

Dossier DATALAKE – Analyse d’offre d’emploi provenant de Linkedin



Table des matières

I. Introduction et contexte	3
II. Démarche à suivre	5
A. Ingestion des métadonnées lors du passage en « 1_Landing_Zone ».....	5
B. Création de fonctions permettant la récupération de données	8
C. Ajout des données récoltés dans un fichier csv	9
III. Exécution du code.....	13
IV. Data Visualisation	14
A. Nettoyage des données	14
B. Modèle de données :.....	14
C. Aperçu du tableau de bord :	15

I. Introduction et contexte

Dans le cadre de l'enseignement sur la gestion de données massives et les Data Lake au sein du Master 2 Business Intelligence et Analytics, nous avons été amenés à réaliser un projet visant à mettre en application les connaissances acquises dans ce cours.

Le Data Lake est une méthode de stockage de données massives utilisée dans les sujets liés au Big Data. Le Data Lake est principalement utilisé pour le stockage de gros volumes de données sous divers formats. Ces informations peuvent être structurées, semi-structurées ou non structurées. Le stockage englobe tous les types de bases données.

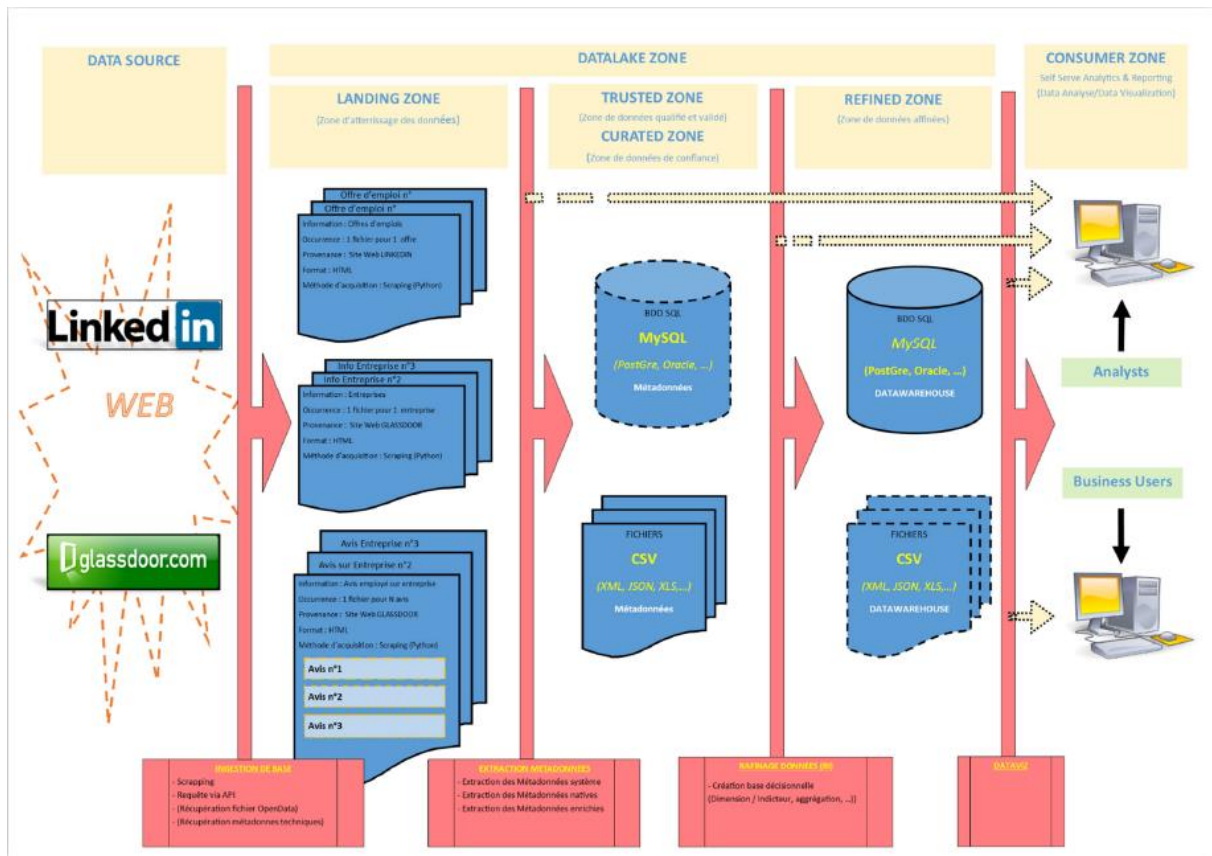
Les Data Lake sont structurés en 3 zones bien distinctes par convention. La « Landing Zone », ou « Zone d'atterrissage » des données, correspond à l'emplacement de stockage de toutes les données dans leur format initial. L'import des données se fait depuis diverses sources externes au Data Lake (directement du web par exemple, au format html, d'un système de gestion de base de données, de fichiers texte, tableurs etc...). Après import des données, cette zone ne va normalement jamais subir de modification car en cas de problème lors du traitement des données dans les couches supérieures, on veut être en mesure d'avoir accès et de retrouver n'importe quelle information utilisée dans son état brut.

La « Curated Zone », ou « Zone de données de confiance », regroupe les données importées dans le Data Lake après un premier traitement. On y stocke toutes les informations extraites des données brutes de la Landing Zone et notamment les différents types de métadonnées (système, natives et enrichies).

Enfin la « Refined Zone » correspond à une zone BI dans laquelle on va préparer les données pour analyse par le consommateur (« Consumer Zone » qui se trouve à l'extérieur du Data Lake dans la chaîne du traitement des données). Dans cette zone, on concevra et formalisera dans un document, un modèle de données pour notre Data Warehouse qui servira à la réalisation d'un reporting par exemple.

Le sujet du projet repose sur la conception d'un outil informatique ayant pour base un Data Lake et qui permettrait un accès en consultations, ainsi que des possibilités d'analyse, sur des offres d'emplois proposées dans le secteur de l'informatique. Les sources sont essentiellement des fichiers HTML contenant des offres d'emploi en provenance des sites web LinkedIn et Glassdoor.

Le schéma ci-dessous une vision globale du projet et de la structure du Data Lake que nous allons devoir mettre en place pour sa réalisation.



Les Objectifs de ce projet sont donc répartis en 3 axes principaux.

- Pour le niveau « Curated Zone / Trusted Zone », développer un ou plusieurs programmes en Python permettant de récupérer et stocker les métadonnées techniques et descriptives pour l'ensemble des fichiers HTML fournis et concernant respectivement les offres d'emplois (site web LinkedIn / Code : INFO-EMP), les entreprises employeurs (site web Glassdoor / Code : SOC) ainsi que les avis des salariés sur ces entreprises (site web Glassdoor / Code : AVI)
- Pour le niveau « Refined Zone », à l'aide d'un programme Python, construire et alimenter à partir des métadonnées stockées en « Curated Zone / Trusted Zone », l'entrepôt de données qui servira de source de données unique pour réaliser notre tableau de bord.
- Pour le niveau « Data visualisation », réaliser un ou plusieurs rapports avec les niveaux d'analyse statistique ciblés sur les entreprises et sur les emplois

II. Démarche à suivre

La structure de notre projet s'est faite en 4 étapes principales qui sont thématiques de la réalisation d'un Data Lake :

- L'Ingestion des données (récupération des métadonnées techniques)
- Les Métadonnées (extraction des métadonnées descriptive)
- Le Stockage des données (archivage et stockage)
- L'Accès aux données (préparation des données et utilisation)

0_SOURCE_WEB

1_LANDING_ZONE

2_CURATED_ZONE

3_REFINED_ZONE

4_CONSUMER_ZONE

Tout d'abord, regardons l'architecture générale de notre Data Lake. On a choisi de créer un répertoire dédié à chaque zone de notre environnement de travail afin de pouvoir séparer efficacement les étapes de traitement des données que nous allons ingérer au Data Lake. On obtient donc un répertoire « 0_Source_Web » contenant tous les fichiers HTML d'offres

d'emploi, d'informations et d'avis sur les sociétés dans leur état initial après téléchargement. Dans un second temps le répertoire « 1_Landing_Zone », dans lequel on copiera à l'identique tous les fichiers du répertoire Source_Web mais en prenant bien soin de créer un fichier de récupération des métadonnées descriptives telle que la date d'ingestion au Data Lake notamment (date à laquelle le transfert du répertoire 0 au répertoire 1 a été effectué) afin de pouvoir historiser les fichiers grâce à leur dernière date de modification. On a rajouté une deuxième arborescence dans le répertoire de la Landing_Zone afin de séparer les fichiers issus de Glassdoor et ceux de LinkedIn. Dans le dossier Glassdoor, on différenciera aussi les fichiers HTML traitant des informations sur les sociétés et ceux sur les avis laissés par les employés. Dans le répertoire « 2_Curated_Zone », on va stocker les informations parsées des fichiers que l'on avait ingéré dans la Landing_Zone directement (on ne repart pas des fichiers de la Source_Web). On ne gardera donc que les fichiers de données stockées dans un format CSV ou JSON et avec un modèle clé-valeur.

Enfin dans le répertoire « 3_Curated_Zone » seront stockées les données de la Curated_Zone après modification, recodification des variables, traitement des valeurs manquantes etc. Les données sont à présent prêtes à être exploitées et intégrées à un reporting pour analyse. Le tableau de bord est quant à lui construit à l'extérieur du Data Lake, après import de la base de données créée dans la Refined_Zone, et sera ensuite stocké et conservé dans le répertoire « 4_Consumer_Zone ».

A. Ingestion des métadonnées lors du passage en « 1_Landing_Zone »

Pour commencer, on va initialiser le fichier des métadonnées descriptives en ouvrant un fichier CSV en mode création et en le stockant dans le répertoire de la Landing_Zone. On y insère la première qui n'est autre que les descripteurs des 3 champs qui composeront chacune

des lignes qu'on y insérera à savoir la clé (l'identifiant), la colonne (le nom de la variable) et la valeur associée.

```
#-- Initialisation des colonnes du fichier de métadonnées
myFilePathName = myPathLog + "1_LANDING_ZONE/metadata.csv"

myFilePtr = open(myFilePathName, "a", encoding = "utf-8")

myListeDeLigneAEcrire = []
myListeDeLigneAEcrire.append('"cle_unique";"colonne";"valeur"')

for myLigneAEcrire in myListeDeLigneAEcrire:
    myFilePtr.write(myLigneAEcrire+"\n")

myFilePtr.close()
```

Ensuite on passe à la phase d'ingestion des données. Pour cela on va devoir créer une ou plusieurs fonctions permettant de collecter les métadonnées souhaitées. Ci-dessous la fonction permettant de récupérer la date d'ingestion ou de chargement des fichiers HTML dans le Data Lake. On veut connaître la date à laquelle le fichier a été implémenté au Data Lake donc on va stocker la date du jour à l'instant où le transfert est effectué.

```
#=====
#-- Fonction renvoyant <datetime_ingestion> du fichier
#=====
from datetime import datetime
def Get_datetime_ingestion_AVI():
    Result = str(datetime.now().strftime('%Y-%m-%d %H:%M:%S'))
    return(Result)

print(Get_datetime_ingestion_AVI())
```

Le morceau de code ci-dessous permet de sélectionner une liste de fichiers. L'objectif est de réaliser le premier transfert entre le répertoire Source et la Landing_Zone. Afin de pouvoir séparer les fichiers Glassdoor des fichiers LinkedIn, on applique une reconnaissance d'un pattern ou mot clé afin de faire un tri parmi tous les fichiers présents dans la Source et de ne sélectionner le fichier qu'en cas de « match », soit si le pattern est trouvé dans le nom du fichier parcouru. On stocke ensuite le nom du fichier dans une liste grâce à la fonction append().

```

=====
#-- Parcourir un dossier et stocker les noms de fichiers dans une liste
=====
#-- Import des bibliotheques
import sys, os, fnmatch

#-- Initialisation des variables
myListOfFile = []
myListOfFileTmp = []

myPathHtml = "C:/TD_DATA LAKE/DATALAKE/0_SOURCE_WEB/"

#-- Recupere les noms longs des fichiers dans le path
myListOfFileTmp = os.listdir(myPathHtml)

#-- Filtrer les fichiers concernés
myPattern = "*INFO-EMP*"

#-- Parcours de tous les fichiers trouvés
for myEntry in myListOfFileTmp :
    #-- On n'ajoute que les fichiers concernés
    if fnmatch.fnmatch(myEntry, myPattern)==True:
        myListOfFile.append(myEntry)

print(len(myListOfFileTmp))
#-- Affichage du résultat
for i in myListOfFile : print("Ligne : " + i)

print(len(myListOfFile))
for myFileName in myListOfFile: print(myPathHtml + " --> " + str(myFileName))

```

A présent, on crée une boucle qui va copier les fichiers sélectionnés auparavant et les coller un à un dans le répertoire souhaité (en l'occurrence 1_LANDING_ZONE > LINKEDIN) grâce à la fonction `shutil.copy()`. On ajoute ensuite à la boucle la récupération des métadonnées liées au fichier, notamment la date d'ingestion, la taille du fichier ainsi que son emplacement source pour connaître en toute circonstance sa provenance. Une fois ces informations collectées, on les stocke dans une liste qui sera ensuite retranscrite dans le fichier `metadata.csv` précédemment créé. On réitérera l'opération à 2 autres reprises, pour les fichiers GLASSDOOR-INFO et GLASSDOOR-AVIS.

```

#-----
#-- Copier des fichiers d'un répertoire vers un autre
#-----
import shutil

for i in myListOfFile: print(i)

myPathHtmlIn = "C:/TD_DATA LAKE/DATALAKE/0_SOURCE_WEB/"
myPathHtmlOut = "C:/TD_DATA LAKE/DATALAKE/1_LANDING_ZONE/LINKEDIN/EMP"

#-----
for monNomDeFichier in myListOfFile:
    print("Traitement :du fichier :", monNomDeFichier)
    monPathFile_Test_IN = myPathHtmlIn + "/" + monNomDeFichier
    monPathFile_Test_OUT = myPathHtmlOut + "/" + monNomDeFichier
    print(monPathFile_Test_IN)
    print(monPathFile_Test_OUT)
    shutil.copy(monPathFile_Test_IN, monPathFile_Test_OUT)

    metadata = os.stat(myPathHtml + monNomDeFichier)
    myFilePtr = open(myFilePathName, "a", encoding = "utf-8")

    myListeDeLigneAEcrire = []
    #enregistrement en clé valeur !

    myListeDeLigneAEcrire.append(monNomDeFichier[:14]+';datetime;'+Get_datetime_ingestion_AVI())
    myListeDeLigneAEcrire.append(monNomDeFichier[:14]+';taille;'+approximate_size(metadata.st_size))
    myListeDeLigneAEcrire.append(monNomDeFichier[:14]+';emplacement_source;'+myPathHtml)

    for myLigneAEcrire in myListeDeLigneAEcrire:
        myFilePtr.write(myLigneAEcrire+"\n")

myFilePtr.close()

```

B. Création de fonctions permettant la récupération de données

Une fois que nous avons collectés nos métadonnées et copier nos fichiers en « Landing zone », nous souhaitons récupérer les informations de nos fichiers html afin de les analyser. Pour cela nous créons des fonctions qui permettent ceci. Le package BeautifulSoup nous sera d'une grande utilité dans cette partie. Il va nous permettre de lire l'intégralité du code HTML présent dans les pages et de le stocker et d'en extraire les champs désirés en fonction de leur catégorisation/balises dans le code.

Nous avons donc créé des fonctions permettant la récupération des différents champs des pages Web grâce à ce package. Parfois, dans certains fichiers, l'information cherchée n'était pas trouvée par la fonction ce qui renvoyait une erreur. Nous avons donc créé une exception pour ces fonctions afin que la récupération des données ne soit pas stoppée à cause de quelques fichiers. Voici un exemple de la fonction récupérant les noms d'entreprise des avis Glassdoor :


```
def Get_nom_entreprise_AVI (Soup):
    myTest = []
    try: myTest = Soup.find_all('div', attrs = {"class":"header cell info"})[0].span.contents[0]
    except : print ("Pas le nom de l'entreprise sur cette page html"+ myFileName)
    if (myTest == []) :
        Result = 'NULL'
    else:
        Result = myTest
    return(Result)
```

C. Ajout des données récoltés dans un fichier csv

Une fois que nous avons créé l'ensemble des fonctions permettant de recueillir les informations dont nous avons besoin. Nous avons pu, de la même manière que pour le fichier des métadonnées, faire une boucle sur l'ensemble des fichiers permettant de prendre les informations dans chaque fichier en les ajoutant dans un fichier csv.

On a tout d'abord créé le fichier csv comme pour le fichier des métadonnées :

```
#-----
#-- Récupération des informations sur les emplois LinkedIn et création fichier csv
#-----

##-- Chargement de ma liste de fichiers pour avoir les fichiers "LinkedIn"

myListOfFile = []

#-- Path ou l'on souhaite mettre notre fichier csv
myPathLog = "C:/TD_DATALAKE/DATALAKE/2_CURATED_ZONE/LINKEDIN/EMP/"

#-- Path ou l'on va chercher nos fichiers html
myPathHtml = "C:/TD_DATALAKE/DATALAKE/1_LANDING_ZONE/LINKEDIN/EMP/"

#-- Recupere les noms longs des fichiers dans le path
myListOfFile = os.listdir(myPathHtml)
print(myListOfFile)

myFilePathName = myPathLog + "LinkedIn.csv"

#-- Ouverture du fichier en création (raz)
myFilePtr = open(myFilePathName, "w", encoding = "utf-8")

#-- Ouverture du fichier en ajout (modification)
myFilePtr = open(myFilePathName, "a", encoding = "utf-8")

ListLinkedIn = []
ListLinkedIn.append('"Clé";"colonne";"valeur"')
```

On a, ensuite, lancé la boucle de stockage de ces informations dans ce nouveau fichier CSV. Voici l'exemple pour les fichiers LinkedIn. Nous créons un fichier avec une clé (avec les 14 premiers caractères du fichier), un nom de colonne correspondant à l'information choisie ainsi que sa valeur correspondante :

```

#-- Boucle permettant de récupérer le poste, le nom de l'entreprise, la ville et la description
for myFileName in myListOfFile:
    myHTMLPathFileName = myPathHtml + myFileName
    myFilePtr = open(myFilePathName, "w", encoding = "utf-8")
    f = open(myHTMLPathFileName, "r", encoding="utf-8", errors='ignore')
    myHTMLContents = f.read()
    f.close()

    mySoup = BeautifulSoup(myHTMLContents, 'Lxml')

    ListLinkedin.append(myFileName[:14]+';"nom_poste";' +Get_libelle_emploi_EMP(mySoup))
    ListLinkedin.append(myFileName[:14]+';"nom_entreprise";' +Get_nom_entreprise_EMP(mySoup))
    ListLinkedin.append(myFileName[:14]+';"ville";' +Get_ville_emploi_EMP(mySoup))
    ListLinkedin.append(myFileName[:14]+';"description";' +Get_texte_emploi_EMP(mySoup))

#-- Test1 : Boucle d'écriture de chaque ligne de la liste d'éléments dans le fichier
for myLigneAEcrire in ListLinkedin:
    myFilePtr.write(myLigneAEcrire+"\n")

myFilePtr.close()

```

A noter que pour les « avis glassdoor » nous avons développé une boucle dans une autre afin de récupérer la note moyenne et le nom de l'entreprise dans la première boucle, puis dans la deuxième, tous les avis plus détaillés concernant cette entreprise.

Au final nous obtenons donc 4 fichiers csv :

- INFO GLASSDOOR :

Clv@	colonne	valeur
13564-INFO-	ville	Leverkusen (Allemagne)
13564-INFO-	nom_entrepi	Bayer
13564-INFO-	taille	Plus de 10 000 employv@s
13553-INFO-	ville	NULL
13553-INFO-	nom_entrepi	NULL
13553-INFO-	taille	NULL
13725-INFO-	ville	Plus de 10 000 employv@s
13725-INFO-	nom_entrepi	Plastic Omnium
13725-INFO-	taille	
13649-INFO-	ville	Uxbridge, Angleterre (Royaume-Uni)
13649-INFO-	nom_entrepi	Manpower
13649-INFO-	taille	De 5 001 v† 10 000 employv@s
13764-INFO-	ville	Villeurbanne
13764-INFO-	nom_entrepi	Artefacts Studio
13764-INFO-	taille	De 1 v† 50 employv@s
13736-INFO-	ville	Fullerton, CA (vâtats-Unis)
13736-INFO-	nom_entrepi	Thales-Raytheon Systems
13736-INFO-	taille	De 501 v† 1 000 employv@s
13693-INFO-	ville	New York, NY (vâtats-Unis)
13693-INFO-	nom_entrepi	Datadog
13693-INFO-	taille	De 1 001 v† 5 000 employv@s
13584-INFO-	ville	AsniV*res-sur-Seine
13584-INFO-	nom_entrepi	Smile Group
13584-INFO-	taille	De 1 001 v† 5 000 employv@s
13641-INFO-	ville	New York, NY (vâtats-Unis)

- LINKEDIN :

Civ@	colonne	valeur
13761-INFO-EMP	nom_poste	Ingv@nieur dv@cisionnel / BI (H/F) / Freelai
13761-INFO-EMP	nom_entrep	Freelance-info.fr
13761-INFO-EMP	ville	Aix-en-Provence, FR
13761-INFO-EMP	description	Choisir BeMore France pour votre carri@re, t
13768-INFO-EMP	nom_poste	Ingv@nieur dv@cisionnel F/H
13768-INFO-EMP	nom_entrep	Axande
13768-INFO-EMP	ville	Courbevoie, FR
13768-INFO-EMP	description	Vous avez dv@jv+ une premiv@re expv@rien
13554-INFO-EMP	nom_poste	DATA ANALYST JUNIOR (H/F)
13554-INFO-EMP	nom_entrep	Feu Vert
13554-INFO-EMP	ville	Vâcully, FR
13554-INFO-EMP	description	Bienvenue chez Feu Vert ! Nous possv@dons
13604-INFO-EMP	nom_poste	Adserving / Ingv@nieur BigData
13604-INFO-EMP	nom_entrep	Squadata
13604-INFO-EMP	ville	Lyon, FR
13604-INFO-EMP	description	Squadata recrute un Ingv@nieur Big Data ,Ài
13670-INFO-EMP	nom_poste	Stage - Data Scientist
13670-INFO-EMP	nom_entrep	SEPHORA

- Les avis Glassdoor (on a pour l'entreprise CACIB une note et 6 avis plus détaillées)

Civ@	colonne	valeur
13710-AVIS-	date ingestion	20/12/2022 16:46
13710-AVIS-	nom_entreprise	CACIB
13710-AVIS-	note moyenne	3.5
13710-AVIS-	numero_avis	1
13710-AVIS-	Type_employe	Ancien employv@ - Responsable informatique de l'API
13710-AVIS-	ville_employe	Paris
13710-AVIS-	commentaire	J'ai travaillv@ chez CACIB v+ temps plein moins d'un an
13710-AVIS-	numero_avis	2
13710-AVIS-	Type_employe	Employv@ actuel - Associate TMT
13710-AVIS-	ville_employe	Paris
13710-AVIS-	commentaire	Je travaille chez CACIB v+ temps plein depuis plus de 3 ans
13710-AVIS-	numero_avis	3
13710-AVIS-	Type_employe	Employv@ actuel - Employv@ anonyme
13710-AVIS-	commentaire	Je travaille chez CACIB v+ temps plein
13710-AVIS-	numero_avis	4
13710-AVIS-	Type_employe	Employv@ actuel - Employv@ anonyme
13710-AVIS-	commentaire	Je travaille chez CACIB v+ temps plein
13710-AVIS-	numero_avis	5
13710-AVIS-	Type_employe	Ancien employv@ - Employv@ anonyme
13710-AVIS-	commentaire	J'ai travaillv@ chez CACIB v+ temps plein
13710-AVIS-	numero_avis	6
13710-AVIS-	Type_employe	Employv@ actuel - Employv@ anonyme
13710-AVIS-	commentaire	Je travaille chez CACIB v+ temps plein

- Les métadonnées

cle_unique	colonne	valeur
13761-INFO-EMP	datetime	20/12/2022 16:43
13761-INFO-EMP	taille	128.8 KiB
13761-INFO-EMP	emplacemer	/Users/brossathugo/Desktop/DATALAKE/0_SOURCE_WEB/
13564-INFO-SOC	datetime	20/12/2022 16:43
13564-INFO-SOC	taille	223.4 KiB
13564-INFO-SOC	emplacemer	/Users/brossathugo/Desktop/DATALAKE/0_SOURCE_WEB/
13710-AVIS-SOC	datetime	20/12/2022 16:43
13710-AVIS-SOC	taille	352.4 KiB
13710-AVIS-SOC	emplacemer	/Users/brossathugo/Desktop/DATALAKE/0_SOURCE_WEB/
13646-AVIS-SOC	datetime	20/12/2022 16:43
13646-AVIS-SOC	taille	396.0 KiB
13646-AVIS-SOC	emplacemer	/Users/brossathugo/Desktop/DATALAKE/0_SOURCE_WEB/
13691-AVIS-SOC	datetime	20/12/2022 16:43
13691-AVIS-SOC	taille	344.2 KiB
13691-AVIS-SOC	emplacemer	/Users/brossathugo/Desktop/DATALAKE/0_SOURCE_WEB/
13768-INFO-EMP	datetime	20/12/2022 16:43
13768-INFO-EMP	taille	150.8 KiB
13768-INFO-EMP	emplacemer	/Users/brossathugo/Desktop/DATALAKE/0_SOURCE_WEB/
13553-INFO-SOC	datetime	20/12/2022 16:43
13553-INFO-SOC	taille	16.0 KiB
13553-INFO-SOC	emplacemer	/Users/brossathugo/Desktop/DATALAKE/0_SOURCE_WEB/

III. Exécution du code

Le code que nous avons développé en Python permet d'exécuter l'ensemble des étapes en une seule fois. En effet, celui-ci permet de prendre les fichiers sources et de les copier dans la landing zone. Ensuite le code permet de créer les 4 fichiers csv avec toutes les données en les plaçant dans la curated zone.

Voici son plan :

Récupération des fichiers dans « 0_Source_Web »

Création des fonctions pour récupérer les métadonnées

Copie des fichiers de « _Source_Web » vers « 1_Landing_Zone » en ingérant les métadonnées par la même occasion.

Pour LinkedIn

Pour les avis Glassdoor

Pour info Glassdoor

Création des fonctions qui récupèrent les données dans les fichiers html de la landing zone.

Pour info Glassdoor

Pour les avis Glassdoor

Pour LinkedIn

Ajout des données récupérer grâce aux fonctions dans des fichiers csv

Pour LinkedIn

Pour les avis Glassdoor (double boucle pour avoir les avis détaillés)

Pour info Glassdoor

Lors de l'exécution, la console affichera :

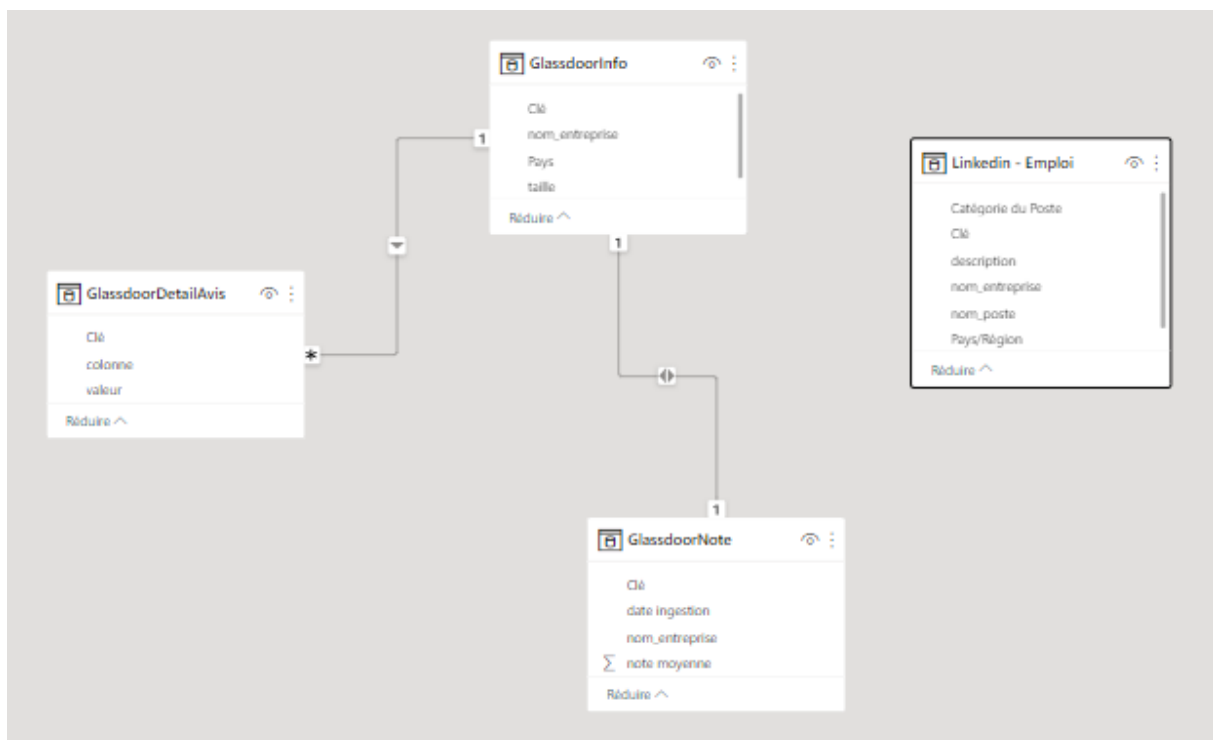
- La copie des fichiers de la source vers la landing zone
- Le nom des fichiers pour lesquels nous n'avons pas réussi à récupérer les données

IV. Data Visualisation

A. Nettoyage des données

Afin de pouvoir analyser nos données nous avons, sur Power Bi (l'outil de visualisation que nous avons choisi) transformé nos données en données sous forme classique car elles étaient difficilement analysable en clé-valeur dans la partie transformer les données.

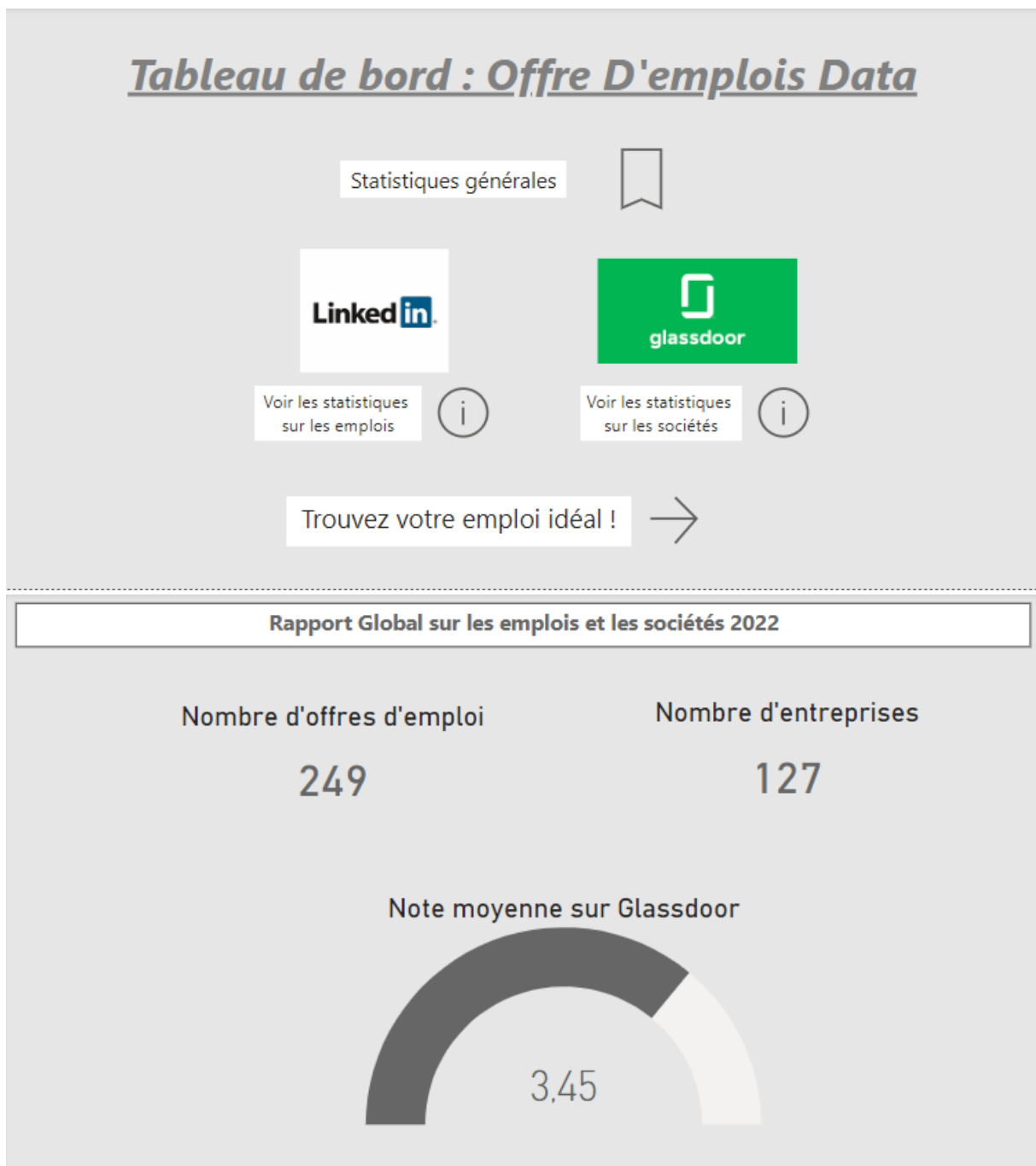
B. Modèle de données :

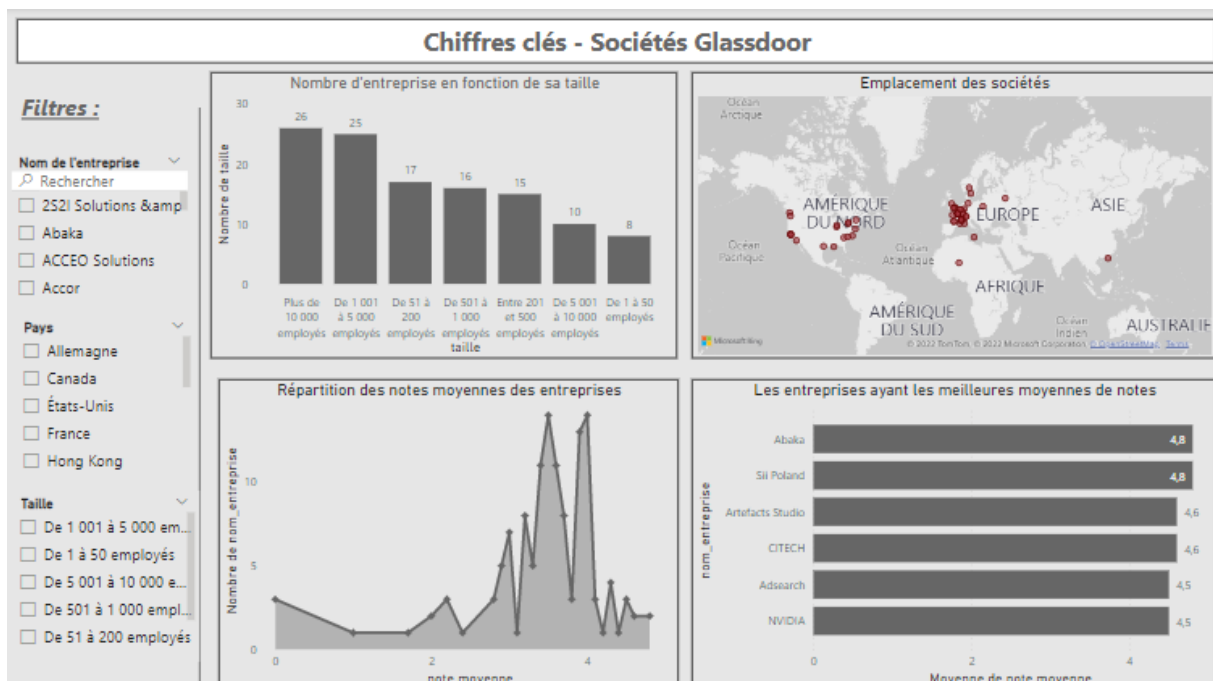
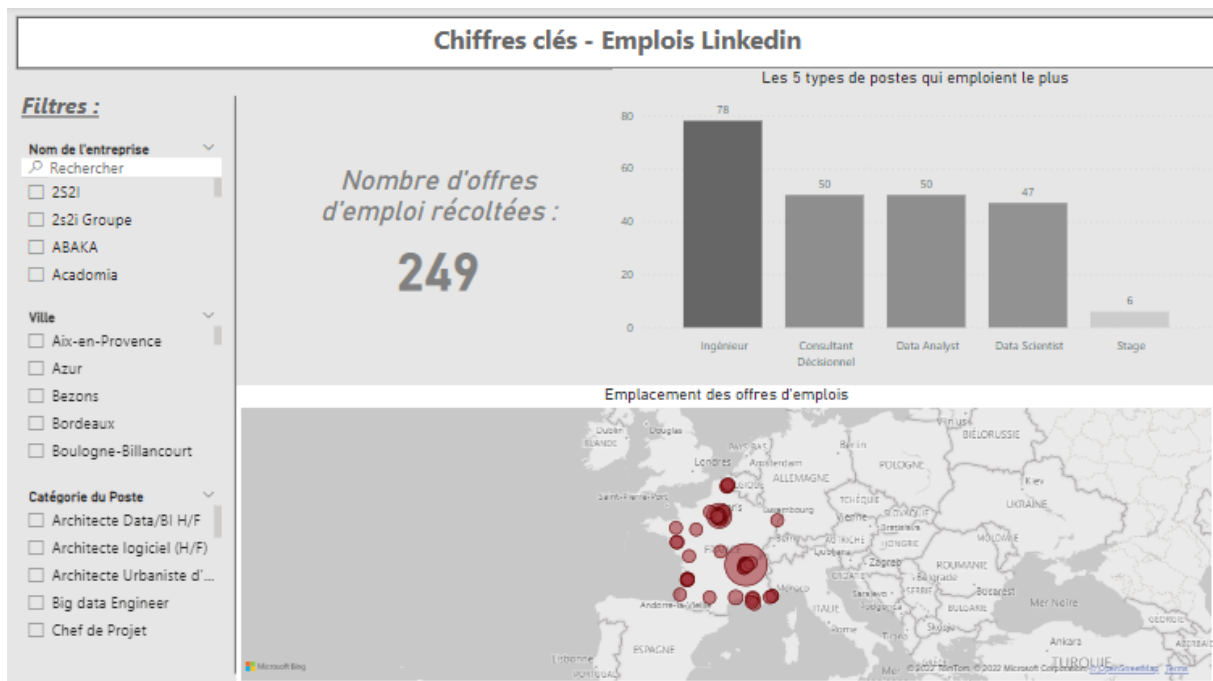


C. Aperçu du tableau de bord :

Le tableau de bord que nous avons réalisé permet d'avoir des informations à la fois sur les offres d'emploi de LinkedIn mais aussi sur des sociétés dont nous avons pu récupérer les informations sur Glassdoor. L'utilisateur peut donc voir si l'entreprise qui offre un emploi est de qualité par exemple.

De plus, il peut affiner ses recherches grâce à la dernière page de notre rapport qui permet de parcourir les différentes offres.





FIND YOUR JOB

Ville

Rechercher

☐ Aix-en-Provence

☐ Azur

☐ Bezons

☐ Bordeaux

☐ Boulogne-Billancourt

Catégorie du Poste

☐ Architecte Data/BI H/F

☐ Architecte logiciel (H/F)

☐ Architecte Urbaniste d'Entreprise

☐ Big data Engineer

☐ Chef de Projet

nom_entreprise

☐ 2S2I

☐ 2s2i Groupe

☐ ABAKA

☐ Acadomia

☐ Accor

Nom de l'entreprise	Nom du poste	Ville
2S2I	Ingénieur décisionnel F/H	Montpellier
2s2i Groupe	Ingénieur Décisionnel H/F / Freelance	Montpellier
ABAKA	NLP Data Scientist	Nice
Acadomia	INGÉNIEUR DBA DÉCISIONNEL (F/H)	Azur
Accor	Data Engineer (H/F)	Issy-les-Moulineaux
Accor	Développeur Talend Big Data H/F	Évry
ADD UP	Data Analyst	Paris
ADERLY / Invest in Lyon	Data scientist en imagerie médicale H/F	Lyon
Adsearch	Data Analyst	Limonest
Adsearch	Ingénieur Développement Java Python	Lyon
Adsearch SAS	DATA ANALYST H/F	Limonest
Adthink	OFFRE D'EMPLOI (H/F) : MEDIA TRADER et DATA ANALYST	Lyon
Adthink	OFFRE DE STAGE (H/F) : MACHINE LEARNING - DATA SCIENTIST	Lyon
AKKA Tech	Consultant Décisionnel F/H	Paris
aldecis	Consultant décisionnel BI F/H	Paris
Alithya	Ingénieur développeur Big Data	Paris
Alteca	Consultant décisionnel BI (F/H)	Paris
Alteca	Stage - Ingénieur d'études décisionnel H/F	Paris
ALTIVIEW	Consultant Décisionnel	Lyon
Altran	Data Scientist	Paris La Défense
Altran	INGÉNIEUR EN DEVELOPPEMENT BIG DATA H/F	Sophia Antipolis
Amilone	Consultant décisionnel ETL/ESB F/H	Villeurbanne
AnotherBrain	Ingénieur.e IA	Paris
Approach People Recruitment	Super opportunité pour Data Engineer – Pure player Foodtech – H/F	Paris