

---

---

# Développement d'un système de stockage distribué pour le jeu de données de GDELT

---

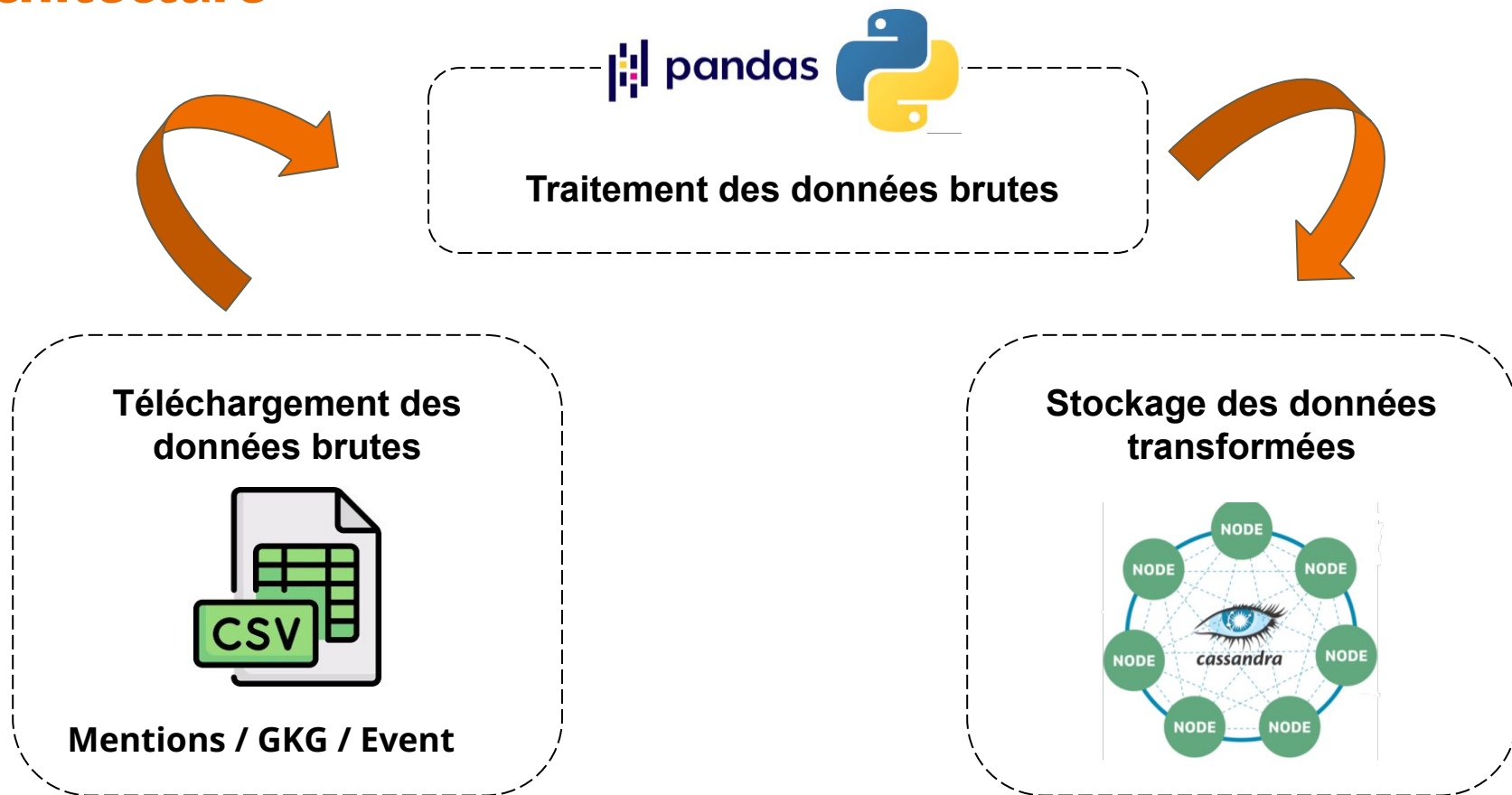
---

SAMB Sokhna , DOUAZI Ghita , SANAD Mohammed , ACKOUNDOUN Abo

# PLAN

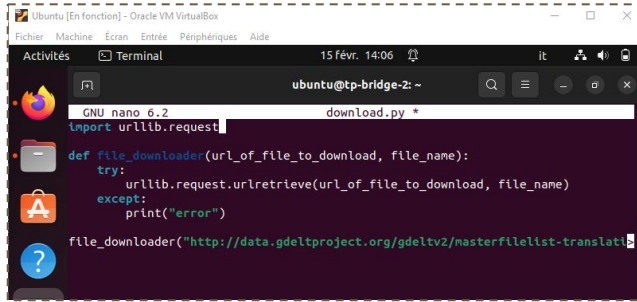
- 1.Présentation et choix de l'architecture
- 2.Modélisation des données et requêtage
- 3.Performances et limites
- 4.Demo

# Architecture



# Téléchargement des données brutes

Télécharger les URL

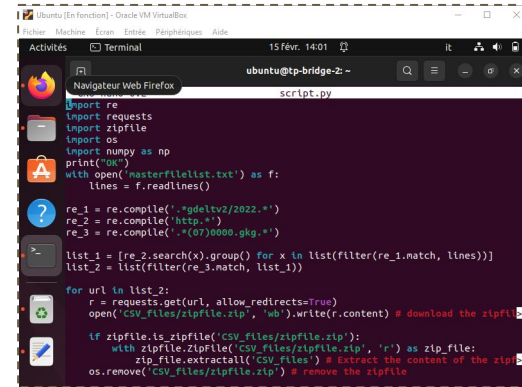


```
GNU nano 6.2 download.py *
import urllib.request

def file_downloader(url_of_file_to_download, file_name):
    try:
        urllib.request.urlretrieve(url_of_file_to_download, file_name)
    except:
        print("error")

file_downloader("http://data.gdeltproject.org/gdeltv2/masterfilelist-translati
```

Télécharger les CSV



```
import re
import requests
import zipfile
import os
import numpy as np
print("OK")
with open('masterfilelist.txt') as f:
    lines = f.readlines()

re_1 = re.compile('.*gdeltv2/2022.*')
re_2 = re.compile('http.*')
re_3 = re.compile('.*(07)0000.gkg.*')

list_1 = [re_2.search(x).group() for x in list(filter(re_1.match, lines))]
list_2 = list(filter(re_3.match, list_1))

for url in list_2:
    r = requests.get(url, allow_redirects=True)
    open('CSV_files/zipfile.zip', 'wb').write(r.content) # download the zipfile

    if zipfile.is_zipfile('CSV_files/zipfile.zip'):
        with zipfile.ZipFile('CSV_files/zipfile.zip', 'r') as zip_file:
            zip_file.extractall('CSV_files') # extract the content of the zip file
        os.remove('CSV_files/zipfile.zip') # remove the zipfile
```

# Choix de l'architecture : Cassandra

## Les avantages :

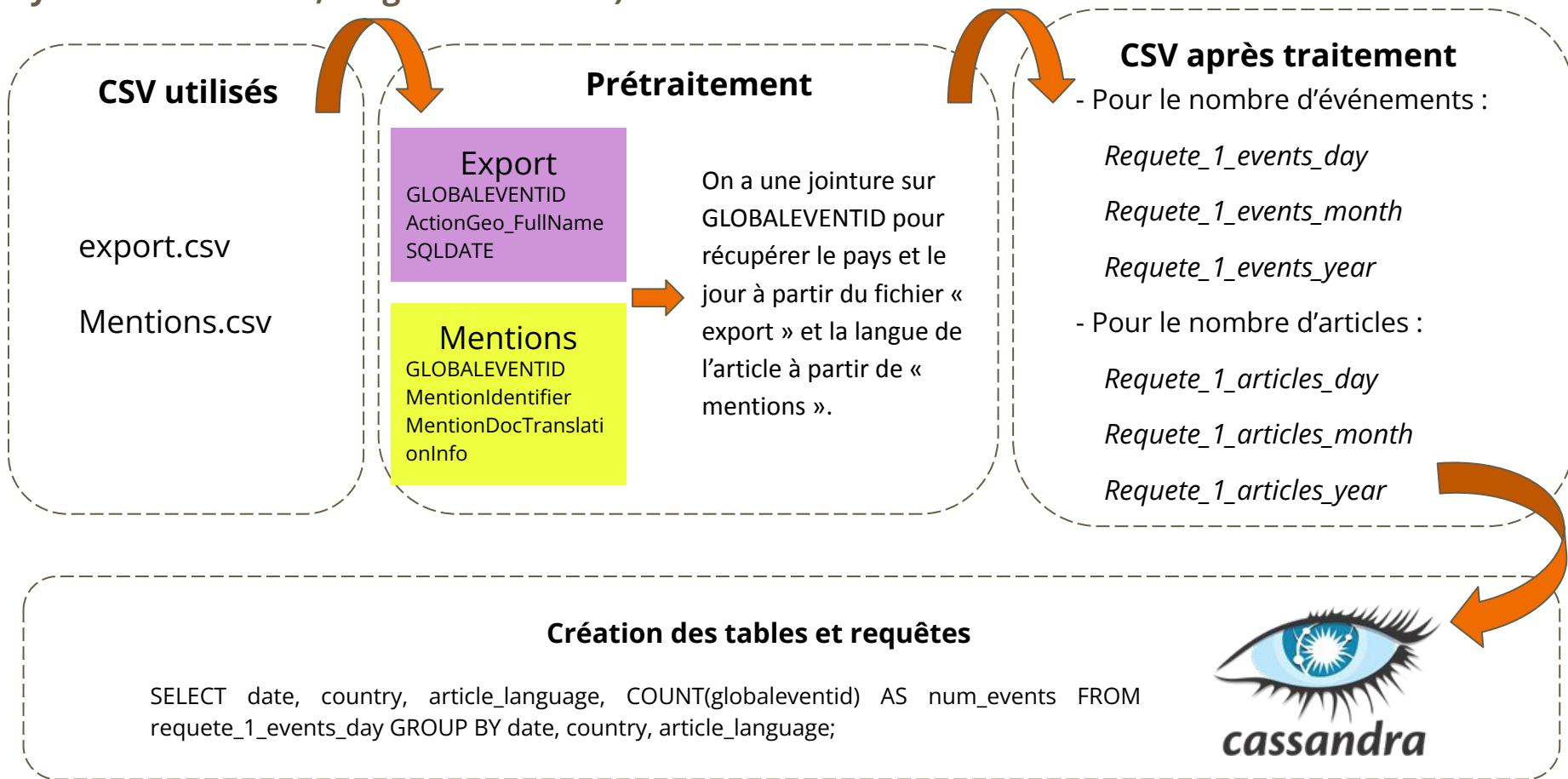
- ❖ Haute disponibilité des données.
- ❖ Scalabilité.
- ❖ Système distribué et résilience à la panne
- ❖ Modèle de données flexible.

## Les inconvénients :

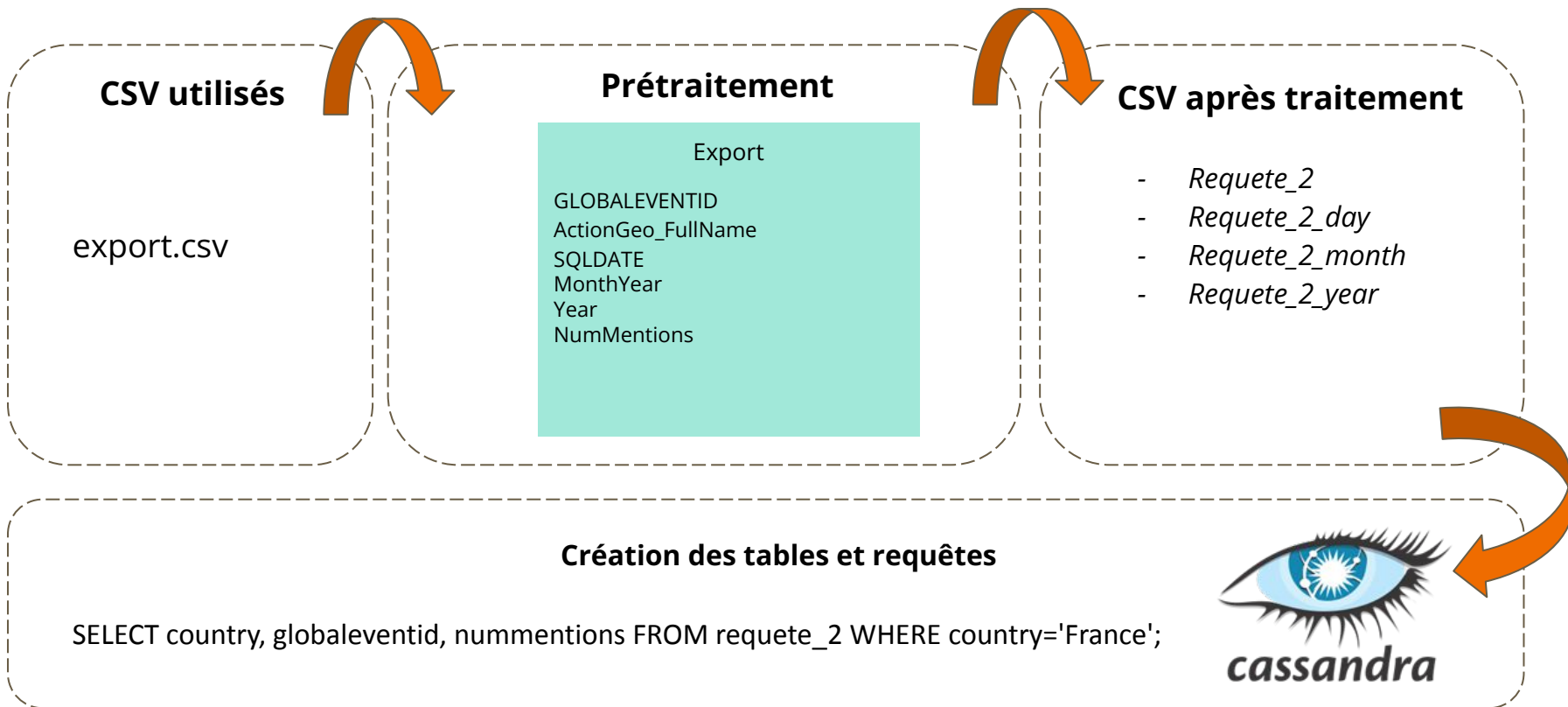
- ❖ Complexité
- ❖ Requêtes limitées
- ❖ Mise à jour de schéma
- ❖ Ne supporte pas les agrégats

# Requêtage

## Requête 1 : Afficher le nombre d'articles/événements qu'il y a eu pour chaque triplet (jour, pays de l'évènement, langue de l'article)

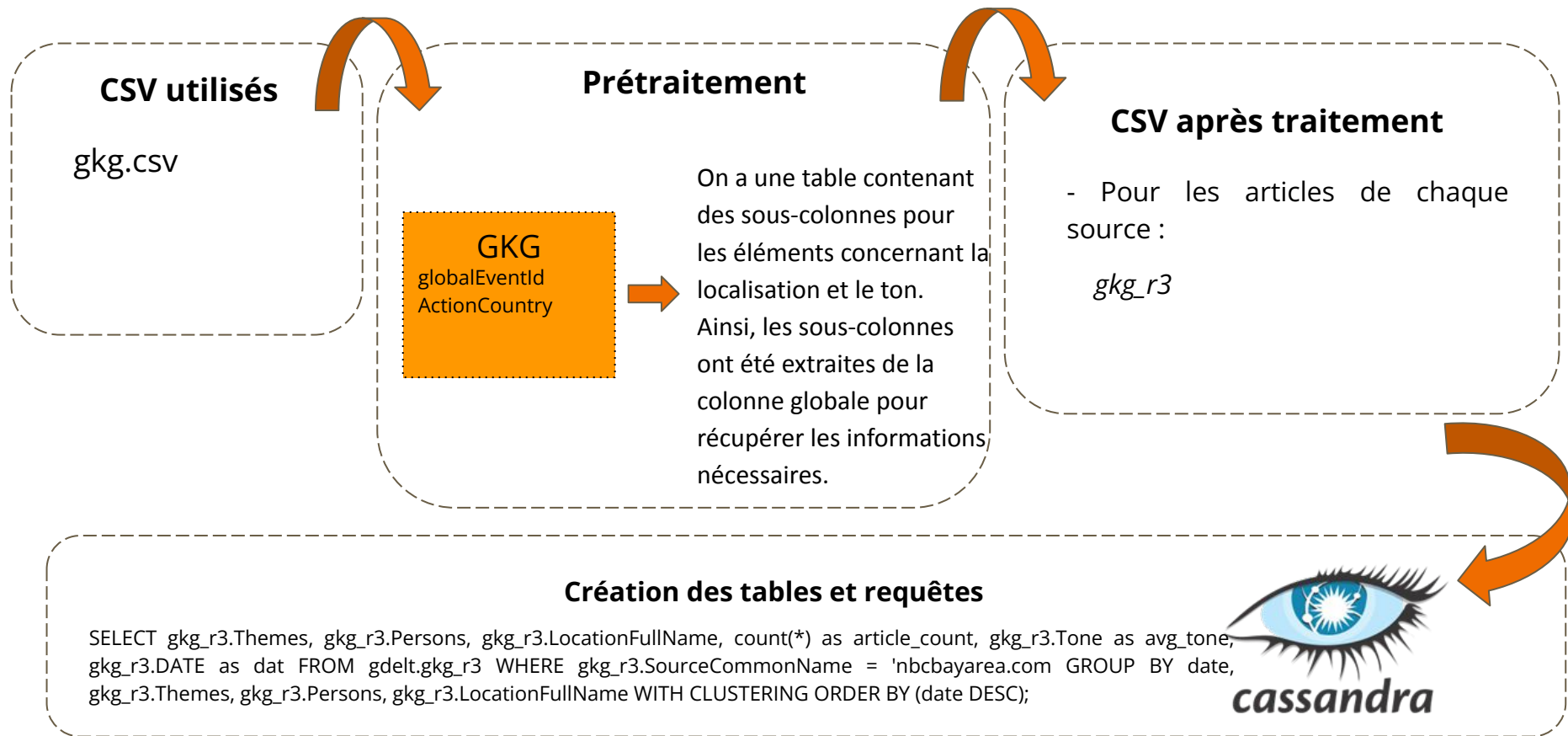


## Requête 2 : Pour un pays donné en paramètre, affichez les événements triés par le nombre de mentions (tri décroissant) et permettez une agrégation par jour/mois/année



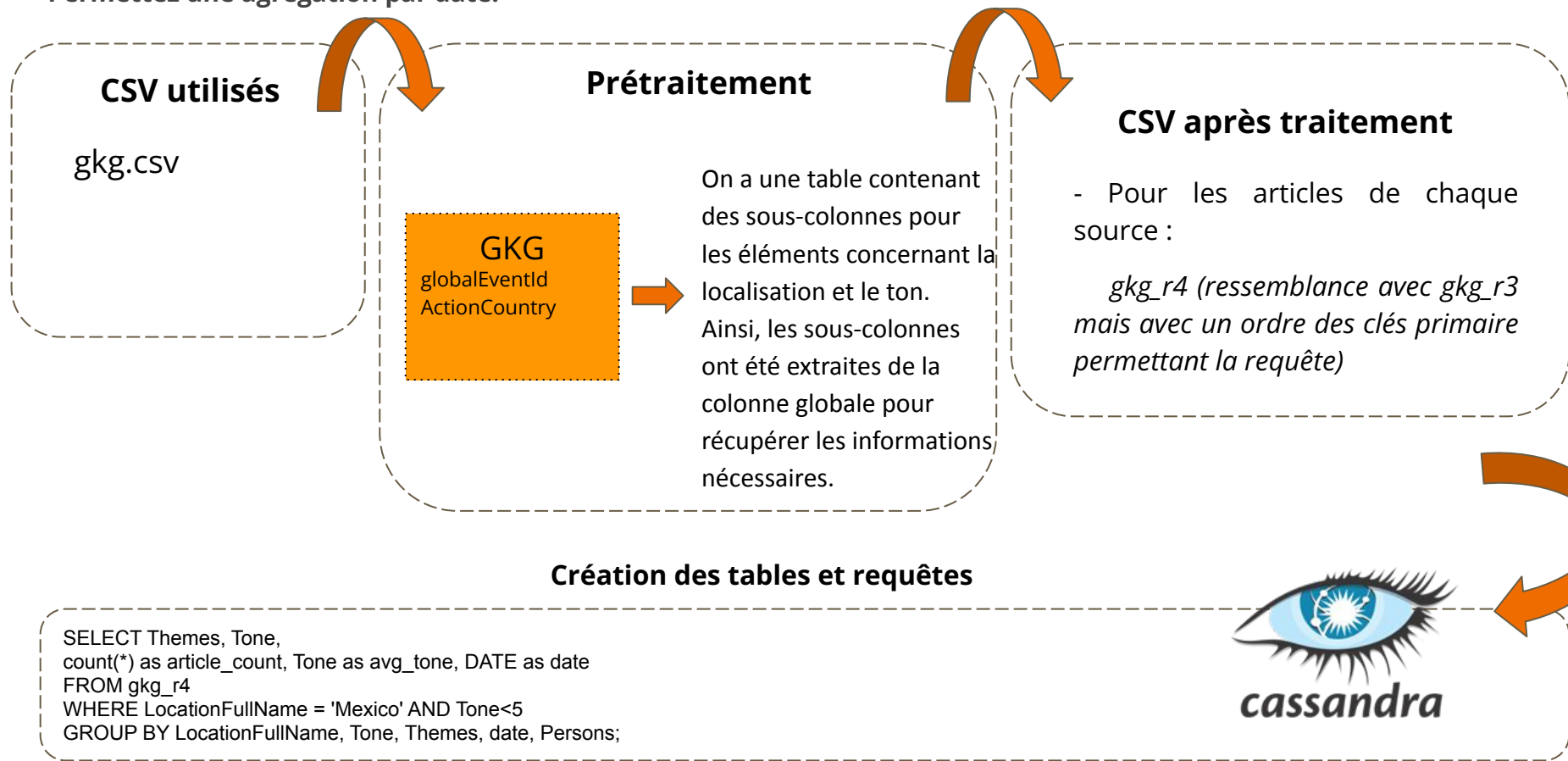


**Requête 3 :** Pour une source de données passée en paramètre, affichez les thèmes, personnes, lieux dont les articles de cette sources parlent ainsi que le nombre d'articles et le ton moyen des articles (pour chaque thème/personne/lieu); permettez une agrégation par jour/mois/année



**Requête 4 :** Pour un pays en paramètre, affichez les thèmes ainsi que le nombre d'articles et le ton moyen des articles (pour chaque Themes/date/Personnes);

Permettez une agrégation par date.



# Performances et limites

## Performances :

- Capacité de gérer un grand volume de données
- Réplication multi-centre des données
- Cohérence (gérabilité des pannes) ajustable
- Langage CQL proche du SQL

## Limites :

- Incapacité de faire des jointures
- système de stockage de clés / valeurs. Ce qui signifie qu'il vous faut « modéliser » vos informations autour des requêtes



**Demo**