,
    'Timbres Client' : timbrecli
})

df_no_timbrecli = df[
    df['Timbres Client']==0
]
```

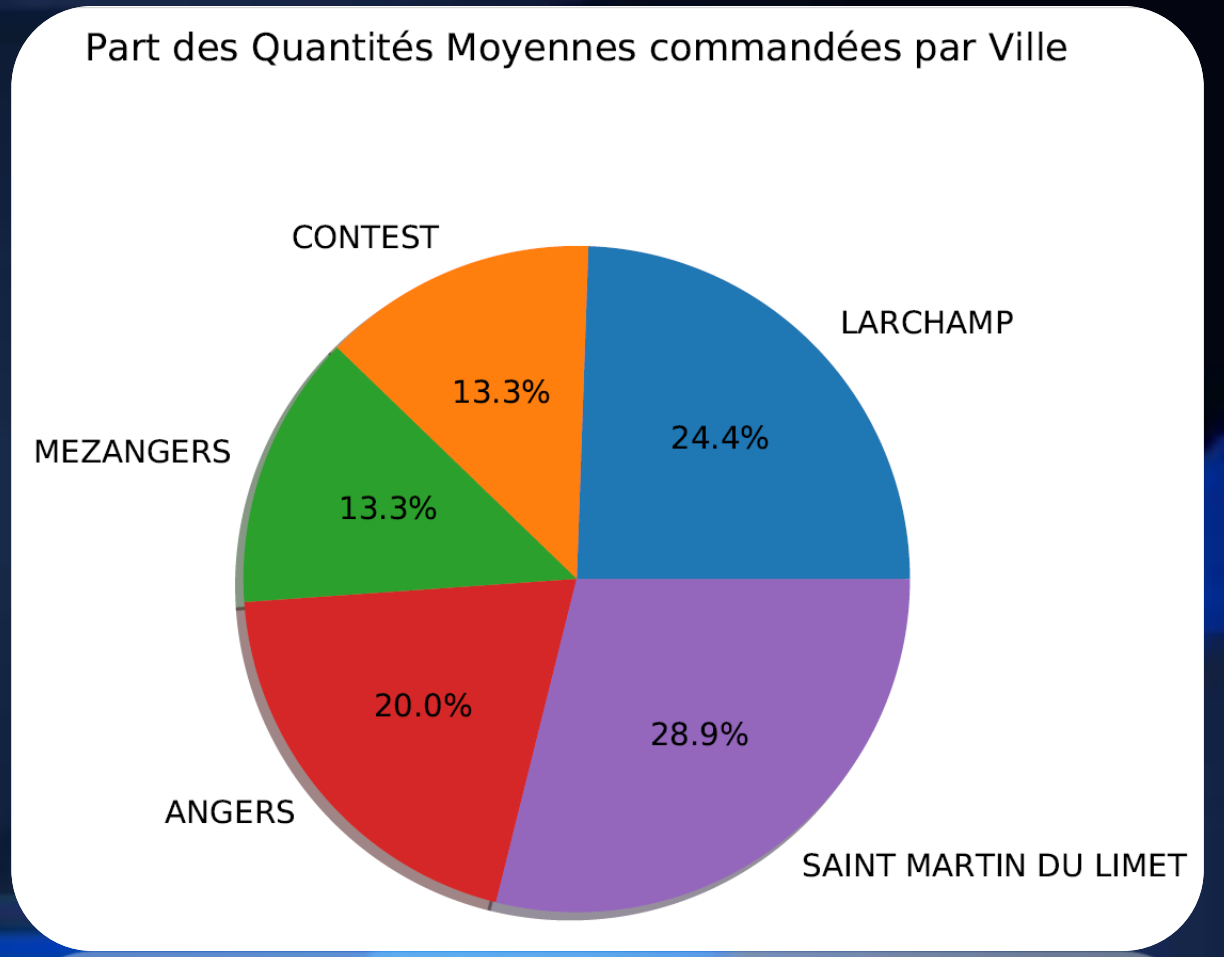
ETAPE DU REDUCER

```python
"""#e:Export csv """
df_sample.to_csv('/datavolume1/lot2.csv')

"""# Graph numéro 1 : Somme des Quantités par ville"""
DF=df_sample
col_x = 'Ville Client'
col_y = 'Somme des quantites'
X = DF[col_x]
Y = DF[col_y]
plt.pie(Y, labels = X, startangle = 0, shadow='True', autopct='%1.1f%%')
plt.title(label='Part des Quantités Commandées par Ville', y=1.1)
plt.savefig('/datavolume1/graph_pie_qte_ville.pdf', format='pdf', bbox_inches='tight')
plt.close()
"""# Graph numéro 2 : Moyenne des Quantités par vie"""
DF=df_sample
col_x = 'Ville Client'
col_y = 'Moyenne des quantites'
X = DF[col_x]
Y = DF[col_y]
plt.pie(Y, labels = X, startangle = 0, shadow='True', autopct='%1.1f%%')
plt.title(label='Part des Quantités Moyennes commandées par Ville', y=1.1)
plt.savefig('/datavolume1/graph_pie_moy_ville.pdf', format='pdf', bbox_inches='tight')

""" Regroupement par Commande, Timbres et Villes """
df_group = df_no_timbrecli.groupby(['Code Commande', 'Timbres Client', 'Ville Client']).agg({'Quantites': ['sum','mean']})

"""  Aplatissez le DataFrame résultant avec reset_index"""
df_group_flat = df_group.reset_index()

"""# Renommez les colonnes pour avoir des noms plus explicites """
df_group_flat.columns = ['Code Commande', 'Timbres Client', 'Ville Client', 'Somme des quantites', 'Moyenne des quantites']

"""# Classement des résultats dans l'ordre des quantités puis des timbres dans l'ordre décroiss"""
df_sort = df_group_flat.sort_values(['Somme des quantites','Timbres Client'], ascending=False)

"""# Isolement des 100 premiers résultats"""
df_100 = df_sort.head(100)

if not df_100.empty:
    """# Isolement de 5 résultats de façon aatoire"""
    indices = np.random.choice(df_100.index, size=int(len(df_100) * 0.05), replace=False)
    df_sample = df_100.loc[indices].reset_index(drop=True)
else:
    df_sample = df_100.copy()
```

CODE – LOT 2

EXPORTS





	A	B	C	D	E	F	G
1	,Code Commande,Timbres Client,Ville Client,Somme des quantites,Moyenne des quantites						
2	0,72129,0.0,NOYANT VILLAGES,19,3.8						
3	1,70779,0.0,DESERTINES,12,4.0						
4	2,72597,0.0,CHAILLAND,9,2.25						
5	3,54350,0.0,LE BIGNON DU MAINE,10,2.0						
6	4,61965,0.0,LE MESNIL EN VALLEE,21,3.0						

LOT 3

Demande du client : Le client souhaite explorer ses données mises en forme via une interface graphique interactive

Réponse apportée :

- Mise en place d'une base de données NoSQL HBase pour restructurer les données présentes sur les clusters Hadoop
- Import via procédure TSV
- Connexion ODBC entre Power BI et Hbase
- Création de Dashboards via Power BI

Démo vidéo

CODE – LOT 3 DASHBOARD POWER BI

Démo :

Résumé du filtrage et nettoyage des données

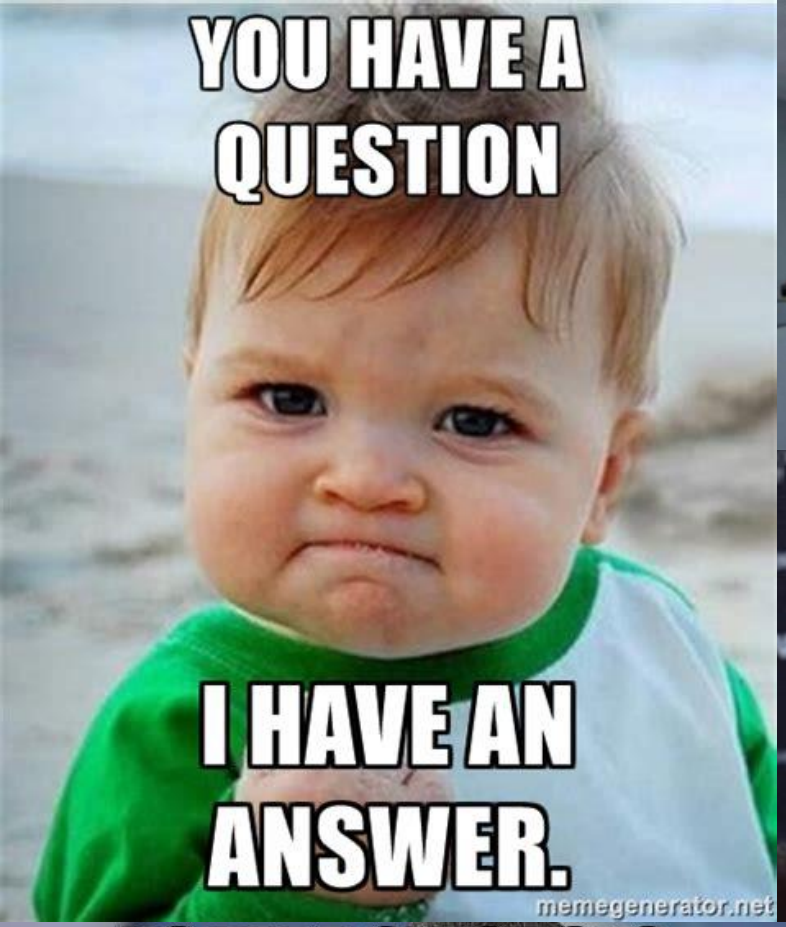
Visualisation du Dashboard sur Power BI

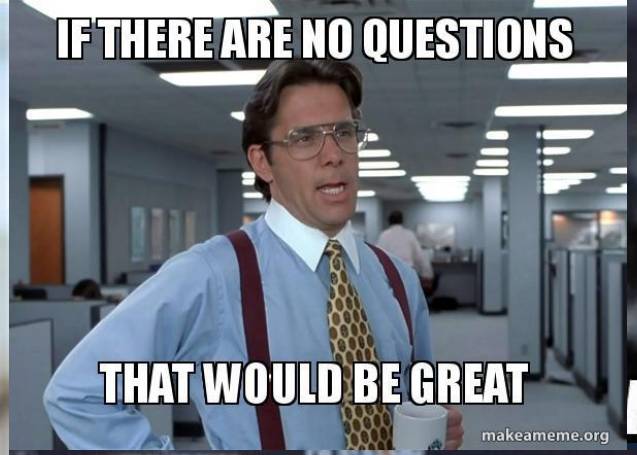
FEEDBACK

- Projet très concret
- Adapté au temps imparti
- Application de pas mal d'outils étudiés au préalable
- A permis une meilleure compréhension d'Hadoop et Hbase

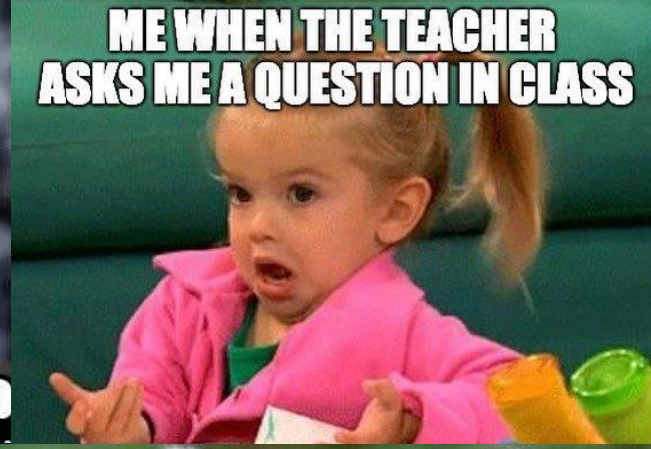
- Questionnement sur la confidentialité des données clients (RGPD)
- Enoncé difficile à comprendre


Merci de votre attention !


YOU HAVE A QUESTION
I HAVE AN ANSWER.
memegenerator.net


IF THERE ARE NO QUESTIONS
THAT WOULD BE GREAT
makeameme.org


ANY QUESTIONS ???


ME WHEN THE TEACHER ASKS ME A QUESTION IN CLASS

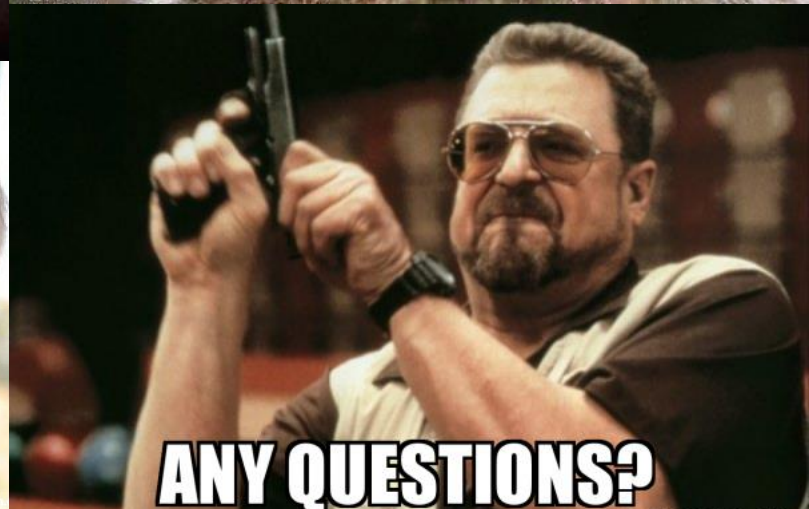

Anybody got any questions?


I have questions


NO HARD QUESTIONS.
NEXT, PLEASE.


Questions?


QUESTIONS?
I DIDNT THINK SO!


ANY QUESTIONS?