

# LIVRABLE 1 : DONNÉES ET MODÉLISATION



**KERGOAT-DUBART EWEN**

**GLAIROT VALENTIN**

**FERAT ALEXANDRE**

**LEGUERNEY CAMILLE**

**VERGER VINCENT**

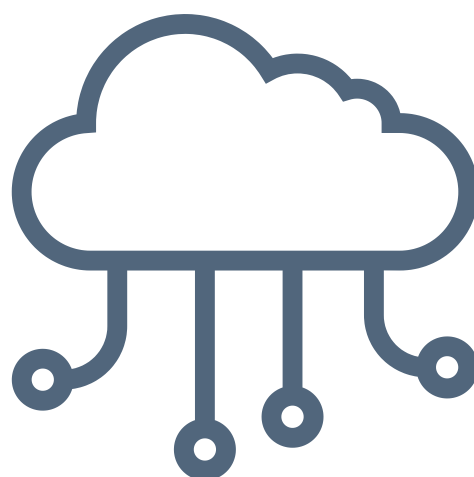
# SOMMAIRE :

<b>I</b>	Introduction	p.3
<b>II</b>	Dictionnaire de données	p.4
<b>III</b>	Modèle conceptuel de données	p.5
<b>IV</b>	Modèle logique de données	p.6
<b>V</b>	Modèle physique de données	p.7
<b>VI</b>	Arbres algébriques des requêtes	p.9
<b>VIII</b>	Conclusion	p.15



# INTRODUCTION :

Suite au dernier rapport du GIEC, le ministère de l'Écologie a chargé notre entreprise DATA-X de développer un outil pour stocker et interroger les données publiques sur la qualité de l'air dans les grandes villes de France. Cette initiative vise à centraliser les informations dispersées dans les agences météorologiques locales. En partenariat avec le ministère, DATA-X doit concevoir une base de données relationnelle pour gérer efficacement ces données et répondre aux exigences spécifiées lors de réunions avec les responsables gouvernementaux. Dans cette première réunion, nous allons donc fournir au ministère de l'écologie le MCD, MLD qui vont leur permettre de visualiser la base de données proposée.



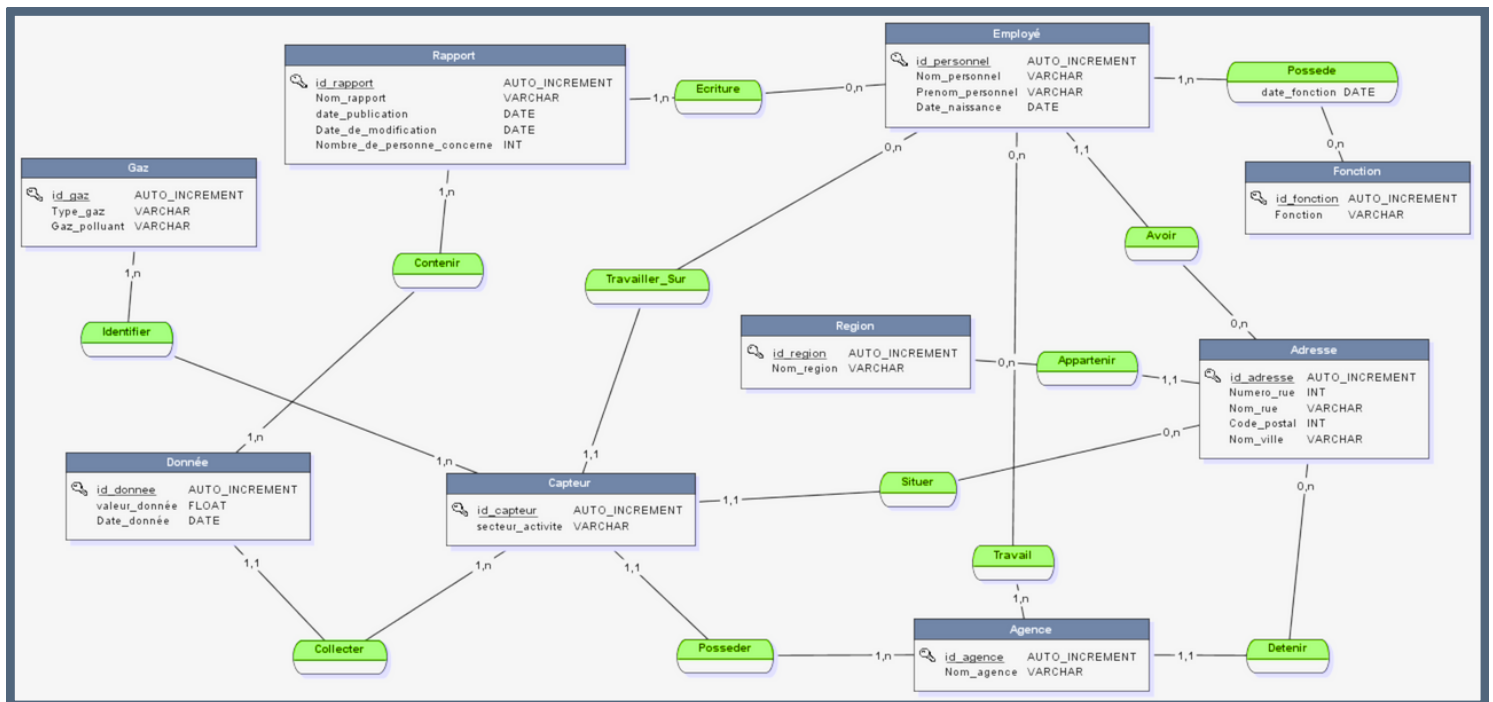
# DICTIONNAIRE DE DONNÉES :

Un dictionnaire de données est une documentation qui définit et décrit les données utilisées dans une base de données. Il fournit des informations détaillées sur chaque données, y compris son nom, son type, sa signification, sa source, ses contraintes, ses relations avec d'autres données, etc... Notre société a débuté l'élaboration de la base de données demandée par la création d'un dictionnaire de données pour répertorier les différentes variables nécessaires par la suite.

nom	type	designation	size
d_agence	AUTO_INCREMENT	identifiant unique de l'agence	
nom_agence	VARCHAR	nom de l'agence	
d_personnel	AUTO_INCREMENT	identifiant unique de la personne	
nom_personnel	VARCHAR	nom de la personne	
prenom_personnel	VARCHAR	prénom de la personne	
date_naissance_personnel	DATE	date de naissance de la personne	
date_embauche_personnel	DATE	date d'embauche de la personne	
d_fonction	AUTO_INCREMENT	identifiant unique de la fonction	
onction	VARCHAR	Fonction du personnel	
d_capteur	AUTO_INCREMENT	identifiant unique du capteur	
ecteur_activite	VARCHAR	secteur d'activité concerné par le capteur	
d_donnee	AUTO_INCREMENT	identifiant unique de la donnée	
valeur_donnee	FLOAT	valeur de la donnée	
date_donnee	DATE	date de la mesure de la donnée	
d_rapport	AUTO_INCREMENT	identifiant unique du rapport	
nom_rapport	VARCHAR	Nom du rapport	
date_publication	DATE	date de publication du rapport	
Date_de_modification	DATE	Date de modification du rapport	
Nombre_de_personne_concerne	INT	Nombre de personne ayant participé au rapport	
d_region	AUTO_INCREMENT	identifiant unique de la région	
nom_region	VARCHAR	nom de la région	
d_gaz	AUTO_INCREMENT	identifiant unique du gaz	
Type_gaz	VARCHAR	Type de gaz détecté	
gaz_polluant	VARCHAR	nom du gaz polluant émis	
d_adresse	AUTO_INCREMENT	identifiant unique de l'adresse	
Numero_rue	INT	Numero de la rue	
Nom_rue	VARCHAR	Nom de la rue	
Code_postal	INT	Code postal associé	
Nom_ville	VARCHAR	Nom de la ville	
date_fonction	DATE	Date de la prise de fonction	

Ci-dessus, nous avons donc détaillé le nom, le type, la désignation ainsi que la taille. Nous n'avons rien renseigné dans "size", ce qui permet d'avoir la limite maximale de notre type directement.

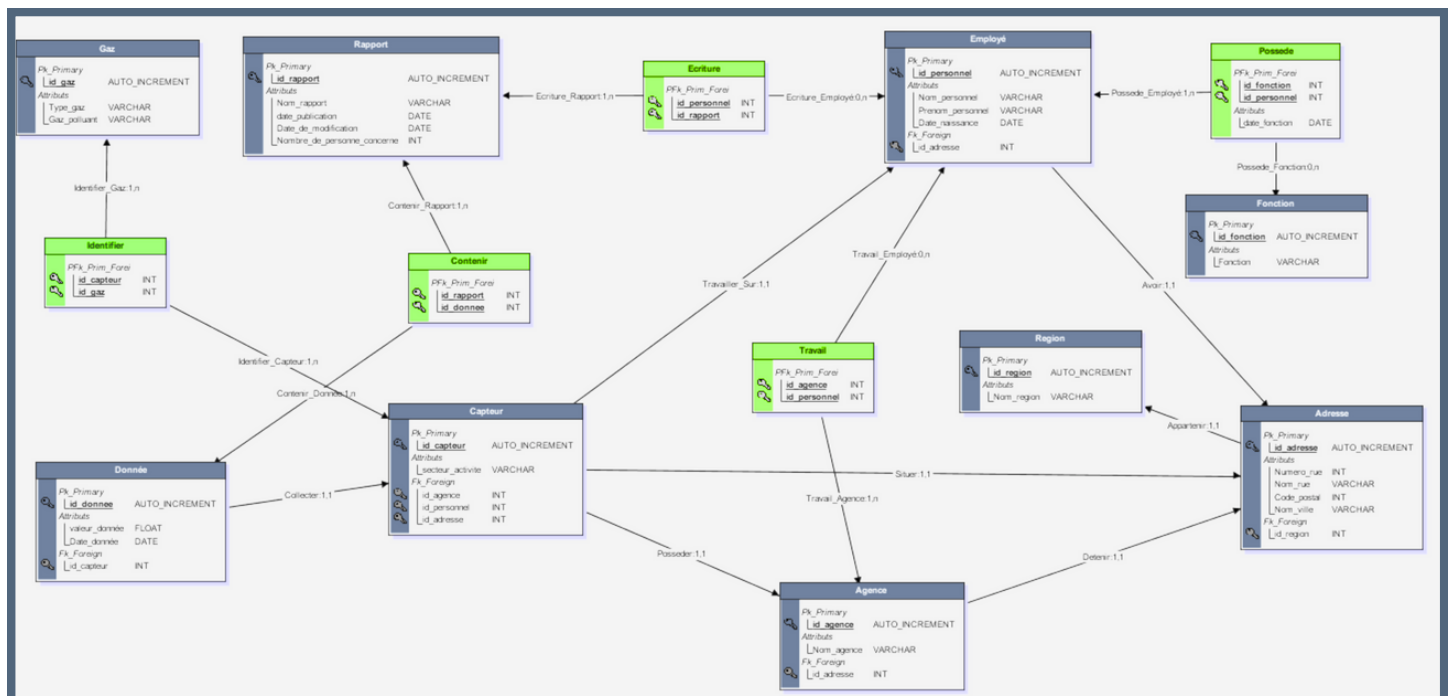
# MODÈLE CONCEPTUEL DE DONNÉES :



Un MCD (Modèle Conceptuel de Données) fournit une vue abstraite et conceptuelle des données et de leurs relations dans un système. Voici notre MCD, nous l'avons construit grâce au dictionnaire de données présenté en aval. Nous avons créé des liens entre les différentes tables (en gris) avec des relations (en vert). Les relations entre les tables sont des verbes à l'infinitif pour montrer clairement les actions menées entre les tables de notre MCD.

On voit également les cardinalités sur ce schéma, (exemple : 1,n) indique le nombre maximum d'occurrences d'une entité dans une relation.

# MODÈLE LOGIQUE DE DONNÉES :



Voici notre MLD (Modèle Logique de Données) qui est une représentation schématique des données et de leurs relations. Contrairement à notre MCD présenté plus tôt dans ce document, qui se concentre sur les concepts et les relations entre eux, le MLD spécifie les structures de données, y compris les tables, les colonnes, les clés primaires et étrangères, ainsi que les contraintes d'intégrité.

# MODÈLE PHYSIQUE DE DONNÉES :

Un MDP (Modèle Physique de Données) représente la façon dont les données sont réellement stockées sur un support de stockage, tel qu'un disque dur, dans une base de données. À la différence de notre MLD qui décrit la structure de la base de données indépendamment de tout système de gestion de base de données (SGBD) spécifique, le MDP tient compte des caractéristiques et des fonctionnalités spécifiques d'un SGBD donné. Le MDP détaille la manière dont les tables, les index, les contraintes d'intégrité, etc... sont physiquement implémentés dans la base de données.



Nous allons à présent créer chaque table grâce à des requêtes SQL. Voici ci-dessous un exemple de requêtes que nous avons utilisées pour créer les tables : Gaz, Région, Rapport, Capteur, Donnée, Agence, Employé, Fonction et Adresse.

## # Table: Adresse

```
CREATE TABLE Adresse(  
    id_adresse Int NOT NULL ,  
    Numero_rue Int NOT NULL ,  
    Nom_rue Varchar NOT NULL ,  
    Code_postal Int NOT NULL ,  
    Nom_ville Varchar NOT NULL ,  
    id_region Int NOT NULL
```

```
CONSTRAINT Adresse_PK PRIMARY KEY (id_adresse)
```



On commence par créer la table avec **"CREATE TABLE"** puis l'on détaille chaque attribut, leur nom **"id\_adresse"**, leur type **"INT"**, **"NOT NULL"** indique seulement que la case ne peut pas rester vide. Une contrainte **"CONSTRAINT"** est appliquée qui est juste la clé primaire **"PRIMARY KEY"** de la table **"gaz"**, la clé sera **"id\_adresse"**. Cette explication est un exemple de création de table, le processus sera identique pour la création des autres tables.



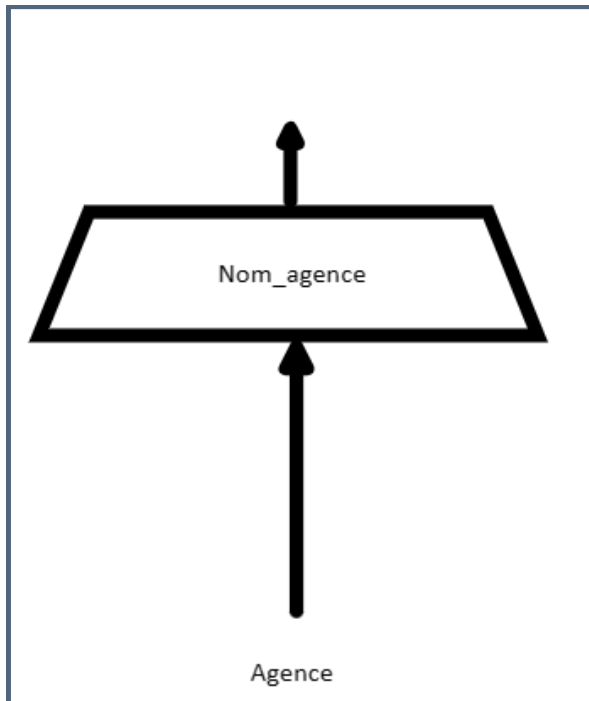
Nous allons à présent ajouter une contraintes avec **"CONSTRAINT"**, la contrainte **"FOREIGN KEY"** est utilisée pour établir une relation entre deux tables. Elle garantit l'intégrité en spécifiant que la colonne de la table **"Agence"** fait référence à une clé primaire de la table **"Adresse"**. La contrainte FOREIGN KEY garantit que les valeurs de la colonne **"id\_adresse"** dans la table **"Agence"** doivent correspondre aux valeurs de la colonne **"id\_adresse"** de la table **"Adresse"**. L'instruction **"REFERENCES"** est utilisée pour établir une relation de clé étrangère entre deux tables. **"ENGINE"** correspond au moteur de stockage à utiliser pour une table dans une SGBD.

```
CONSTRAINT Agence_Adresse_FK FOREIGN KEY (id_adresse) REFERENCES  
Adresse(id_adresse)  
)ENGINE=InnoDB
```



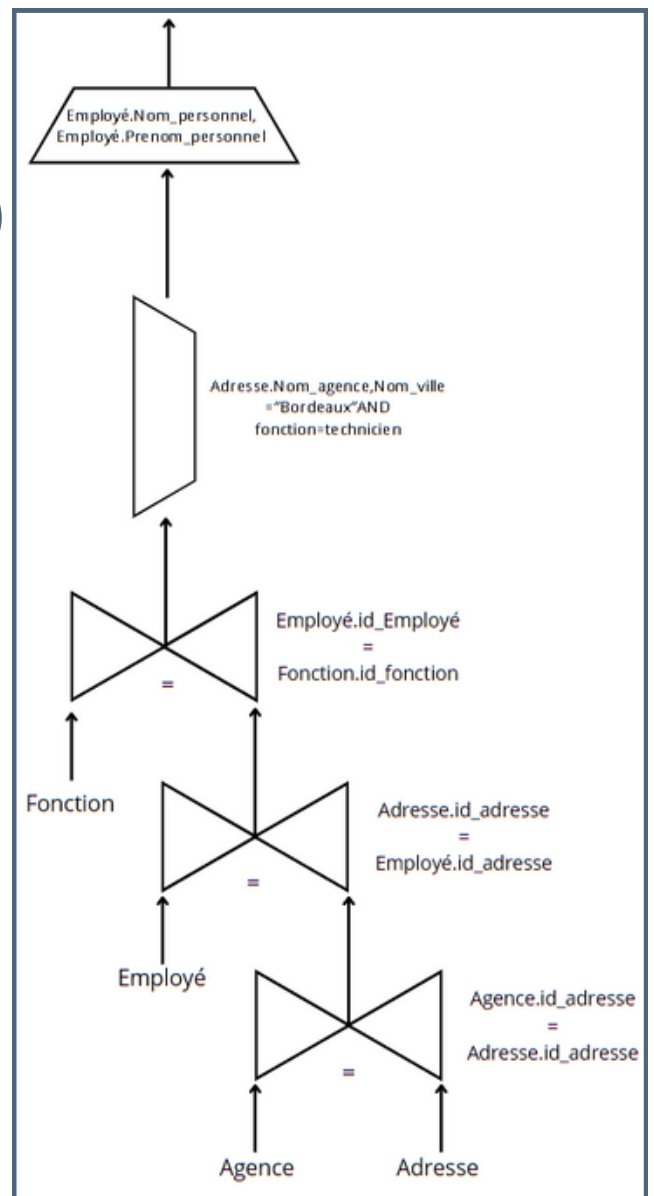
# ARBRES ALGÈBRIQUES DES REQUÊTES :

1)



L'arbre algébrique ci-contre, nous permet de lister l'ensemble des agences à partir de la table "**Agence**".

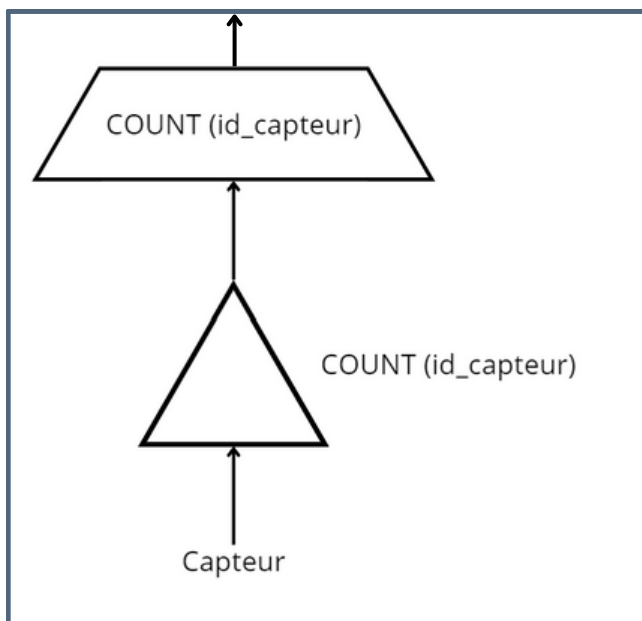
2)



L'arbre algébrique ci-contre nous permet de lister le personnel technique de l'agence de Bordeaux. Cet arbre est constitué d'une première jointure entre la table "**Agence**" et "**Adresse**". Le résultat est par la suite joint à la table "**Employé**", elle même joint avec la table "**Fonction**" par la suite. Ceci est ensuite associé à une condition de type "Adresse.Nom\_agence, Nom\_ville", qui est associé à un "AND" et nous renvoie ensuite le résultat.



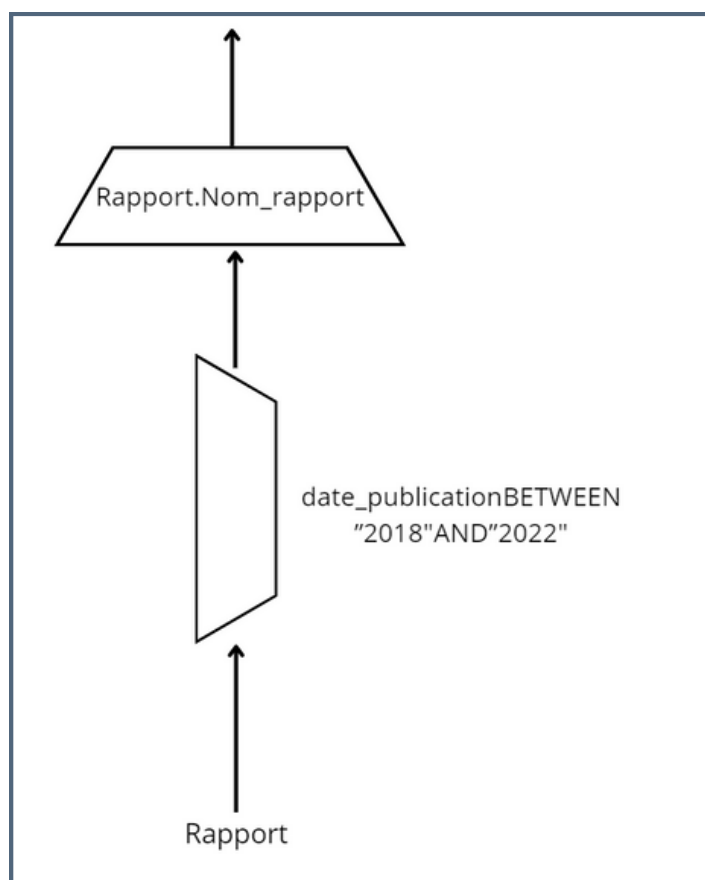
3)



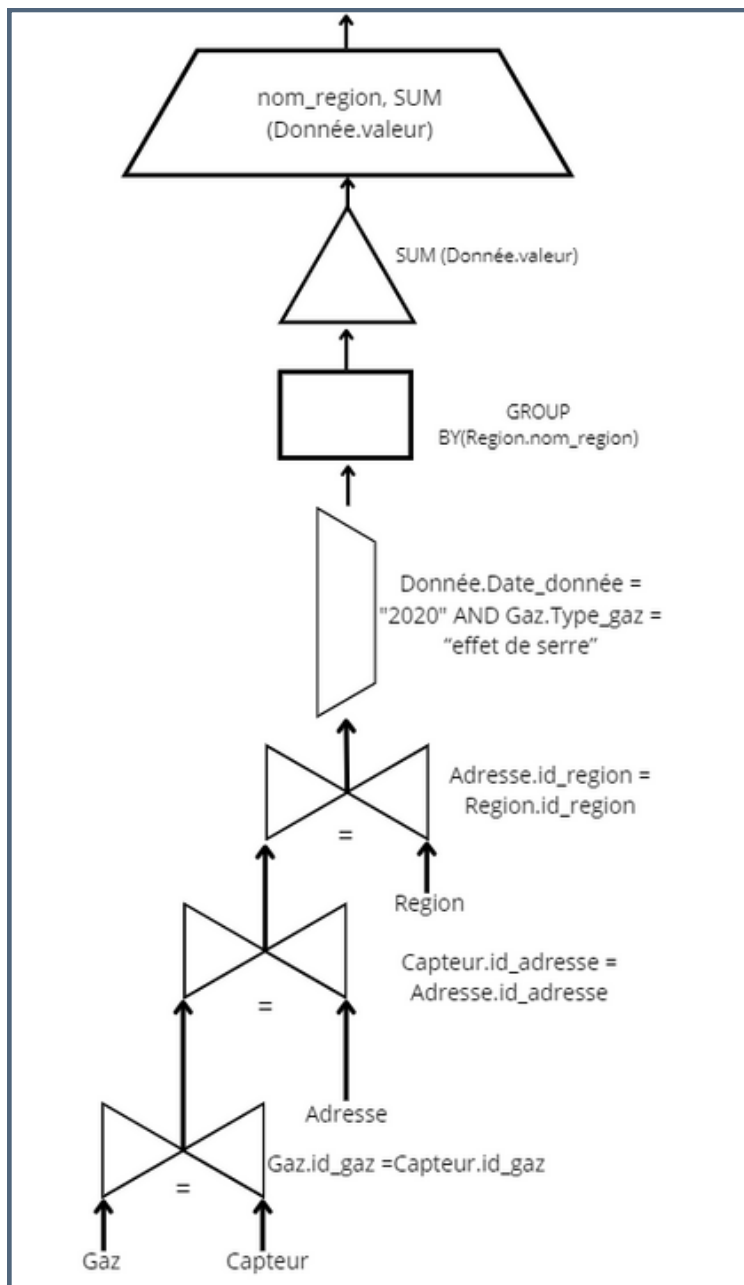
L'arbre algébrique ci-contre, nous permet d'indiquer le nombre total de capteurs déployés, à partir de la table "Capteur". Il est fait à partir de la table "Capteur" puis subit l'agrégation "COUNT" et nous renvoie le résultat.

L'arbre algébrique ci-contre nous permet de lister les rapports publiés entre 2018 et 2022. Il est constitué en partant de la base de la table "Rapport" et est associé à une condition "date\_publication" modélisée par le trapèze vertical, associée à un BETWEEN, le résultat, modélisé par le trapèze horizontal, sera le nom des rapports répondant à cette condition.

4)



5)

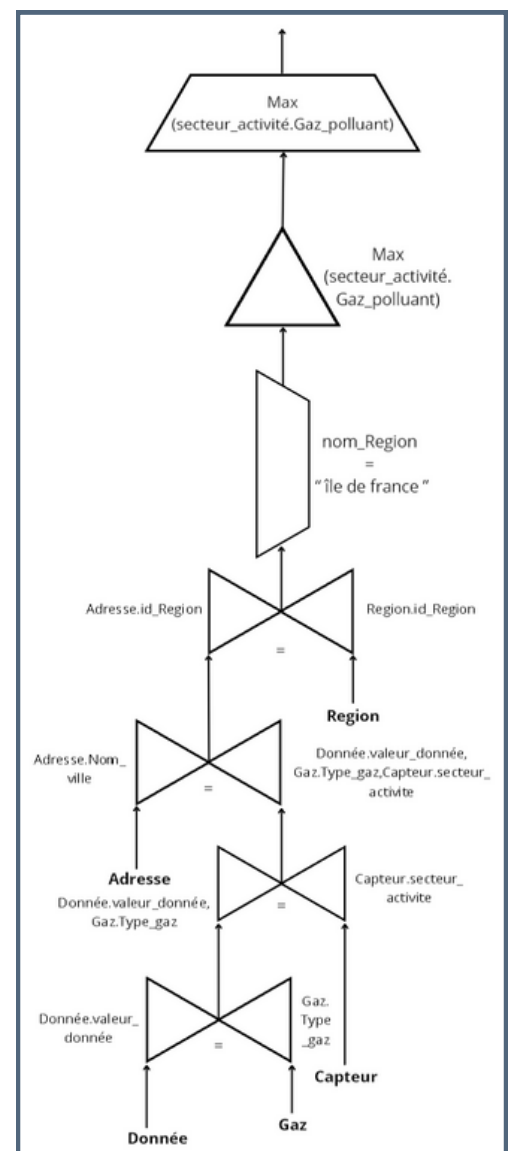


L'arbre algébrique ci-contre nous permet d'afficher le secteur industriel le plus polluant d'Île-de-France. Il est tout d'abord constitué d'une jointure entre les tables **"Gaz"** et **"Donnée"**, elles-mêmes associées par la suite à la table **"Capteur"** dans une nouvelle jointure, qui sera associée à une dernière jointure avec la table **"Adresse"**.

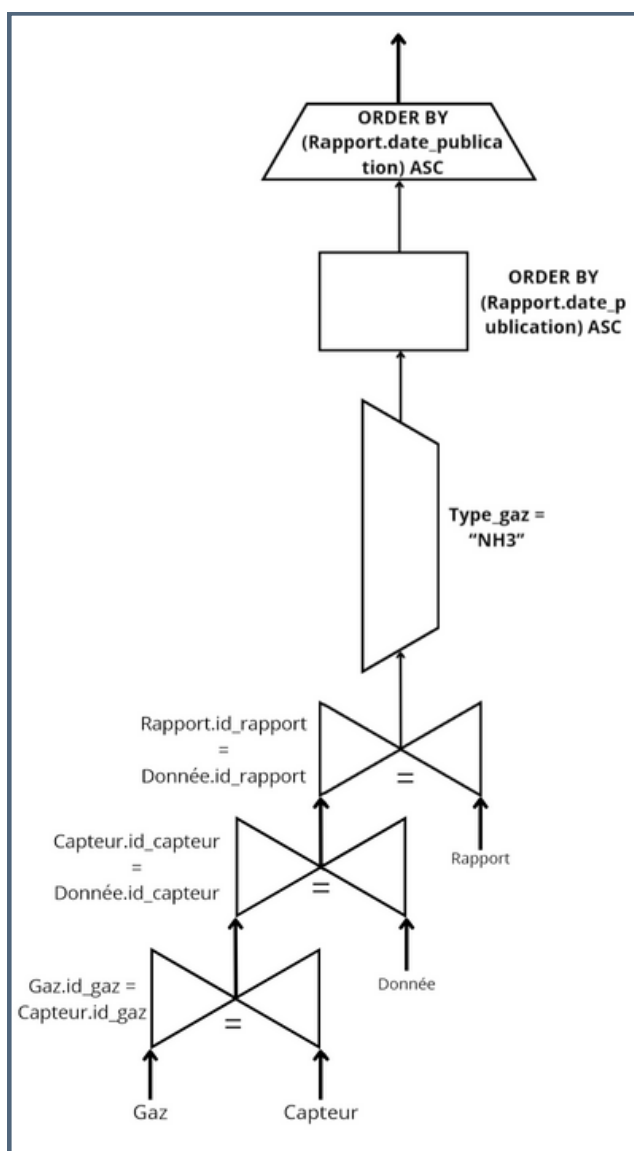
Ces jointures sont ensuite associées à la condition "nom\_Region", puis subissent une agrégation de type "MAX" nous donnant donc le secteur industriel le plus polluant d'Île de France dans le résultat.

L'arbre algébrique ci-contre, nous permet de calculer les émissions totales de gaz à effet de serre par région en 2020. Il est constitué d'une jointure entre les tables **"Gaz"** et **"Capteur"**, ce résultat est ensuite associé à une seconde jointure concernant la table **"Adresse"**, et ce résultat est ensuite associé à une dernière jointure concernant la table **"Région"**. Le résultat des jointures est soumis à une condition recherchant dans l'année 2020 les données avec une date = "2020", il est ensuite associé à un **GROUP BY**, puis subit par la suite une agrégation de type **"SUM"** faisant la somme entre les données de 2018 et de 2020 et nous donne ensuite le résultat.

6)

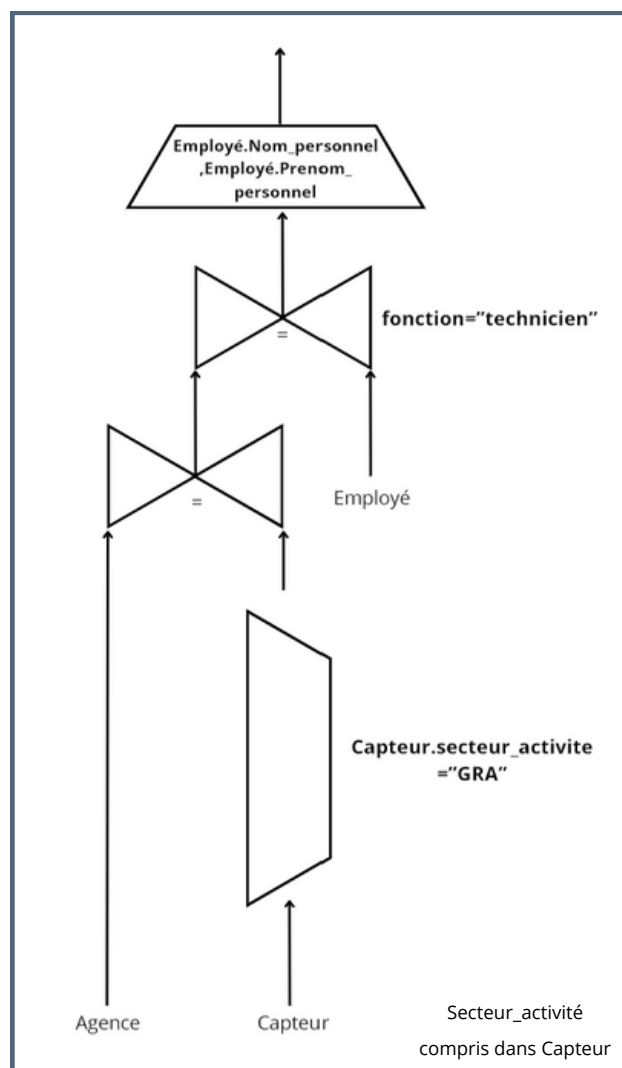


7)



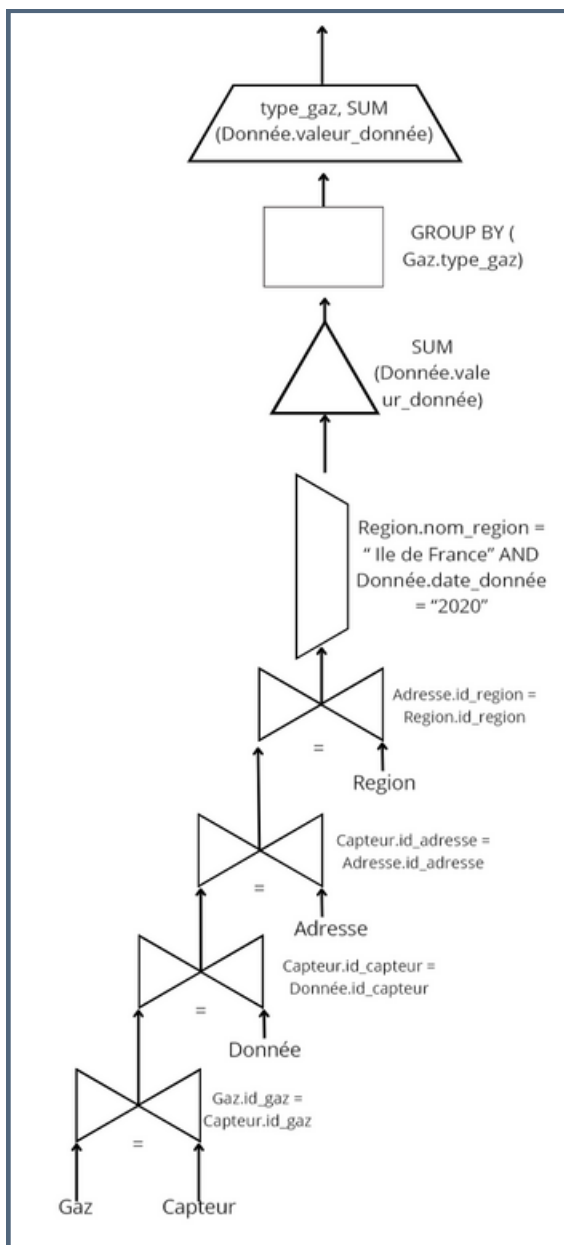
Cet arbre algébrique nous sert à classer les rapports qui relèvent les émissions de NH3 par ordre chronologique. Il est tout d'abord constitué d'une jointure agissant sur les tables "**Gaz**" et "**Capteur**", celle-ci est associée ensuite à la table "**Donnée**" par une seconde jointure, qui est elle-même associée à une troisième jointure concernant la table "**Rapport**". Ces trois jointures sont ensuite soumises à une condition " $Type\_gaz$ ", puis ce résultat est soumis à un " $ORDER BY$ ", qui nous renvoie ensuite le résultat.

8)



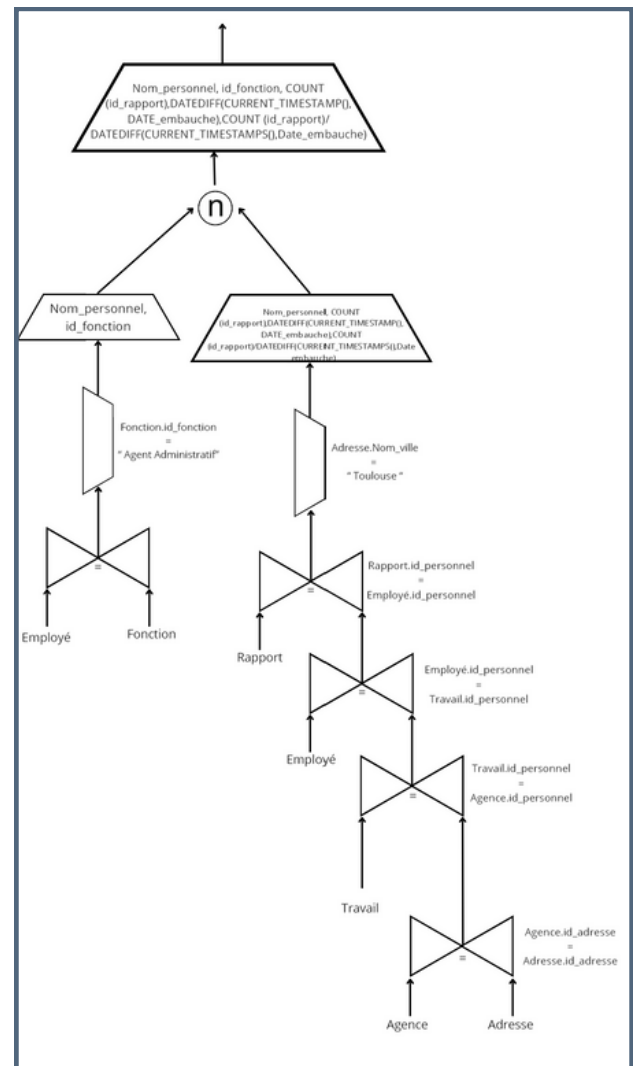
L'arbre algébrique ci-contre nous permet de fournir les noms des agents techniques chargés de la maintenance des capteurs de polluants acidifiants ( GRA ). Une condition est donc faite sur la table "**Capteur**", cette condition est ensuite associée à la table "**Agence**" par une jointure. Cette jointure est elle-même placée dans une autre jointure avec la table "**Employé**". Cette dernière jointure nous renvoie ensuite le résultat.

9)



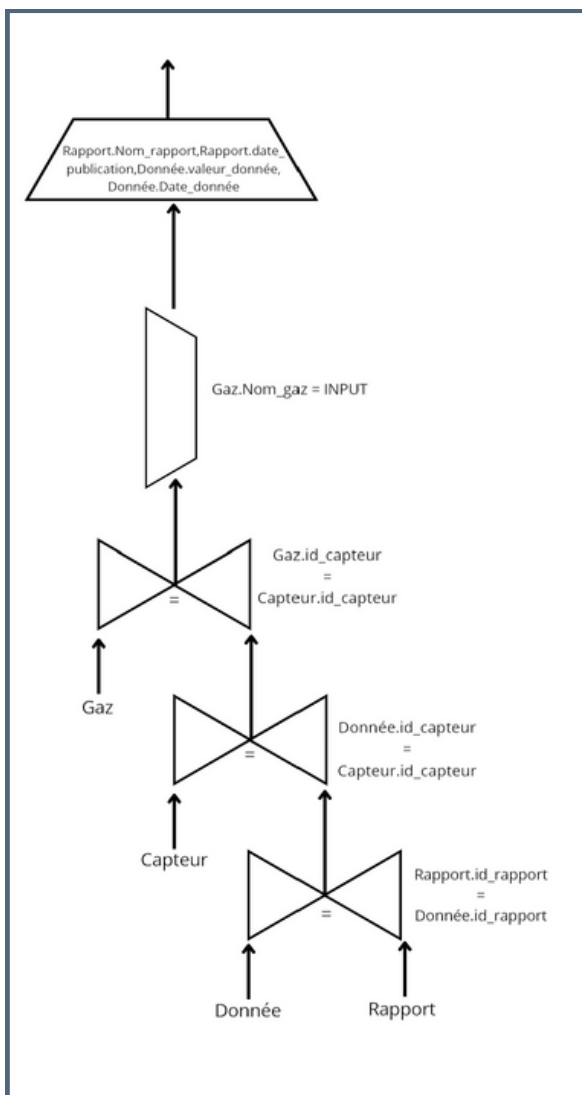
L'arbre algébrique ci-contre nous permet, pour chaque gaz, de fournir la somme des émissions (en tonnes) dans la région Île-de-France en 2020. Il est constitué tout d'abord d'une jointure entre les tables "**Gaz**" et "**Capteur**", cette jointure est ensuite associée à une seconde jointure avec la table "**Donnée**", ces deux jointures sont elles même associées à une troisième jointure concernant la table "**Adresse**", et enfin elles sont toutes les trois associées à une quatrième jointure concernant la table "**Région**". Ces quatre jointures subissent par la suite deux conditions de type "Region.nom\_region" et "Donnée.date\_donnée". Tout cela est ensuite soumis à une agrégation de type "SUM", et le résultat de ceci est soumis à un "GROUP BY", avant de nous renvoyer le résultat.

10)



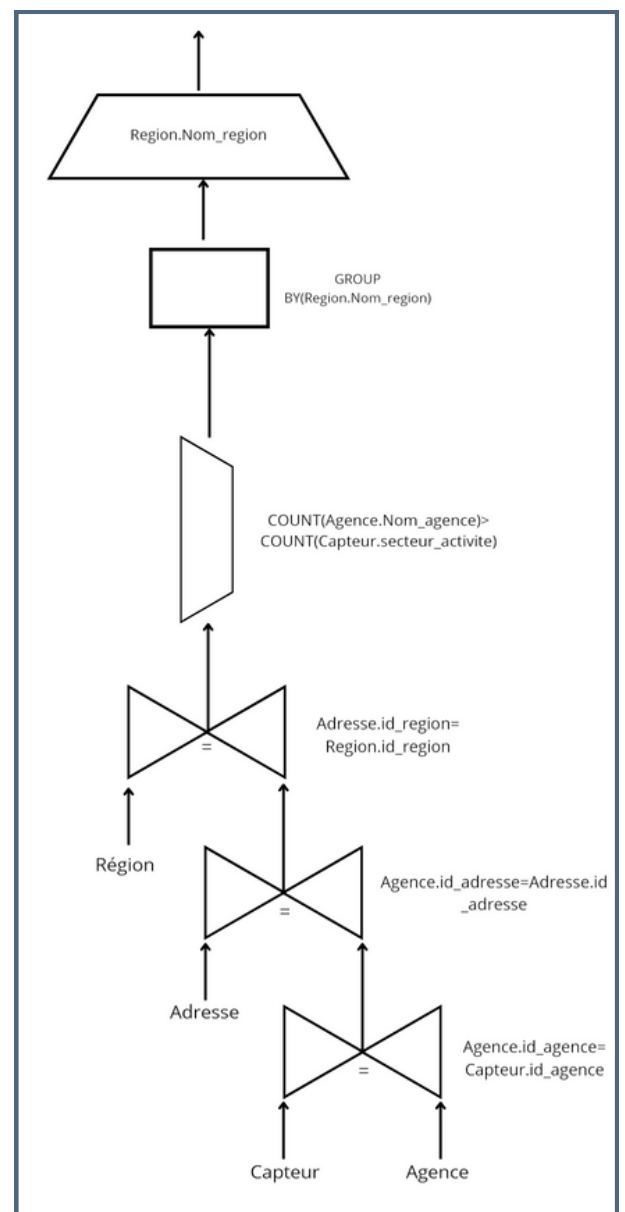
L'arbre algébrique ci-contre nous permet de fournir le taux de productivité des agents administratifs de l'agence de Toulouse (basé sur le nombre de rapports rédigés et leur ancienneté dans le poste). Il est tout d'abord constitué d'une jointure entre les tables "**Agence**" et "**Adresse**", le résultat de cette jointure constitue la jointure suivante entre celles-ci et la table "**Travail**", qui est ensuite associée à la prochaine jointure avec la table "**Employé**". Pour finir, la table "**Rapport**" est jointe. Cette jointure finale est soumise à la condition que la ville soit "Toulouse". En parallèle, les tables "**Employé**" et "**Fonction**" sont jointes puis filter avec la condition que ce soit un agent administratif. Un union est fait pour avoir le résultat final. Nous renvoyons le nom du personnel, puis la division du nombre de rapports fait avec la date actuelle / date d'embauche sur ce poste.

11)



L'arbre algébrique ci-contre nous permet, pour un gaz donné, d'énumérer les rapports contenant des données le concernant (le nom du gaz doit être un paramètre). Il est tout d'abord constitué d'une jointure entre les tables "**Rapport**" et "**Donnée**", le résultat de cette jointure constitue la jointure suivante entre celle-ci et la table "**Capteur**", qui est ensuite associée à la prochaine jointure avec la table "**Gaz**". Cette jointure finale est soumise à une condition et nous donne ensuite le résultat final.

12)



L'arbre algébrique ci-contre nous permet de lister des régions où il y a moins de capteurs que d'agences. Il est tout d'abord constitué d'une jointure entre les tables "**Capteur**" et "**Agence**", le résultat de cette jointure constitue la jointure suivante entre celle-ci et la table "**Adresse**", qui est ensuite associée à la prochaine jointure avec la table "**Région**". Cette jointure finale est soumise à une condition, puis subit ensuite une agrégation de type "COUNT", elle même associée à une agrégation de type "GROUP BY" nous permettant donc de lister des régions où il y a moins de capteurs que d'agences dans le résultat final.

# CONCLUSION :

Notre société a correctement établi les différents points de développement, tels que le dictionnaire de données, le modèle conceptuel de données, le modèle logique de données, le modèle physique de données, ainsi que les arbres algébriques des requêtes requises.

Notre entreprise reste à votre disposition pour tout complément d'information concernant votre projet.

