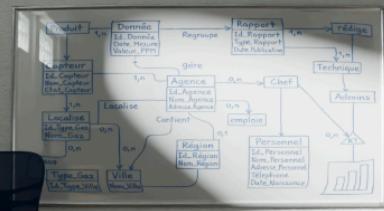


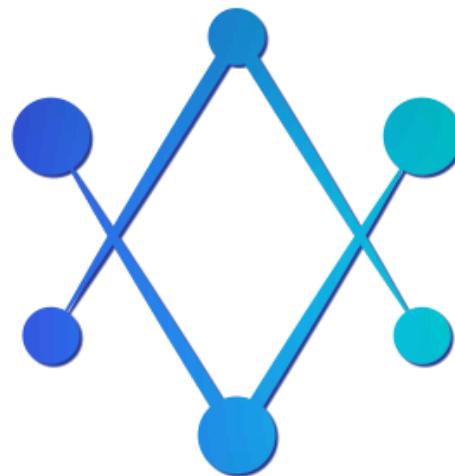
# BDD GROUPE 1 LIVRABLE 1





# SOMMAIRE

<b>1/Présentation du groupe :</b>	<b>3</b>
<b>2/Reformulation du besoin :</b>	<b>3</b>
<b>3/Dictionnaire de données :</b>	<b>4</b>
<b>4/Modèle conceptuel de données :</b>	<b>6</b>
<b>6/Modèle physique de données :</b>	<b>8</b>
<b>7/Arbres algébriques :</b>	<b>11</b>
<b>8/Explication des choix de modélisation :</b>	<b>17</b>
<b>9/Conclusion :</b>	<b>17</b>



**ClearData**



## 1/Présentation du groupe :

### Membres du groupe:

///

///

///

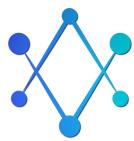
///

Nom	Rôle	Compétences principales
Frigga, Benkherouf	Responsable base de données	MySQL, MCD/MLD, SQL
Amokrane	Responsable des données	Données réalistes, Excel, automatisation
Frigaa, Benkherouf	Responsable requêtes SQL	Requêtes complexes, agrégation
Lebaili	Responsable documentation	Gestion de projet, LaTeX, planification

## 2/Reformulation du besoin :

Le ministère de la Transition Écologique souhaite centraliser les données sur la qualité de l'air actuellement dispersées dans les agences régionales. L'entreprise ClearData est missionnée pour concevoir une base de données relationnelle prototype permettant de :

- Gérer les agences, le personnel (chefs, agents techniques, administratifs),
- Suivre les capteurs, les gaz mesurés et les relevés mensuels,
- Gérer les rapports rédigés par les agents,
- Fournir des requêtes pertinentes aux chefs d'agence. Le système doit garantir :
  - L'intégrité des données (contraintes, cohérence),
  - Le respect du RGPD (notamment pour les données RH),
  - Des fonctions d'ajout et suppression, sans modification possible des rapports ou mesures,
  - Une base de données test fonctionnelle, réaliste, exploitable en démonstration.



### 3/Dictionnaire de données :

#### DICTIONNAIRE

Nom de l'attribut	Type SQL	Taille	Clé	Contraintes
id_agence	INT	//	PK	NOT NULL, AUTO_INCREMENT
nom_agence	VARCHAR	100	//	NOT NULL
adresse_agence	VARCHAR	200	//	NOT NULL
id_personnel	INT	//	PK	NOT NULL, AUTO_INCREMENT
nom_personnel	VARCHAR	100	//	NOT NULL
prenom_personne l	VARCHAR	100	//	NOT NULL
date_naissance	DATE	//	//	NOT NULL
date prise poste	DATE	//	//	NOT NULL
adresse_personnel	VARCHAR	200	//	NOT NULL
diplome_chef	VARCHAR	100	//	NULLABLE
id_rapport	INT	//	PK	AUTO_INCREMENT
titre_rapport	VARCHAR	150	//	NOT NULL
date_publication	DATE	//	//	NOT NULL
id_donnée	INT	//	PK	AUTO_INCREMENT
date_mesure	DATE	//	//	NOT NULL



Nom de l'attribut	Type SQL	Taille	Cle	Contraintes
valeur_ppm	FLOAT	//	//	CHECK (valeur_ppm BETWEEN 0.01 AND 500)
id_capteur	INT	//	PK	AUTO_INCREMENT
Nom_Capteur	VARCHAR	50	//	Nom du capteur
etat_capteur	BOOLEAN	//	//	NOT NULL
id_gaz	INT	//	PK	AUTO_INCREMENT
Type_Gaz	CHAR	50	//	NOT NULL
Id_Ville	INT	//	PK	AUTO_INCREMENT
Nom_Ville	VARCHAR	50	//	NOT NULL
id_Region	INT	//	PK	AUTO_INCREMENT
Nom_Region	VARCHAR	50	//	NOT NULL

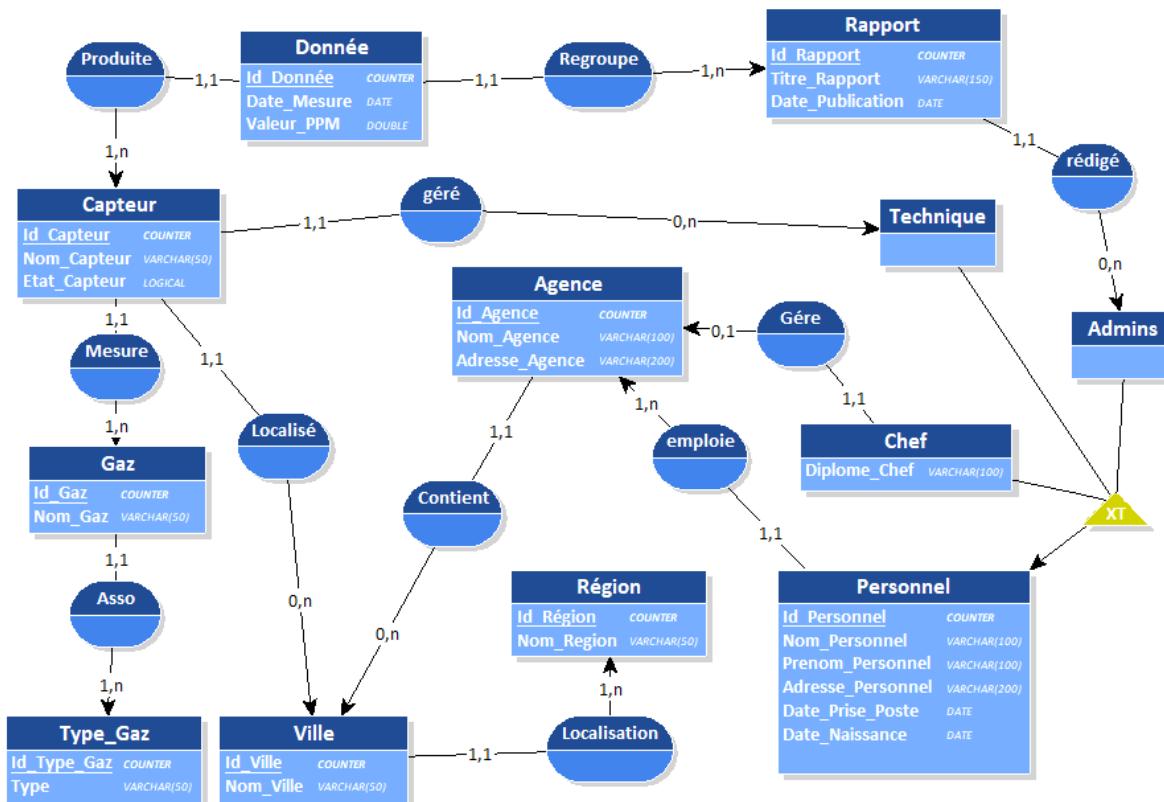
### IMPORTANT :

L'attribut 'valeur\_ppm' a une contrainte de valeur entre 0.01 et 500 car il a été demandé ainsi.



## 4/Modèle conceptuel de données :

Le modèle conceptuel de données (MCD) est une représentation logique des données du projet. Il décrit les entités importantes (comme Agence, Agent, Capteur, Rapport), leurs attributs (par exemple nom, date, adresse), et les relations entre elles. Le MCD sert à structurer les informations sans entrer dans les détails techniques. Il est utilisé pour bien comprendre les besoins avant de créer la base de données.



### IMPORTANT:

Le MCD a été structuré d'après le texte du projet qui stipule la structuration du déroulement des opérations de l'organisation puis normalisé à la 3eme forme normale (On enlève les dépendances **transitives**, c'est-à-dire que les informations ne dépendent que de la clé, et pas d'un autre champ).

-les valeurs comme Etat\_Capteur n'ont pas été mis dans leur propre table car il a été déterminé qu'il était non nécessaire de créer une nouvelle table juste pour des valeur de type boolean (True (1), False(0)) qui originellement ne prenne pas beaucoup d'espace.

### Exemple de relations:

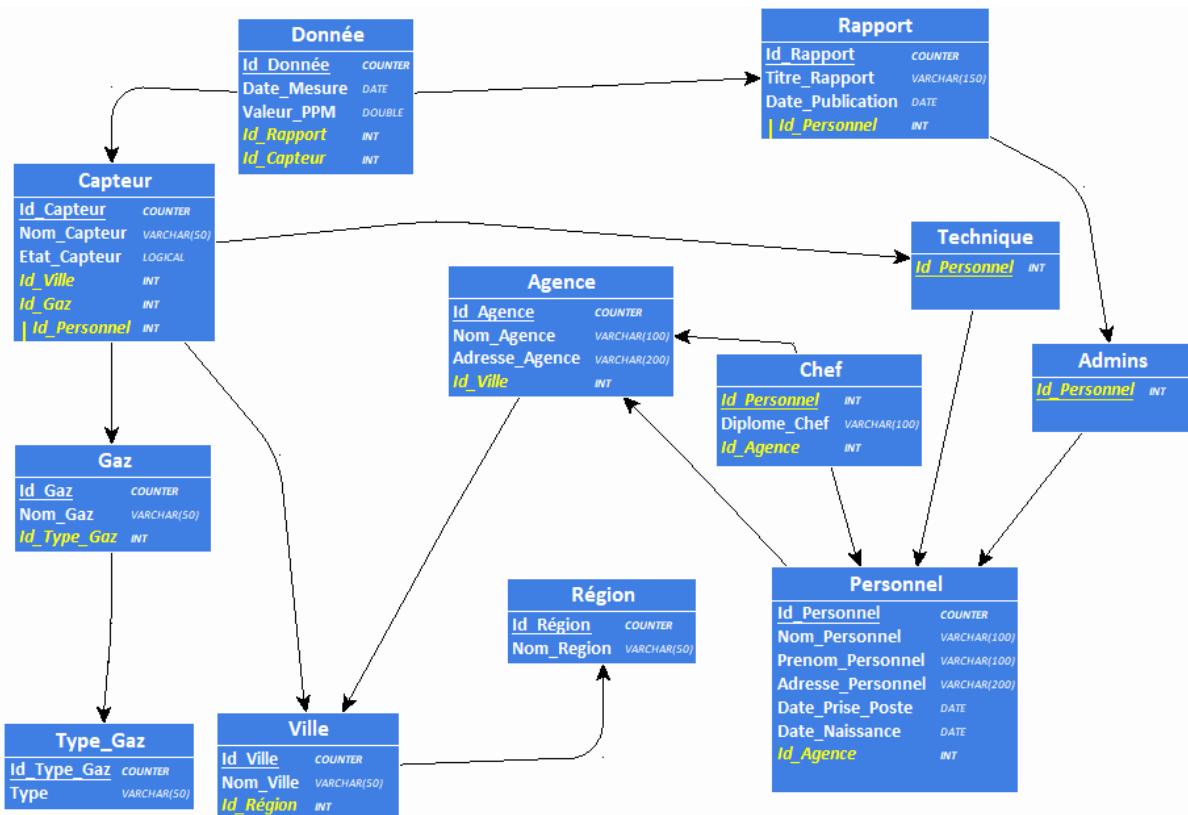
-Une Agence peut employer plusieurs membres du personnel et chaque personnel doit travailler dans une agence.



-Une Agence peut être gérée par 1 chef ou ne pas avoir de chef comme l'agence de Limoges.

## 5/Modèle logique de données :

Le modèle logique de données est une version plus technique du modèle conceptuel. Il décrit la structure de la base de données en tenant compte du système de gestion choisi (par exemple MySQL). Il précise les noms des tables, les noms des colonnes, les types de données (comme texte, nombre, date), les clés primaires et étrangères ainsi que les relations entre les tables. Contrairement au modèle conceptuel, le modèle logique est prêt à être transformé en script SQL pour créer la base de données.



La transformation du Modèle Conceptuel de Données (MCD) en Modèle Logique de Données (MLD) consiste à traduire les entités, les relations et les attributs du MCD en structures adaptées à un système de gestion de base de données relationnelle. Chaque entité devient une table, et ses attributs se transforment en colonnes. Les identifiants des entités sont conservés comme clés primaires. Les relations de type 1,N ou N,M sont converties en ajoutant des clés étrangères ou en créant des tables associatives selon les cas. Les cardinalités guident la structuration des liens entre les tables. Cette étape permet de passer d'un modèle abstrait à une structure relationnelle cohérente, prête à être traduite en modèle physique et implémentée dans un SGBD.



## 6/Modèle physique de données :

Le Modèle Physique de Données (MPD) représente la dernière étape de la modélisation avant l'implémentation dans un système de gestion de base de données. À ce stade, les tables issues du MLD sont traduites en instructions SQL précises. C'est lors de cette phase que les contraintes d'intégrité sont générées, notamment les clés primaires (PK) qui assurent l'unicité des enregistrements, les clés étrangères (FK) qui maintiennent la cohérence entre les tables, les contraintes d'unicité (UNQ) qui empêchent les doublons sur certains attributs, et les contraintes de vérification (CHK) qui contrôlent la validité des valeurs selon des règles définies. Le MPD prépare ainsi la base de données à être créée concrètement dans un SGBD en garantissant la structure et l'intégrité des données.

**IMPORTANT : MPD Généré à l'aide du logiciel Looping**

### **CODE SQL**

---

```
CREATE TABLE Gaz(
    Id_Gaz COUNTER,
    Nom_Gaz VARCHAR(50),
    PRIMARY KEY(Id_Gaz)
);
```

```
CREATE TABLE Région(
    Id_Région COUNTER,
    Nom_Region VARCHAR(50),
    PRIMARY KEY(Id_Région)
);
```

```
CREATE TABLE Ville(
    Id_Ville COUNTER,
    Id_Région INT NOT NULL,
    PRIMARY KEY(Id_Ville),
    FOREIGN KEY(Id_Région) REFERENCES Région(Id_Région)
);
```

```
CREATE TABLE Type_Gaz(
    Id_Type_Gaz COUNTER,
    Type VARCHAR(50),
    PRIMARY KEY(Id_Type_Gaz)
);
```



```
CREATE TABLE Agence(
    Id_Agence COUNTER,
    Nom_Agence VARCHAR(100),
    Adresse_Agence VARCHAR(200),
    Id_Ville INT NOT NULL,
    PRIMARY KEY(Id_Agence),
    FOREIGN KEY(Id_Ville) REFERENCES Ville(Id_Ville)
);
```

```
CREATE TABLE Personnel(
    Id_Personnel COUNTER,
    Nom_Personnel VARCHAR(100),
    Prenom_Personnel VARCHAR(100),
    Adresse_Personnel VARCHAR(200),
    Date_Prise_Poste DATE,
    Date_Naissance DATE,
    Id_Agence INT NOT NULL,
    PRIMARY KEY(Id_Personnel),
    FOREIGN KEY(Id_Agence) REFERENCES Agence(Id_Agence)
);
```

```
CREATE TABLE Admins(
    Id_Personnel INT,
    PRIMARY KEY(Id_Personnel),
    FOREIGN KEY(Id_Personnel) REFERENCES Personnel(Id_Personnel)
);
```

```
CREATE TABLE Technique(
    Id_Personnel INT,
    PRIMARY KEY(Id_Personnel),
    FOREIGN KEY(Id_Personnel) REFERENCES Personnel(Id_Personnel)
);
```

```
CREATE TABLE Chef(
    Id_Personnel INT,
    Diplome_Chef VARCHAR(100),
    Id_Agence INT NOT NULL,
    PRIMARY KEY(Id_Personnel),
    UNIQUE(Id_Agence),
    FOREIGN KEY(Id_Personnel) REFERENCES Personnel(Id_Personnel),
    FOREIGN KEY(Id_Agence) REFERENCES Agence(Id_Agence)
```



);

```
CREATE TABLE Capteur(
    Id_Capteur COUNTER,
    Nom_Capteur VARCHAR(50),
    Etat_Capteur LOGICAL,
    Id_Ville INT NOT NULL,
    Id_Gaz INT NOT NULL,
    Id_Personnel INT NOT NULL,
    PRIMARY KEY(Id_Capteur),
    FOREIGN KEY(Id_Ville) REFERENCES Ville(Id_Ville),
    FOREIGN KEY(Id_Gaz) REFERENCES Gaz(Id_Gaz),
    FOREIGN KEY(Id_Personnel) REFERENCES Technique(Id_Personnel)
);
```

```
CREATE TABLE Rapport(
    Id_Rapport COUNTER,
    Titre_Rapport VARCHAR(150),
    Date_Publication DATE,
    Id_Personnel INT NOT NULL,
    PRIMARY KEY(Id_Rapport),
    FOREIGN KEY(Id_Personnel) REFERENCES Admins(Id_Personnel)
);
```

```
CREATE TABLE Donnée(
    Id_Donnée COUNTER,
    Date_Mesure DATE,
    Valeur_PPM DOUBLE,
    Id_Rapport INT NOT NULL,
    Id_Capteur INT NOT NULL,
    PRIMARY KEY(Id_Donnée),
    FOREIGN KEY(Id_Rapport) REFERENCES Rapport(Id_Rapport),
    FOREIGN KEY(Id_Capteur) REFERENCES Capteur(Id_Capteur)
);
```

```
CREATE TABLE Asso(
    Id_Gaz INT,
    Id_Type_Gaz INT,
    PRIMARY KEY(Id_Gaz, Id_Type_Gaz),
    FOREIGN KEY(Id_Gaz) REFERENCES Gaz(Id_Gaz),
    FOREIGN KEY(Id_Type_Gaz) REFERENCES Type_Gaz(Id_Type_Gaz)
```

);

---

**FIN DU CODE SQL**

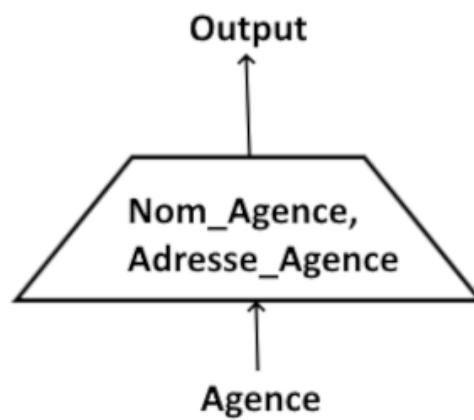
## 7/Arbres algébriques :

- $\pi$  : *Projection*
- $\sigma$  : *Sélection*
- $\bowtie$  : *Jointure*
- $\wedge$  : *Et logique (AND)*
- $\vee$  : *Ou logique (OR)*

**Liste des requêtes demandées:**

### 1. Liste de l'ensemble des agences

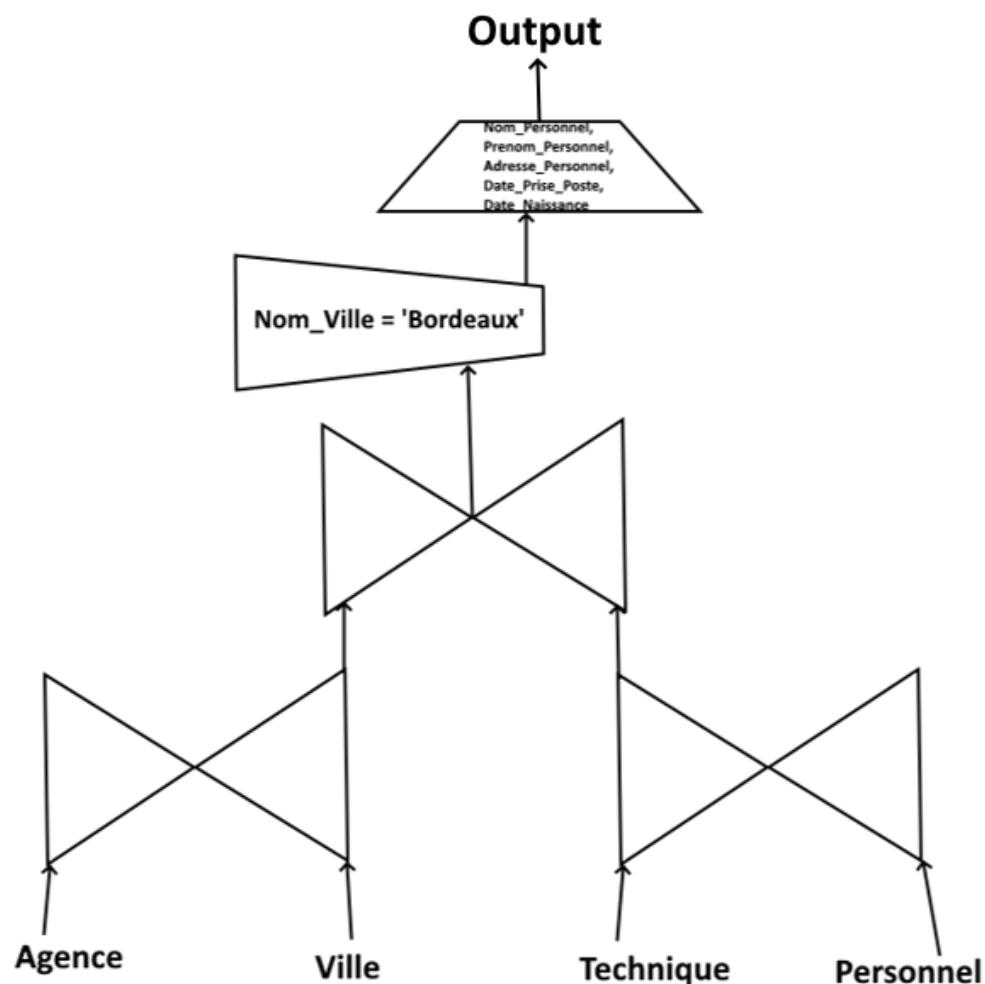
$\pi [Nom\_Agence, Adresse\_Agence] (Agence)$





## 2. Liste de l'ensemble du personnel technique de l'agence de Bordeaux

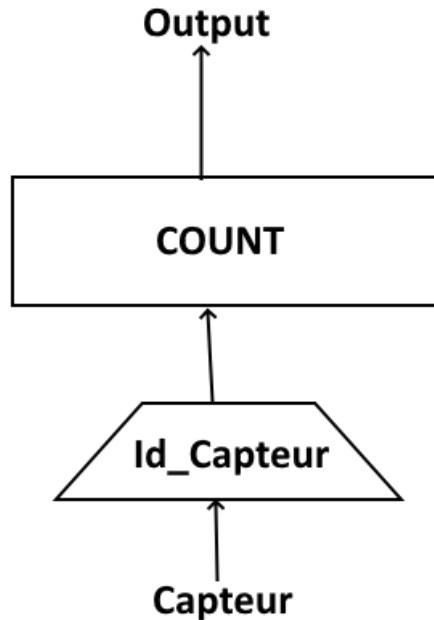
```
Π Nom_Personnel,  
Prenom_Personnel,Adresse_Personnel,Date_Prise_Poste,Date_Naissance (  
σ[Nom_Ville = 'Bordeaux'] (  
    (Technique ⋈[Technique.Id_Personnel = Personnel.Id_Personnel]  
    Personnel ⋈[Personnel.Id_Agence = Agence.Id_Agence]  
    Agence ⋈[Agence.Id_Ville = Ville.Id_Ville]  
    Ville)  
)  
)
```





### 3. Nombre total de capteurs déployés.

$\text{COUNT}(\text{Id\_Capteur})(\text{Capteur})$



### 4. Liste des rapports publiés entre 2018 et 2022

$\pi[\text{Id\_Rapport}, \text{Titre\_Rapport}]()$

$\sigma[\text{Date\_Publication} \geq 2018 \wedge \text{Date\_Publication} \leq 2022](\text{Rapport})$

)

### 5. Afficher les concentrations de CH4 (en ppm) dans les régions « Ile-de-France », « Bretagne » et « Occitanie » en mai et juin 2023.

$\pi[\text{Valeur\_PPM}, \text{Nom\_Région}]()$

$\sigma[$

$(\text{Date\_Mesure} \geq '01/05/2023' \wedge \text{Date\_Mesure} \leq '30/06/2023') \wedge$

$(\text{Nom\_Région} = 'Ile-de-France' \vee \text{Nom\_Région} = 'Bretagne' \vee \text{Nom\_Région} = 'Occitanie')$

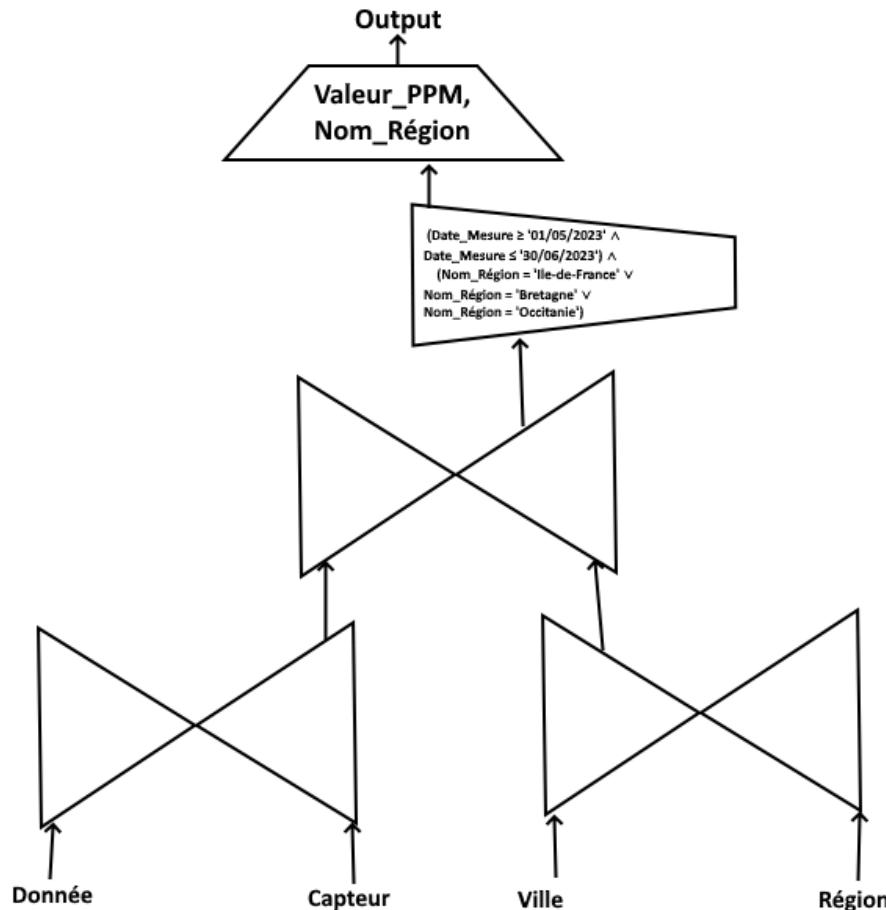
$]$

$($

$((\text{Donnée} \bowtie \text{Donnée}.\text{Id\_Capteur} = \text{Capteur}.\text{Id\_Capteur}) \text{Capteur})$

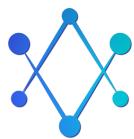


$\bowtie \text{Capteur.Id\_Ville} = \text{Ville.Id\_Ville Ville}$   
 $\bowtie \text{Ville.Id\_Région} = \text{Région.Id\_Région Région}$   
 )  
 )



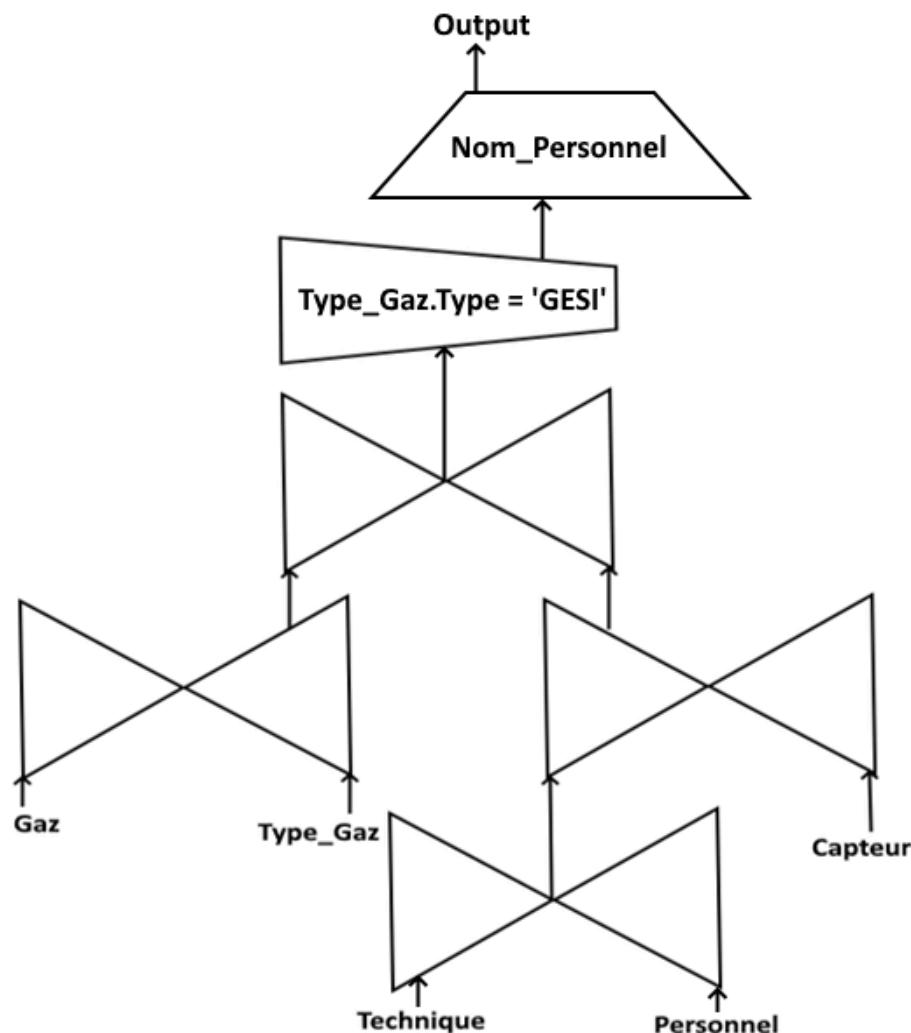
## 6. Liste des noms des agents techniques maintenant des capteurs concernant les gaz à effet de serre provenant de l'industrie (GESI).

$\pi[\text{Nom\_Personnel}] ($   
 $\sigma[\text{Type\_Gaz.Type} = \text{'GESI'}] ($   
 $((\text{Capteur}$   
 $\bowtie [\text{Capteur.Id\_Gaz} = \text{Gaz.Id\_Gaz}] \text{Gaz}$   
 $\bowtie [\text{Gaz.Id\_Type\_Gaz} = \text{Type\_Gaz.Id\_Type\_Gaz}] \text{Type\_Gaz}$   
 $\bowtie [\text{Capteur.Id\_Personnel} = \text{Technique.Id\_Personnel}] \text{Technique}$   
 $\bowtie [\text{Technique.Id\_Personnel} = \text{Personnel.Id\_Personnel}] \text{Personnel})$



)

)



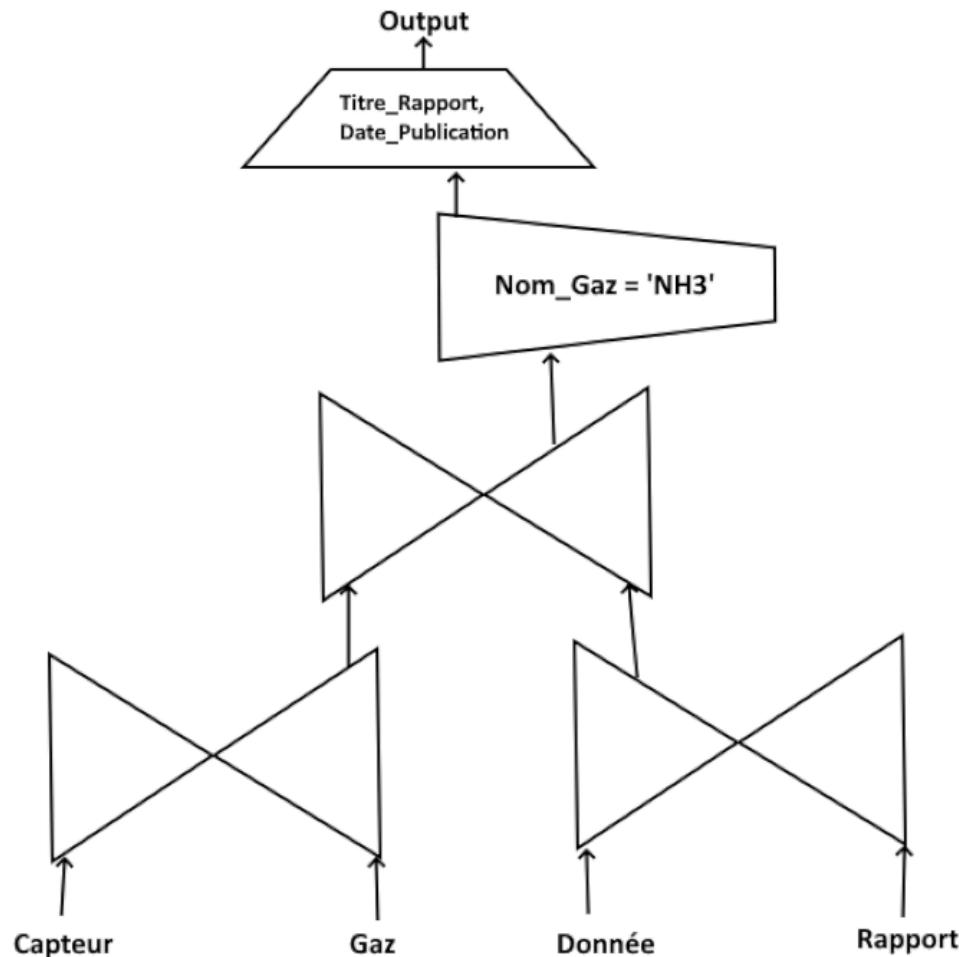
## 7. Titres et dates des rapports concernant des concentrations de NH3 (sans classement)

```
π[Titre_Rapport, Date_Publication] ( σ[Nom_Gaz = 'NH3'] ( (((Donnée  
    ⊗[Donnée.Id_Rapport = Rapport.Id_Rapport] Rapport)  
    ⊗[Donnée.Id_Capteur = Capteur.Id_Capteur] Capteur)  
    ⊗[Capteur.Id_Gaz = Gaz.Id_Gaz] Gaz)
```



)

)



## 8/Explication des choix de modélisation :

L'ensemble des choix de modélisation repose sur une analyse approfondie des besoins fonctionnels et des contraintes du système à concevoir. Le Modèle Conceptuel de Données (MCD) a été structuré de manière à refléter fidèlement les entités réelles du domaine étudié ainsi que les interactions entre elles. Chaque entité représente un concept central identifié dans les exigences, et possède des attributs définis selon leur pertinence métier et leur nécessité technique.

Les relations ont été définies en respectant les cardinalités issues des règles de gestion, afin de garantir la cohérence des liens entre les entités. Les identifiants ont été choisis avec soin pour assurer l'unicité des enregistrements, en optant pour des clés naturelles lorsque cela était pertinent, ou des clés artificielles dans le cas contraire.

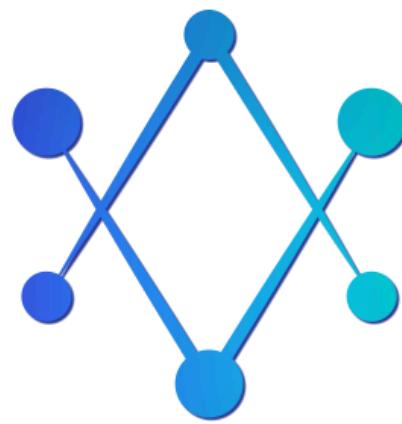


Lors du passage au Modèle Logique de Données (MLD), les entités ont été traduites en tables relationnelles, en conservant l'intégrité des dépendances. Les relations de type N:M ont été systématiquement transformées en tables associatives, afin de respecter les contraintes du modèle relationnel.

Enfin, dans le Modèle Physique de Données (MPD), les contraintes d'intégrité ont été ajoutées pour renforcer la qualité et la fiabilité des données. Les clés primaires (PK), les clés étrangères (FK), les contraintes d'unicité (UNQ) et de vérification (CHK) ont été générées pour formaliser les règles de gestion au niveau de la base. Ce processus assure un modèle robuste, cohérent et prêt à être implémenté dans un SGBD relationnel.

## 9/Conclusion :

Ce livrable constitue une étape essentielle du projet, car il pose les fondations de l'infrastructure de la base de données finale. En effet, il est indispensable de concevoir et structurer la base de données en amont, avant de pouvoir y intégrer les données et exploiter pleinement le système.



**ClearData**

**FIN DU DOCUMENT**