Université Lumière Lyon 2 — Institut de la communication

Novembre 2025

Gestion de données massives

Etude et réalisation du processus complet de création d'un Data Lakehouse

Melissa BOULOUFA

Zyad NAISSALI

Jérémy PELLISSIER

Master 2 BI&A

Supervisé par : Eric KLOECKLE

Sommaire

Introduction

Description du projet

Ce projet de "*Gestion de données massives*" consiste à la mise en œuvre d'un pipeline complet de traitement et d'ingestion de données, allant de la collecte de données brutes de type HTML, jusqu'à la visualisation avec un outil BI (Business Intelligence), et en passant par leur traitement pour chacune des zones du Data Lakehouse.
L'objectif est de comprendre et d'appliquer les principales étapes techniques et méthodologiques de la chaîne de traitement des données dans sa conception et sa réalisation.

GitHub : https://github.com/blfmelissa/projet-datalake.git

Présentation des source de données

Les sources sont des fichiers HTML issus du web : Offres d'emploi (INFO-EMP), Informations sur les entreprises (INFO-SOC) et Avis donnés par des employés sur ces entreprises (AVIS-SOC). Ces fichiers HTML proviennent des sites professionnels d'emplois GLASSDOOR et LINKEDIN . Ces données ne seront jamais modifiées.

Conception et implémentation du Data Lake

Architecture et logique des zones

Le projet s'appuie sur une architecture de Data Lake organisée en trois zones distinctes : la Landing Zone, la Curated Zone et la Refined Zone. Cette structuration garantit une amélioration progressive de la qualité des données ainsi qu'une séparation claire des responsabilités à chaque étape du traitement.

Progression du flux de données

Le pipeline de traitement suit une progression claire. Chaque zone remplit un rôle spécifique et transmet une donnée progressivement enrichie à la zone suivante :

1. Landing Zone :
   Point d'entrée du flux, la Landing Zone reçoit les données brutes issues du 0_Source_Web. Elle assure leur ingestion, leur tri et leur classification selon leur typologie (EMP, SOC, AVI), tout en préservant leur état d'origine. Elle constitue la première couche de gouvernance et sert de socle à la traçabilité technique.

2. Curated Zone :
   La Curated Zone transforme les données de la Landing Zone via des opérations de parsing et de standardisation. Son rôle est d'extraire et structurer les éléments clés pour produire les métadonnées descriptives, généralement sous forme de triplets (clé_unique ; colonne ; valeur). Cette étape convertit la donnée brute en information

métier exploitable.

3. Production Zone :
   Dernière étape du pipeline, la Production Zone (ou Refined Zone) exploite les données structurées de la Curated Zone pour alimenter le modèle décisionnel (Data Lakehouse). Elle est optimisée pour l'analyse, la restitution et l'alimentation des outils de Business Intelligence.

Gestion des métadonnées

La traçabilité et l'exploitabilité des données reposent sur deux catégories de métadonnées, chacune stockée séparément pour répondre à des besoins distincts :

- Métadonnées Techniques :
  Générées dès l'ingestion en Landing Zone (date, source, chemin du fichier, typologie), elles sont centralisées dans le répertoire 99_Metadata. Elles constituent un registre d'audit essentiel pour la gouvernance, la supervision et la conformité.

- Métadonnées Descriptives :
  Issues des opérations d'extraction et de parsing réalisées en Curated Zone, elles regroupent les informations métier (ex : nom d'entreprise, libellé d'emploi, notation). Ces données structurées forment la base du futur modèle décisionnel dans la Refined Zone. Elles sont stockées dans la Curated Zone.

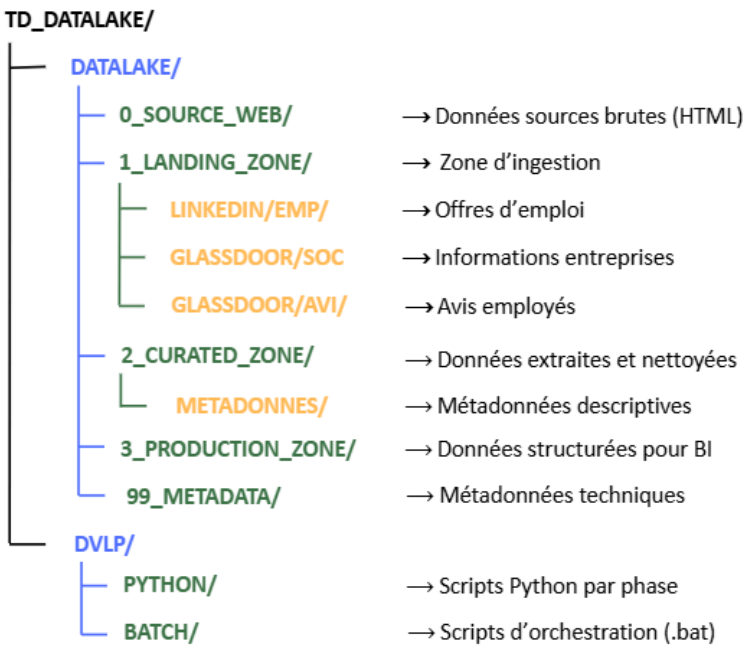


Schéma descriptif de l'architecture technique du projet

Mise en place de la Landing Zone : Collecte et stockage du brut

La première étape de l'implémentation du Data Lake a consisté à construire la Landing Zone. Conforme à ses principes fondamentaux, son objectif principal est de recevoir, conserver et d'organiser les données brutes issues du répertoire source (0_SOURCE_WEB) tout en les conservant strictement dans leur format original (HTML).

Processus de copie et de classification (Script Python Phase 1)

L'opération a été entièrement automatisée par le premier script Python. Il remplit 2 fonctions essentielles : transférer les fichiers HTML bruts vers la Landing Zone et les classifier selon leur typologie.

1. Maintien de l'intégrité **:** La librairie shutil et la fonction shutil.copy ont été utilisées pour garantir le transfert des fichiers sans aucune altération. Ce choix répond directement au principe fondateur de la Landing Zone : conserver une trace fidèle et immuable des données sources.
2. Logique de tri **:** La classification repose sur la nomenclature des noms de fichiers, permettant d'identifier l'origine et la nature du contenu (EMP, SOC, AVI). Le script distribue ainsi les fichiers dans des sous-répertoires dédiés, assurant une organisation claire pour les traitements futurs :
    - Les fichiers contenant la chaîne "LINKEDIN" ont été dirigés vers LINKEDIN/EMP (offres d'emplois).
    - Les fichiers contenant "AVIS-SOC-GLASSDOOR" ont été dirigés vers GLASSDOOR/AVI (avis salarié).
    - Les fichiers contenant "INFO-SOC-GLASSDOOR" ont été dirigés vers GLASSDOOR/SOC (informations entreprise).

Cette première phase pose les fondations de la traçabilité des sources et de la structuration physique du Data Lake, garantissant un point d'entrée fiable et maîtrisé pour l'ensemble du pipeline.

Curated Zone : Extraction des métadonnées techniques et descriptives

Cette étape joue un rôle central dans la structuration du Data Lake. Elle repose sur une double extraction, assurée par deux scripts distincts, permettant de couvrir à la fois les besoins de gouvernance (métadonnées techniques) et ceux liés à la valeur métier (métadonnées descriptives).

Génération des métadonnées techniques (Gouvernance)

La production des métadonnées techniques est directement intégrée au Script Python Phase 1, utilisé lors de l'ingestion et de la classification des fichiers en Landing Zone.

Rôle : Ces métadonnées sont générées au moment exact où les fichiers bruts sont copiés et triés en Landing Zone. Elles constituent la base de la traçabilité du Data Lake et garantissent la possibilité d'un audit.

Stockage : Elles sont centralisées dans le répertoire dédié 99_Metadata, sous la forme du fichier metadata_technique.csv.

Contenu :
Ce fichier regroupe :

- Une clé d'identification unique,
- Le nom du fichier HTML,
- Sa provenance,
- Sa localisation finale en Landing Zone,
- La date et l'heure d'ingestion.

Ces informations forment la base de la gouvernance indispensable pour tous les traitements ultérieurs.

Extraction et standardisation des métadonnées descriptives

Le cœur fonctionnel de la Curated Zone repose sur l'extraction des informations métier, réalisée via le Script Python Phase 2.

Liaison à la Source : Le script commence par lire metadata_technique.csv, qui lui fournit la liste exhaustive des fichiers à analyser ainsi que leur localisation en Landing Zone. Cette étape assure un lien direct et systématique entre la donnée brute et la donnée enrichie.

Parsing ciblé : L'analyse syntaxique est effectuée à l'aide de la librairie BeautifulSoup. Des fonctions d'extraction spécialisées ont été construites pour chaque type d'objet (EMP, SOC, AVI) afin d'isoler uniquement les informations pertinentes :

- Nom de l'entreprise,
- Localisation de l'emploi,
- Description de l'organisation,
- Notes et avis des salariés, etc.

Standardisation du format : Les données extraites sont ensuite converties dans un format long structuré (clé / colonne / valeur) avant d'être enregistrées dans metadata_descriptive.csv.

Ce format facilite la gestion de structures complexes (ex. : plusieurs avis pour une même entité) tout en garantissant la conservation du lien avec la clé unique issue de la Landing Zone.

Production Zone : ETL et préparation des données

Cette dernière étape réalisée via le script Python Phase 3, assure la pré-transformation structurante des données. Elle prend les métadonnées descriptives produites dans la Currated Zone pour les convertir en trois tables relationnelles prêtes à être intégrées dans la Refined zone.

Transformation et pivotage des données

Le script effectue un travail de conversion, consistant à passer du format (clé ; colonne ; valeur) vers des tables relationnelles.

Pivotage des données :
Le fichier metadata_descriptive.csv est lu puis regroupé par clé unique. Pour chaque entité, les paires colonne/valeur sont pivotées afin de former un enregistrement complet, résultant à une structure tabulaire classique qui est adaptée au modèle relationnel.

Attribution d'identifiants uniques :
Pour préparer les futures jointures et assurer la cohérence du modèle, des identifiants uniques sont générés pour chaque entité : idsociete, idemploi, idavis.
Ces identifiants constituent la base des relations dans la tables finales

Résolution de l'unification des entités et standardisation avancée

Le défi principal de cette étape consiste à lier les trois sources : SOC (Entreprise), AVI (Avis) et EMP (Emploi) en s'appuyant sur le nom de l'entreprise. Ce travail de rapprochement conceptuel constitue l'élément central du linked data implémenté par ce script.

Normalisation des clés :
Une fonction dédiée standardise les noms d'entreprise (casse, accents, ponctuation), afin de produire une clé normalisée unique. (ex : "Cegid, inc" → "cedig inc") Cette normalisation garantit un regroupement fiable des entités issues de sources hétérogènes.

Dédoublonnage et gestion des clés étrangères :
À partir de la clé normalisée, le script :

- réutilise l'idsociete si l'entreprise existe déjà
- créer un nouvel identifiant si elle est nouvelle

Cela assure la cohérence et l'intégrité entre les différentes tables du modèle.

Parsing avancé des localisations :
La fonction parse_location décompos les chaînes de localisation en champs structurées : ville, région,code postal, pays.
Cette granularité facilitera les analyses géographiques dans la Production Zone et les outils BI.

Exportation des fichiers finalisée

Le processus se conclut par l'exportation des données structurées dans trois fichiers CSV : sociétés.csv, emplois.csv et avis.csv. Ces fichiers sont normalisés et reliés via la clé idsociete, constituent l'état final des données avant leur chargement et transformation final dans la Production Zone.

Résumé des 3 phases :

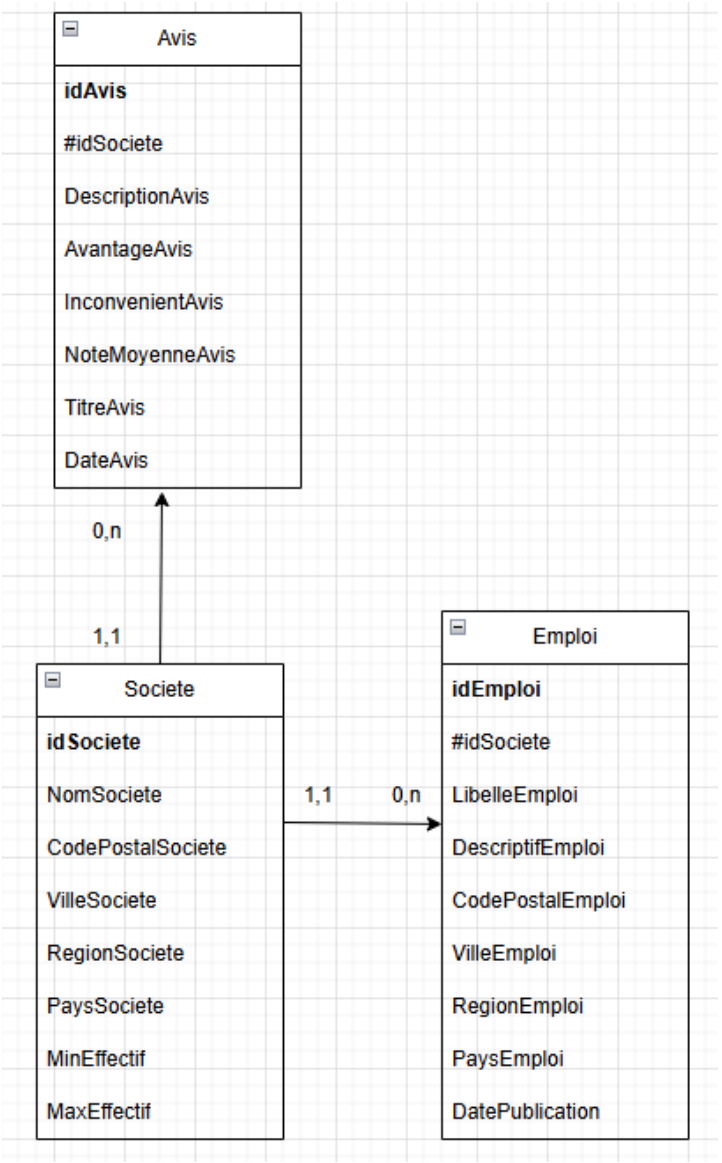
Phase	Nom	Description	Entrée	Sortie
1	Ingestion	Copie fichiers HTML vers Landing Zone	0_SOURCE_WEB	1_LANDING_ZONE + metadata_technique.csv
2	Extraction	Parse HTML et extrait données descriptives	1_LANDING_ZONE	2_CURATED_ZONE/metadata_descriptive.csv
3	Transformation ETL	Transforme en tables relationnelles	2_CURATED_ZONE	3_PRODUCTION_ZONE

Architecture et modélisation des données

Afin d'assurer un suivi des données cohérent, notre architecture de données est conçue en deux phases principales :

- Un modèle de données relationnel : pour l'insertion et l'intégrité des données transactionnelles.
- Un Data Lakehouse : pour l'analyse et le reporting.

Modèle de données relationnel



Le premier modèle est le modèle relationnel, il s'articule autour de trois tables principales, reflétant les aspects clés du projet : Société, Emploi et Avis.

On retrouve d'abord **la table Société**. Elle contient les informations de base de l'entreprise, notamment son nom, son code postal, sa ville, sa région, son pays et l'intervalle de l'effectif avec le champ min et max.

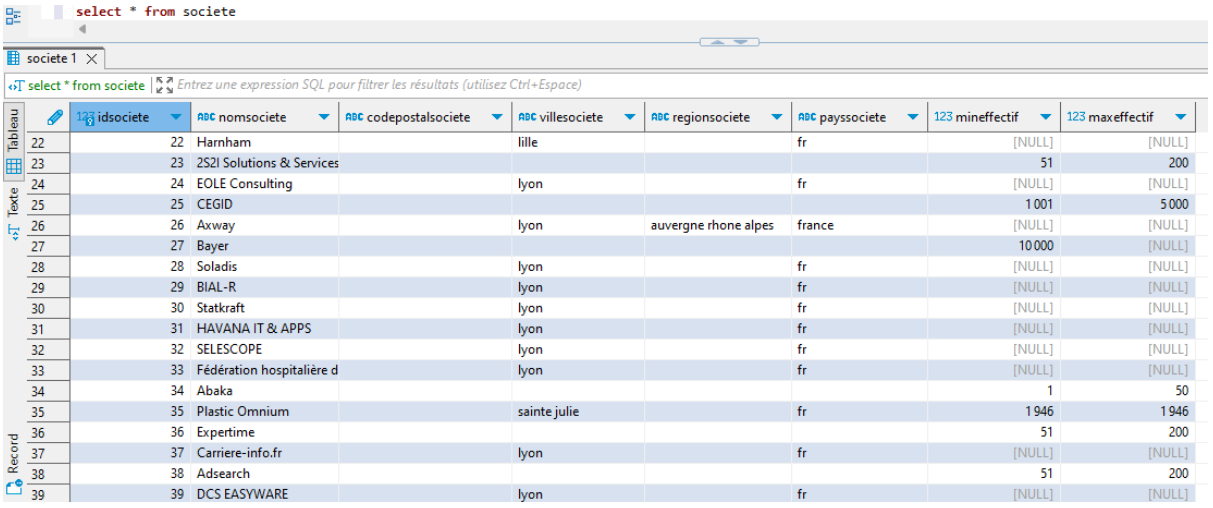
Ensuite, nous avons deux tables qui sont directement liées à la table Société :
La table Emploi : Cette table stocke tous les détails relatifs aux offres d'emploi. Pour chaque emploi, on y retrouve son ID, son libellé, son descriptif, ainsi que les informations de localisation comme le code postal, la ville, la région, le pays et la date de publication de l'emploi. De plus, elle contient l'ID Société, qui sert de clé étrangère pour identifier l'entreprise spécifique à laquelle cet emploi est lié.

La table Avis : Cette table est dédiée à la collecte des avis des utilisateurs. Pour chaque avis, on y retrouve son ID, le titre de l'avis, la description de l'avis, les avantages, les inconvénients, la note moyenne attribuée et la date de l'avis. Elle contient également l'ID Société, permettant de lier l'avis à l'entreprise concernée.

L'intégration des données dans le modèle relationnel s'effectue à partir des fichiers CSV, produits en amont par des scripts Python.

Concrètement, ces fichiers CSV sont ensuite importés dans une base de données PostgreSQL en utilisant l'outil d'administration DBeaver pour le chargement final.

Voici un exemple pour la table Société :



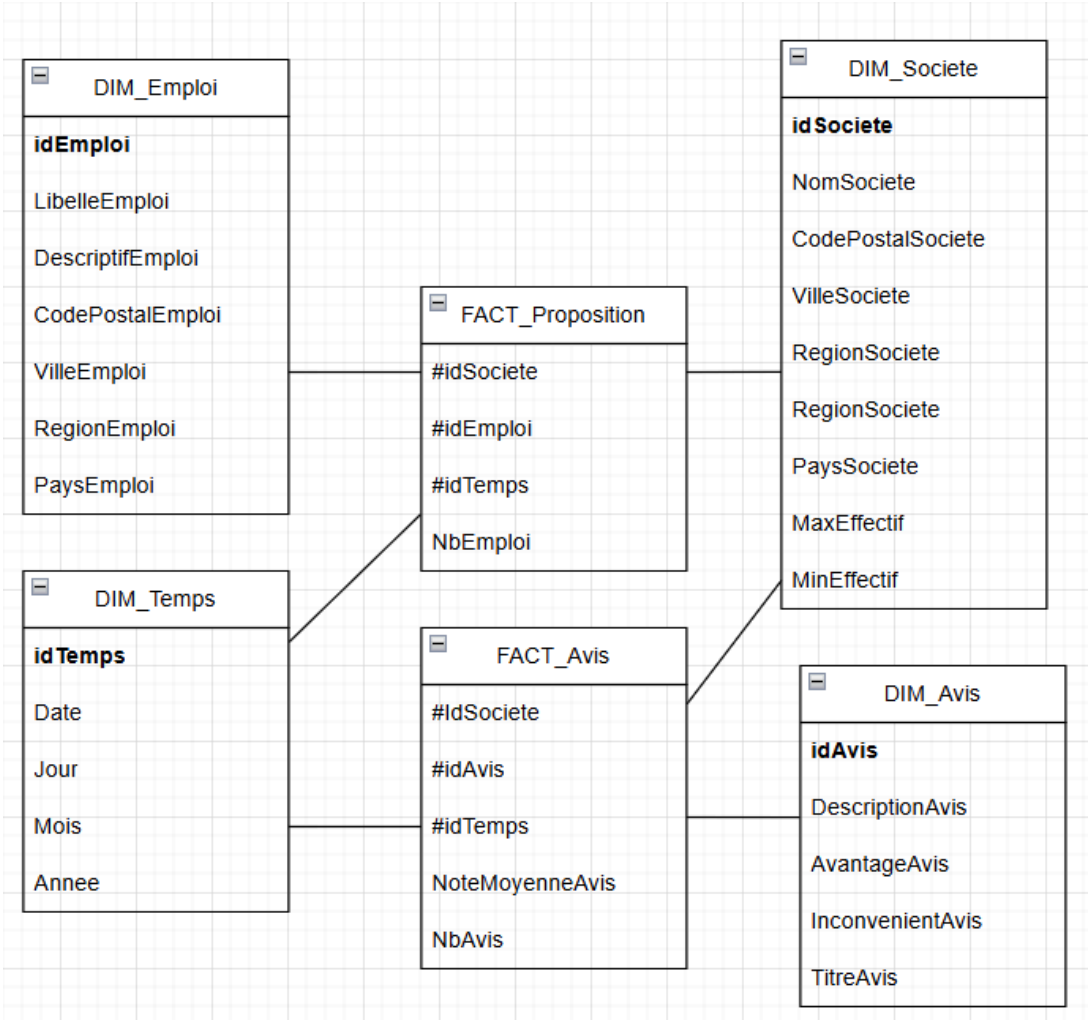
```
select * from societe
```

societe 1

select * from societe | Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

	idsociete	nomsociete	codepostalsociete	villesociete	regionsociete	payssociete	mineffectif	maxeffectif
22	22	Harnham		lille		fr	[NULL]	[NULL]
23	23	2S2I Solutions & Services					51	200
24	24	EOLE Consulting		lyon		fr	[NULL]	[NULL]
25	25	CEGID					1 001	5 000
26	26	Axway		lyon	auvergne rhone alpes	france	[NULL]	[NULL]
27	27	Bayer					10 000	[NULL]
28	28	Soladis		lyon		fr	[NULL]	[NULL]
29	29	BIAL-R		lyon		fr	[NULL]	[NULL]
30	30	Statkraft		lyon		fr	[NULL]	[NULL]
31	31	HAVANA IT & APPS		lyon		fr	[NULL]	[NULL]
32	32	SELESCOPE		lyon		fr	[NULL]	[NULL]
33	33	Fédération hospitalière d		lyon		fr	[NULL]	[NULL]
34	34	Abaka					1	50
35	35	Plastic Omnium		sainte julie		fr	1 946	1 946
36	36	Expertime					51	200
37	37	Carriere-info.fr		lyon		fr	[NULL]	[NULL]
38	38	Adsearch					51	200
39	39	DCS EASYWARE		lyon		fr	[NULL]	[NULL]

Data Lakehouse



Notre Data Lakehouse (DLH) est conçu selon un modèle en constellation. Cette architecture comprend quatres tables de dimension et deux tables de faits, permettant une analyse multidimensionnelle cohérente.

Les tables présentes dans notre DLH reprennent les données des tables de notre modèle relationnel.

Les axes d'analyse sont les suivants :

- La dimension Emploi a pour rôle de détailler l'offre d'emploi. Ses attributs clés sont : l'id de l'emploi, le libellé, le descriptif, le code postal, la ville, la région, et le pays.
- La dimension Société contient les informations de base de l'entreprise. Ses attributs clés sont : l'id de la société, le nom, le code postal, la ville, la région, le pays et l'intervalle de l'effectif avec le champ min et max.
- La dimension Avis fournit les détails des avis utilisateurs. Ses attributs clés sont : l'id de l'avis, la description, les avantages, les inconvénients, et le titre de l'avis.
- La dimension Temps contient les informations sur la date de publication de l'emploi et la date de l'avis, elles sont distinguées avec un idtemps et ont pour champ la date, le jour, le mois, et l'année.

La séparation en deux tables de faits distinctes est justifiée par la nécessité de lier des indicateurs différents à des combinaisons de dimensions spécifiques :

- Le fait Proposition a pour rôle de mesurer le volume d'activité lié aux offres d'emploi. Elle est liée aux dimensions Société, Emploi et Temps et son indicateur clé est le nombre d'emploi, qui représente le nombre d'offres d'emploi enregistrées.

- Le fait Avis est conçu pour l'analyse des avis utilisateurs. Elle est reliée aux dimensions Société, Avis, et Temps, et ses KPIs sont la note moyenne des avis et le nombre d'avis.

L'alimentation du Data Lakehouse s'effectue à partir de la base de données du modèle relationnel préalablement créé. Pour ce faire, nous utilisons des procédures stockées SQL. Ces procédures permettent d'extraire les données de la source relationnelle et de les charger dans le modèle du Data Lakehouse, en alimentant à la fois les tables de dimensions et les tables de faits.

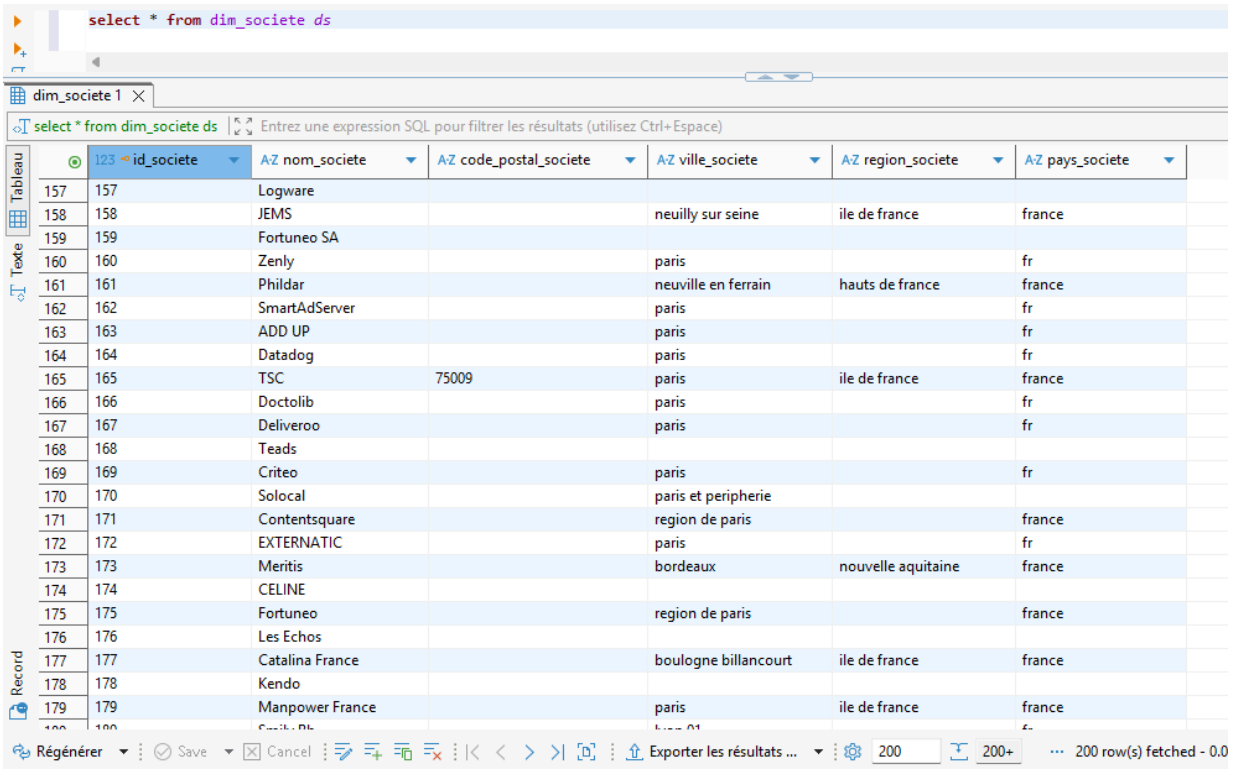
Cette procédure est un exemple illustrant le processus d'intégration : elle sélectionne les champs de la table source Société (issue du modèle relationnel) pour insérer les données dans la dimension Société.

```sql
CREATE OR REPLACE PROCEDURE sp_load_dim_societe()
LANGUAGE plpgsql
AS $$
BEGIN

    TRUNCATE TABLE dim_Societe CASCADE;
    INSERT INTO dim_Societe (
        idSociete,
        nomSociete,
        CodePostalSociete,
        VilleSociete,
        RegionSociete,
        PaysSociete,
        MinEffectif,
        MaxEffectif
    )

    SELECT
        idSociete,
        NomSociete,
        CodePostalSociete,
        VilleSociete,
        RegionSociete,
        PaysSociete,
        MinEffectif,
        MaxEffectif
    FROM Societe;
END;
$$;
```

Voici le résultat pour la dimension Société :



```
select * from dim_societe ds
```

dim_societe 1

select * from dim_societe ds — Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

	id_societe	nom_societe	code_postal_societe	ville_societe	region_societe	pays_societe
157	157	Logware				
158	158	JEMS		neuilly sur seine	ile de france	france
159	159	Fortuneo SA				
160	160	Zenly		paris		fr
161	161	Phildar		neuville en ferrain	hauts de france	france
162	162	SmartAdServer		paris		fr
163	163	ADD UP		paris		fr
164	164	Datadog		paris		fr
165	165	TSC	75009	paris	ile de france	france
166	166	Doctolib		paris		fr
167	167	Deliveroo		paris		fr
168	168	Teads				
169	169	Criteo		paris		fr
170	170	Solocal		paris et peripherie		
171	171	Contentsquare		region de paris		france
172	172	EXTERNATIC		paris		fr
173	173	Meritis		bordeaux	nouvelle aquitaine	france
174	174	CELINE				
175	175	Fortuneo		region de paris		france
176	176	Les Echos				
177	177	Catalina France		boulogne billancourt	ile de france	france
178	178	Kendo				
179	179	Manpower France		paris	ile de france	france

Régénérer | Save | Cancel | Exporter les résultats ... | 200 | 200+ | 200 row(s) fetched - 0.0

Pour plus de détails, veuillez consulter l'intégralité des scripts SQL concernant la création des tables et des procédures stockées associées. `(3_PRODUCTION_ZONE/seed_data.sql)`

Par ailleurs, il est notable que l'étape d'ETL n'est pas encore complètement finalisée (comme l'indique l'absence de transformation sur des champs tels que pays_societe). Les ajustements et les modifications nécessaires seront effectués lors de la prochaine phase, directement dans QLIK.

Analyse et Data Visualisation

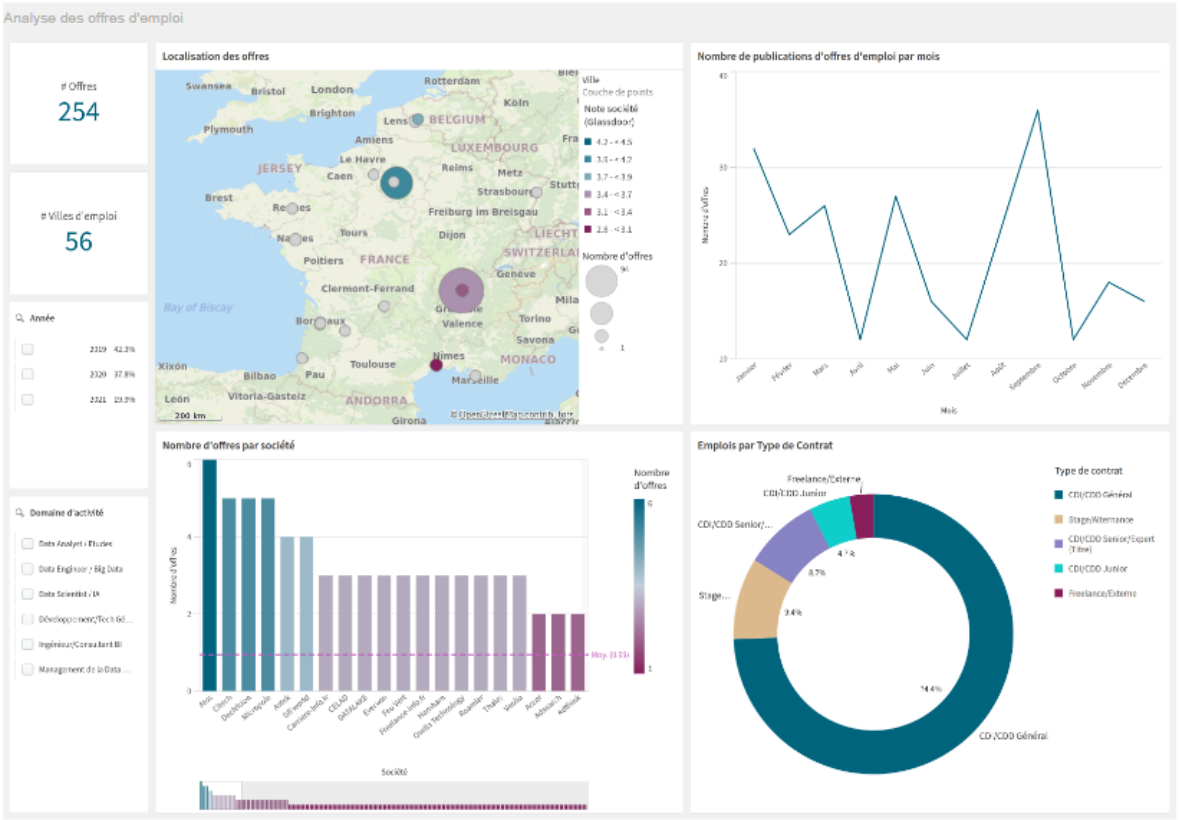
Cette étape est en réalité la *Phase 4* du pipeline, mais nous avons choisi d'effectuer les agrégations et calculs en temps réel dans le dashboard, plutôt que d'écrire un script Python pour chacune des analyses. Cela offre une plus grande flexibilité d'analyse.

Nous avons choisi l'outil BI Qlik Sense pour la data visualisation.
Ce choix est dû au fait que Qlik Sense est un outil que nous n'avions jamais utilisé auparavant. Nous connaissions Power BI et Tableau Software, mais pas Qlik. Pourtant, ce dernier, très puissant, est l'un des outils les plus utilisés en BI.
La prise en main de Qlik Sense a nécessité quelques heures de pratique.

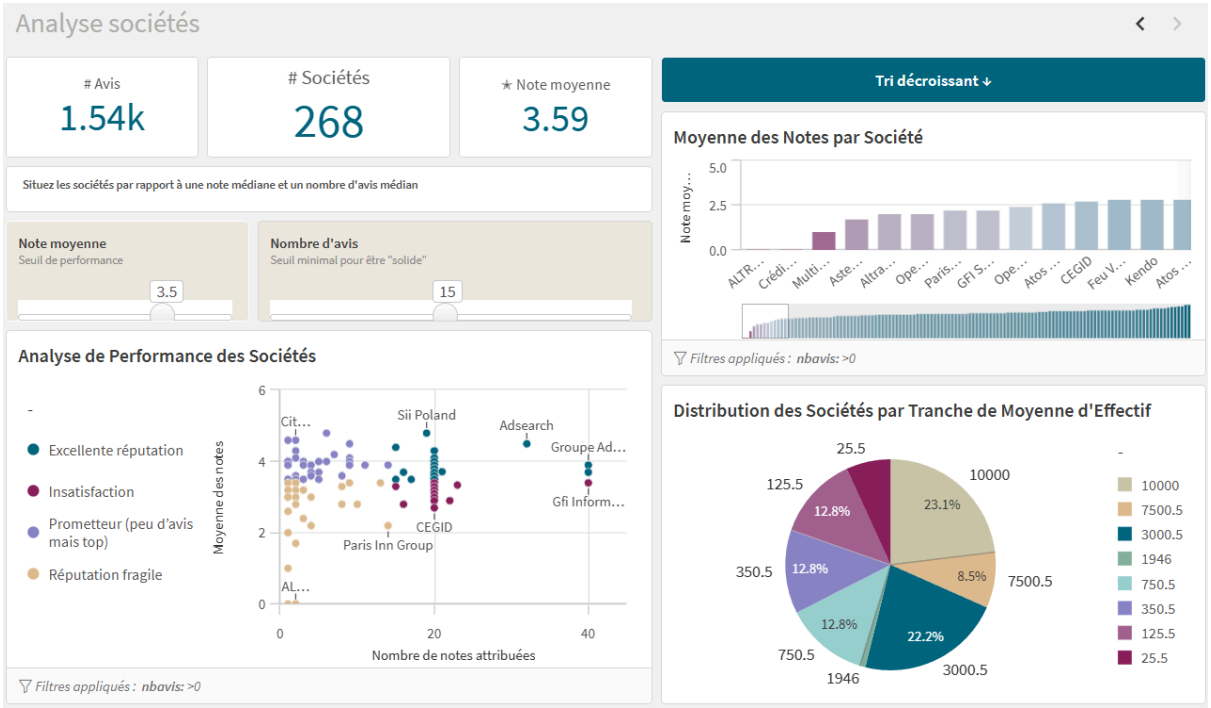
Vue d’ensemble du tableau de bord (dashboard)

Ce tableau de bord de 3 pages offre une vue d'ensemble de 254 offres d'emploi en France, provenant de LinkedIn et Glassdoor. Il analyse ces offres sous plusieurs angles : le volume, la localisation, le type d'emploi et la réputation des entreprises qui recrutent.
Lien vers le dashboard interactif → Lien.

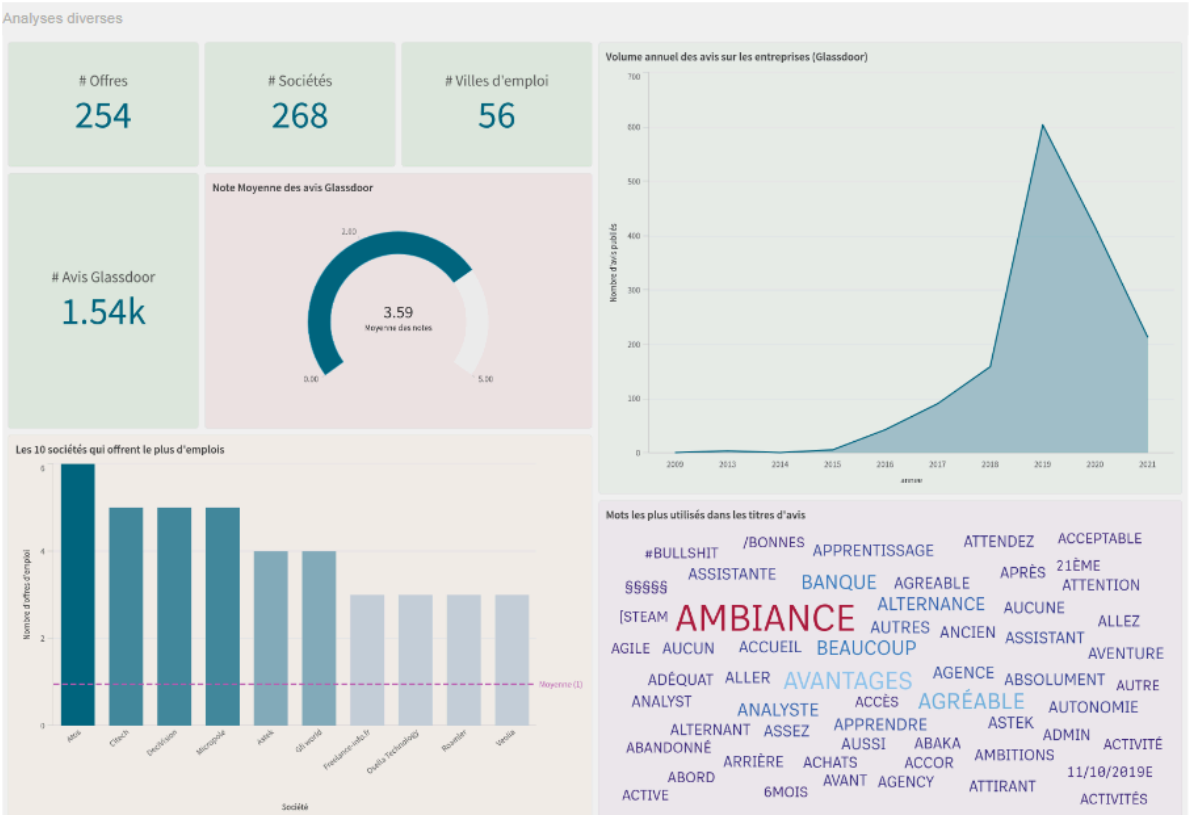
Page 1 : Analyse des Offres d’Emploi



Page 2 : Analyse des Sociétés



Page 3 : Analyses Diverses



Explication de quelques graphiques :

Indicateurs KPI

Emplois : notre analyse porte sur 254 offres d'emploi
Note moyenne : la moyenne des notes de nos sociétés sur Glassdoor est de 3.59/5
Avis : 1540 avis ont été collectés pour les entreprises
Sociétés : 268 sociétés sont présentes dans nos données
Villes : les offres d'emplois sont distribuées sur 56 différentes villes françaises

Analyse sur la page n°1 : Analyse des Offres d'Emploi

Localisation des offres

Cette carte montre où se situent les différentes offres d'emploi en France. La taille des cercles indique le nombre d'offres, et les couleurs indiquent les moyennes des notes (allant de rouge (mauvaises) à bleu (bonnes)). On voit une forte concentration à Paris et à Lyon.

- Certains points sont gris car les sociétés n'ont pas eu d'avis.
- Toutes les sociétés n'y sont pas présentes car certaines avaient leur champ "ville" vide.

Importance : on voit immédiatement si sa région est dynamique et si les entreprises y sont de qualité.

Emplois par Type de Contrat

Nous avons ajouté deux colonnes à la table *dim_emploi* : "Type de contrat" et "Catégorie d'activité". Pour faire ça, nous avons dû écrire une requête qui cherche des mots clés très spécifiques dans les champs "libellé emploi" et "descriptif emploi" afin d'automatiser au mieux le processus. Le résultat est très satisfaisant, et les emplois sont bien catégorisés.

Analyses sur la page n°2 : Analyse des Sociétés

Analyse de performance des sociétés

C'est l'analyse la plus complexe. Elle croise deux données : le nombre de notes et leurs moyennes, pour chaque société.
Les société sont ensuite catégorisées en 4 :

- Excellente réputation : beaucoup d'avis, bonne moyenne
- Prometteur : peu d'avis, bonne moyenne
- Réputation fragile : peu de notes, mauvaise moyenne
- Insatisfaction : beaucoup de notes, mauvaise moyenne

Ces chiffres, >15 pour "beaucoup de notes" et >3.5 pour "bonne moyenne", sont arbitraires et pas forcément idéaux. Nous avons donc ajouté 2 curseurs pour que ces deux valeurs soient dynamiques et définies par l'utilisateur.
Cette analyse permet d'évaluer la fiabilité des entreprises. Une moyenne seule peut être trompeuse, ce graphique aide alors à distinguer les bonnes entreprises des mauvaises.

Moyenne des notes par société

Ce graphique montre la moyenne des notes par société. Toutes les sociétés ayant au moins 10 avis sur Glassdoor y sont présentes. Comme il y en a beaucoup, nous avons ajouté un bouton pour changer l'ordre de tri et avoir plus rapidement les informations qui nous intéressent.

L'importance de cette analyse est qu'elle fournit un classement clair des entreprises.

Analyse sur la Page n°3 : Analyses Diverses

Les 10 sociétés qui offrent le plus d'emplois

Ce graphique exprime le nombre d'offres d'emploi par société, pour les 10 sociétés offrant le plus d'emplois. On peut également voir la ligne horizontale qui représente la moyenne du nombre d'offres pour toutes les 268 sociétés présentes dans les données.

Cette analyse est importante car elle identifie les recruteurs les plus actifs.

Les mots les plus utilisés dans les avis

Pour cette analyse, nous avons dû créer une nouvelle table, directement sur Qlik Sense, qui fait du "MapReduce" sur le champ "titre avis". En gros, elle compte le nombre d'occurrences de chaque mot. Nous n'affichons que les mots qui ne sont pas des *stop words* (ex : le, la, mais, avec, etc.), afin que l'analyse se concentre uniquement sur des mots porteurs de sens.

- Tous les graphiques sont dynamiques.

Conclusion

Principes retenus

Grâce à ce projet, nous avons mis en œuvre un pipeline de données de A à Z, de l'extraction initiale jusqu'à la visualisation. Cela nous a permis d'appliquer concrètement les bonnes pratiques d'ingénierie des données, tant en matière d'architecture projet que dans la construction complète d'un Data Lakehouse.

Difficultés rencontrées

- Nous avons rencontré quelques difficultés avec le script qui permet de récupérer les avis car ces derniers ont des sauts à la ligne, des guillemets, et autres. La fonction qui récupère les avis les renvoie sous forme de liste de listes : chaque avis est représenté par une liste, et ses informations (titre, note, avantages, inconvénients…) sont des sous listes.
- Une autre difficulté rencontrée a été sur la partie ETL, où nous avons dû regrouper toutes les villes qui sont écrites différemment (ex. Paris, Région de Paris, Paris 06…). Beaucoup de données sont également manquantes, notamment les régions.
- La prise en main de Qlik Sense ne s'est pas faite naturellement, mais nous sommes contents d'avoir découvert un nouvel outil.