



## Livrable 1



**Blanc Thomas**  
**Dutertre-Thouan Lou**  
**Kadiri Shainèze**  
**Ouach Bilal**

## --- Sommaire ---

- **Le dictionnaire de données**
- **Le modèle conceptuel de données et les explications**
- **Le modèle logique de données et les explications**
- **Le modèle physique de données**
- **Les arbres algébriques de chaque requête**

## Le dictionnaire de données

Le dictionnaire des données représente une organisation des différentes données nécessaire pour vous, en d'autres termes, c'est un récapitulatif des données de votre demande.

Vous retrouverez toutes les informations ci-dessous dans le Excel ci-joint :

Nom données	Code	Type	Taille	Min	Max	Commentaires
Code Agence	ID_Agence	Nombre	4	0	13	
Chefs-Lieux Agence	Region_Agence	Texte	50	X	X	
Code Capteur	ID_Capteur	Nombre	4	0	X	
Lieu Capteur	Lieu_Capteur	Texte	50	X	X	
État Capteur	Etat_Capteur	Logique	2	X	X	
Code Gaz	ID_Gaz	Nombre	4	0	X	
Type de Gaz	Type_Gaz_Capteur	Texte	50	X	X	
Nom Gaz	Nom_Gaz	Texte	50	X	X	
Sigle Gaz	Sigle_Gaz	Texte	50	X	X	
Code Mesure	ID_Mesure	Nombre	4	0	X	
Valeur Mesure (en ppm)	Valeur_Mesure	Texte	50	X	X	
Code Agent	ID_Agent	Nombre	4	0	X	
Nom Agent	Nom_Agent	Texte	50	X	X	
Prenom Agent	Prenom_Agent	Texte	50	X	X	
Poste Agent	Poste_Agent	Texte	50	X	X	
Rôle Agent	Role_Agent	Texte	50	X	X	Le métier Agent RH, Agent Administratif, Agent Technique
Dernier diplôme acquis	Dernier_Diplome_Chef_Agent	Texte	50	X	X	
Code Rapport	ID_Rapport	Nombre	4	0	X	
Rapport complet Fichier	Texte_Rapport	Texte	50 000	X	X	
Nom_Rapport	Nom_Rapport	Texte	50	X	X	
Code Date	ID_Date	Nombre	4	0	X	
Date	Date	Texte	8	X	X	
Code Adresse	ID_Adresse	Nombre	4	0	X	
Adresse Postale	Code_Postal	Texte	50	X	X	
Rue	Rue	Texte	50	X	X	
Numéro Rue	Numero_Rue	Texte	50	X	X	
Région	Région	Texte	50	X	X	
Ville	Ville	Texte	50	X	X	

Tout d'abord, nous avons les données liées à l'agence

- Code agence : Identifiant de l'agence. Cette donnée sera notée ID\_agence dans la table
- Chefs-Lieux agence : Région où est située l'agence. Cette donnée sera notée Region\_Agence dans la table.

Ensuite, nous avons les données liées aux capteurs déployés :

- Code capteur : Identifiant du capteur, noté ID\_Capteur dans la table
- Lieu Capteur : Location du capteur, notée Lieu\_Capteur dans la table
- Etat Capteur : Etat du capteur, notifiant si le capteur est défaillant ou en bon état, noté Etat\_Capteur

Puis, nous avons les données liées aux gaz :

- Code Gaz : Identifiant du gaz, noté ID\_Gaz dans la table
- Type de Gaz : Type du gaz identifié par le capteur, noté Type\_Gaz\_Capteur dans la table
- Nom Gaz : Nom du gaz identifié par le capteur, noté Nom\_Gaz dans la table
- Sigle Gaz : Sigle (ou formule) liée au nom du gaz. Par exemple le sigle du protoxyde d'azote est N2O. Cette donnée est notée Sigle\_Gaz dans la table

Ensuite, nous avons les données liées aux différents types de mesures effectuées :

- Code Mesure
- Valeur Mesure : Valeur de la mesure, prise en ppm et notée Valeur\_Mesure dans la table

A la suite, nous avons les données reliées à l'agent :

- Code Agent : Identifiant de l'agent, noté ID\_Agent dans la table
- Nom Agent : Nom de famille de l'agent, noté Nom\_Agent dans la table
- Prénom Agent : Prénom de l'agent, noté Prenom\_Agent dans la table
- Poste Agent : Métier, poste que l'agent occupe dans l'agence, noté Poste\_Agent dans la table
- Rôle Agent : Rôle de l'agent au sein de l'agence, il peut être agent RH, Agent administratif, agent technique... etc. Il est noté Role\_Agent dans la table
- Dernier diplôme acquis : Représente le dernier diplôme que l'agent a obtenu, noté Dernier\_Diplome\_Chef\_Agent dans la table

Ensuite, nous avons les données liées aux rapports publiés :

- Code Rapport : Identifiant du rapport effectué, noté ID\_Rapport dans la table
- Rapport complet Fichier : Texte contenant le contenu du fichier du rapport en entier, noté Texte\_Rapport dans la table
- Nom Rapport : Nom du rapport, noté Nom\_Rapport dans la table

Puis, nous avons les données liées aux dates :

- Code Date : Identifiant de la date inscrite, noté ID\_Date dans la table
- Date : Représente la date que l'on souhaite inscrire, notée Date dans la table

Et pour finir, nous avons les données liées aux adresses :

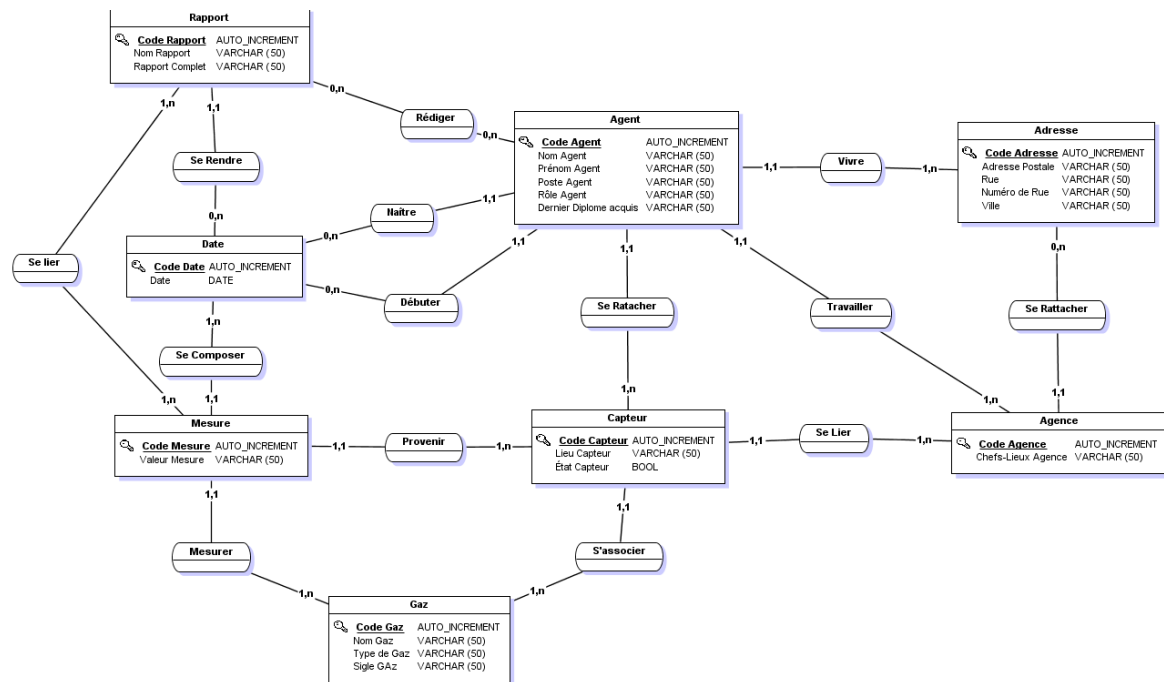
- Code Adresse : Identifiant lié a une adresse, noté ID\_Adresse dans la table
- Adresse Postale : Code postal de l'adresse, noté Code\_Postal dans la table
- Rue : Nom de la rue, noté Rue dans la table
- Numero Rue : Numero de rue de l'adresse, noté Numero\_Rue dans la table
- Région : Region de l'adresse, noté Region dans la table
- Ville : Ville de l'adresse, noté Ville dans la table

Chaque code permettra d'obtenir par la suite une table, c'est une donnée qui fait dépendre d'autres donnée, on les met donc en tête de colonne car elles sont importantes pour voir quelles données sont dépendantes.

Par exemple, le Code Agence permettra d'obtenir les chefs-lieux de l'Agence : Par la suite cela nous permettra de faire une table Agence composé d'un clé primaire Agence, et d'une autre colonne Chefs-Lieux-Agence.

	Code Agence	Code Capteur	Code Gaz	Code Mesure	Code Agent	Code Rapport	Code Date	Code Adresse
Code Agence	X							
Chefs-Lieux Agence	1							
Code Capteur		X						
Lieu Capteur		1						
État Capteur		1						
Code Gaz			X					
Type de Gaz			1					
Nom Gaz			1					
Sigle Gaz			1					
Code Mesure				X				
Valeur Mesure (en ppm)				1				
Code Agent					X			
Nom Agent					1			
Prenom Agent					1			
Poste Agent					1			
Rôle Agent					1			
Dernier diplôme acquis					1			
Code Rapport						X		
Rapport complet Fichier						1		
Nom_Rapport						1		
Code Date							X	
Date							1	
Code Adresse								X
Adresse Postale								1
Rue								1
Numéro Rue								1
Région								1
Ville								1

## Le modèle conceptuel de données et les explications



Nous avons créé, au total, 8 entités principales distinctes, car elles représentent des domaines de données identifiables et possèdent leur propre donnée caractéristique. Exemple : l'entité « Agent » possède ses propres données caractéristiques tel que « nom agent, poste agent, rôle agent, etc... » et possède plusieurs associations, notamment « rédiger, naitre, vivre, etc... »

Les associations, quant à elles, permettent de faire des liens logiques entre les différentes entités principales. Cela évite la redondance. Exemple « tel rapport peut être rédigé par tel agent et être rendu à tel date »

De plus, pour chaque entité, une forme normale du type « 3e forme normale » a été appliquée de tel sorte à ce que les attributs de chaque table possèdent une valeur atomique, que chaque attribut dépend entièrement de la clé primaire et qu'en plus de cela, ils ne dépendent pas entre eux. Exemple : dans l'entité « Agent », les données ne dépendent que de la donnée « Code Agent » qui est la clé primaire.

Par conséquent, chaque table possède une clé primaire qui est unique et permet de retrouver rapidement des données liées à celle-ci et de garantir l'unicité des données d'une entité.

Et enfin, les cardinalités permettent de déterminer les contraintes logiques tel que le minimum et le maximum d'une entité envers une autre. Exemple « un agent peut rédiger au minimum aucun rapport comme il peut au maximum en rédiger plusieurs. Et inversement, le rapport peut être rédigé au minimum par 1 agent et au maximum par plusieurs d'entre eux ».

Ce MCD, pour qu'il soit optimisé est théoriquement bon doit respecter certaines normes que l'on va expliquer :

### **1re forme normale (1FN) - Atomicité :**

Chaque attribut contient une valeur atomique (indivisible). Chaque attribut doit donc être clair, et être composé d'une seule valeur.

Ex : Nous ne pouvons pas avoir une colonne nom ET prénom --> [Dutertre,Lou] il faut que chaque donnée soit unique : nom --> [Lou] ET prénom --> [Dutertre]

### **2e forme normale (2FN) - Dépendance complète :**

Tous les attributs dépendent entièrement de la clé primaire.

Grossièrement, chaque entité doit posséder une clé primaire, c'est à dire une donnée qui permettra d'accéder à la future table, et de caractériser celle-ci.

Elle doit être unique pour chaque entité du MCD

### **3e forme normale (3FN) - Pas de dépendance transitive**

Les attributs ne dépendent que de la clé primaire, pas entre eux.

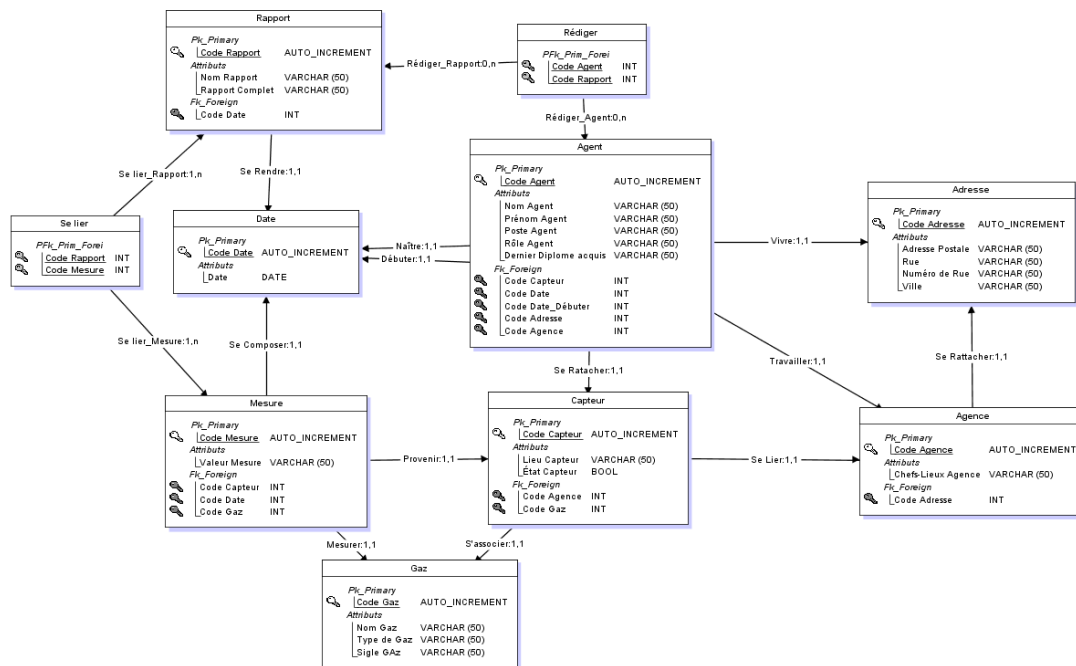
Quand nous ferons les liens entre les tables, il faudra que les jointures se fassent de deux manières différentes.

--> Clé primaire relié à une clé étrangère

--> ou 2 clé primaire qui en forme une nouvelle

Dans les deux cas, ce sont les clefs qui feront les clients pas les attributs tierces de la table.

## **Le modèle logique de données et les explications**



Tout d'abord, chaque entité maintenant devenue des Tables, possèdent des clés primaires AUTO\_INCREMENT afin de faciliter la gestion et pour garantir l'unicité des enregistrements de données. Cela permet de retrouver et d'enregistrer des données simplement, rapidement et efficacement.

Ensuite, nous avons les associations tel que « rédiger, vivre, se lier, etc. » qui deviennent à présent des relations composées de clés primaires des tables qu'elles relient. Cela permet de modéliser des liens et correspondances multiples entre les tables de manière simple et efficace.

Les clés dites étrangères, quant à elles, traduisent les cardinalités et permettent de s'assurer qu'une relation entre deux tables est valide et correct. Exemple : « Code Capteur » dans « Mesure » montre que la mesure enregistrée provient d'un capteur existant.

On utilise aussi des types de données tel que « VARCHAR (50) » ou « INT » afin de préciser la nature des données et de ne pas avoir d'erreurs dans notre base. Exemple : « INT » pour les identifiants (AUTO\_INCREMENT) et « VARCHAR (50) pour des chaînes » de caractères

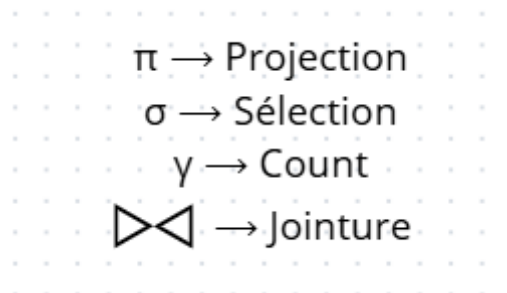
## Le modèle physique de données

Cela correspond au script que nous devons écrire pour créer la base de données, vous la retrouverez dans le fichier ZIP.



## Les arbres algébriques de chaque requête

- Votre demande nécessitait que certaines données, questions, puissent être obtenable dans la base de données par la suite. Ces arbres algébriques pourront être traduit par la suite en commande SQL pour obtenir les données recherchées directement depuis la base de données concernée
- Voici tous les opérateurs algébriques que nous allons utiliser :



La projection correspond à récupérer une colonne de données avec une en-tête en particulier (ex : si une colonne s'appelle nom, alors je peux projeter la colonne nom pour récupérer toutes les données de cette colonne)

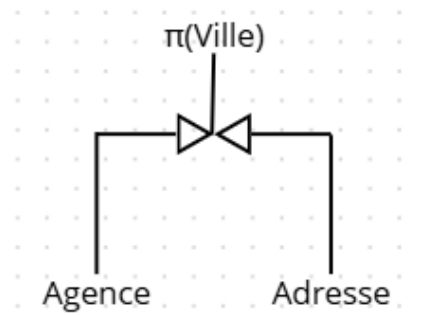
La sélection est une condition des données (ex : Si je veux le nom de famille DonJuan dans la colonne nom)

Le "Count" sert à compter le nombre de données dans la colonne.

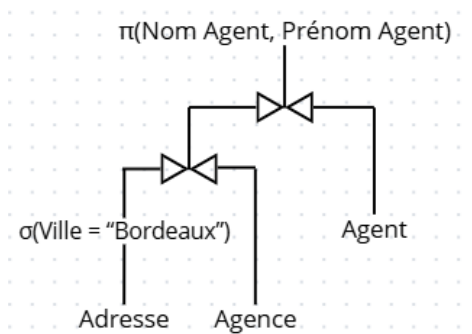
La jointure sert à relier 2 tables entre-elles.

Voici les 7 données que nous devons récupérer :

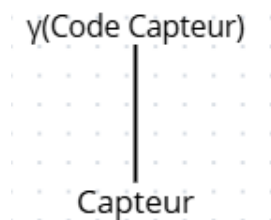
- 1. Liste de l'ensemble des agences :



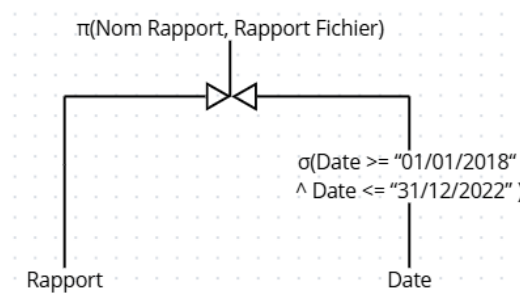
- 2. Liste de l'ensemble du personnel technique de l'agence de Bordeaux :



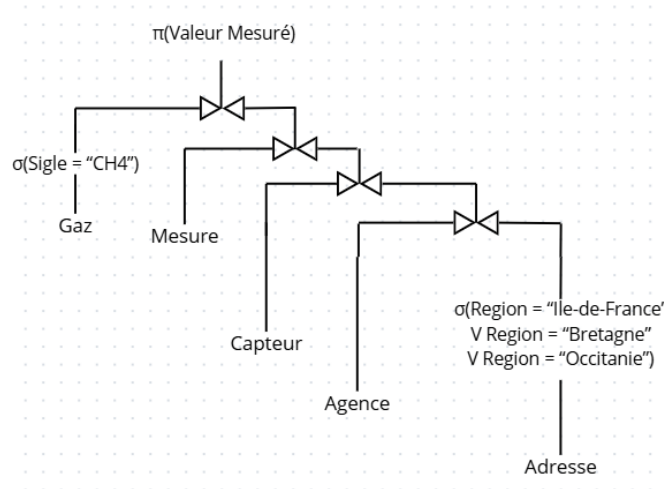
- 3. Nombre total de capteurs déployés :



- 4. Liste des rapports publiés entre 2018 et 2022

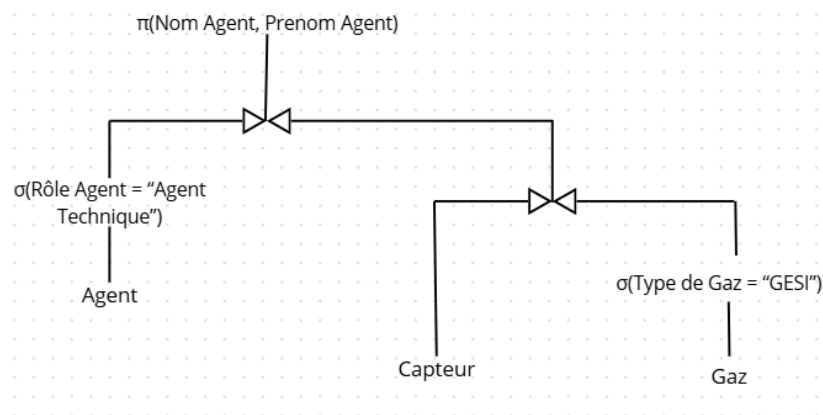


- 5. Afficher les concentrations de CH<sub>4</sub> (en ppm) dans les régions « Ile-de-France », « Bretagne » et « Occitanie » en mai et juin 2023.

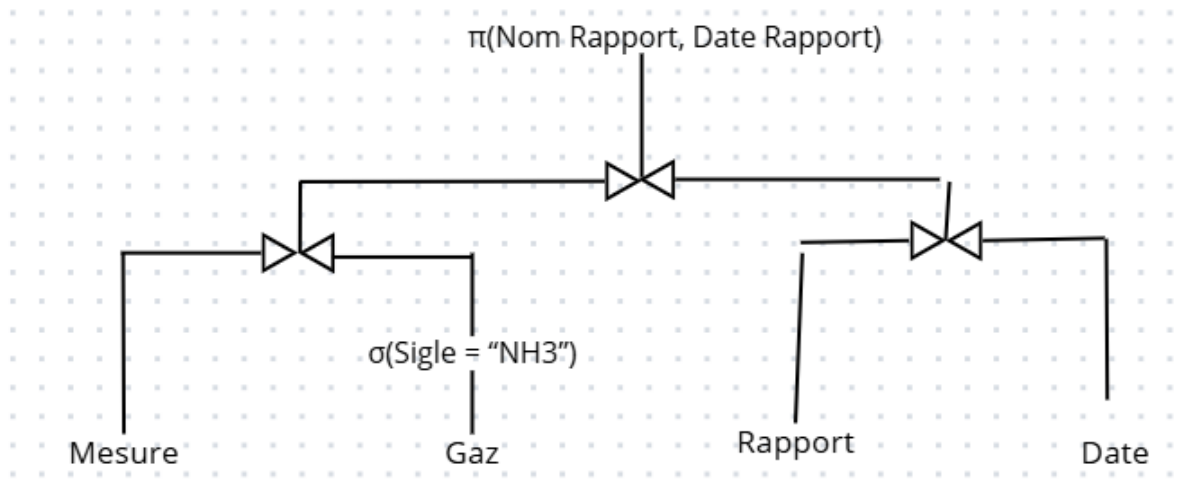


Cette commande sera un peu plus longue en termes de jointure puisque la donnée des régions se situe dans l'adresse et est assez loin de la table mesure.

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



- 7. Titres et dates des rapports concernant des concentrations de NH<sub>3</sub>, classés par ordre antichronologique.

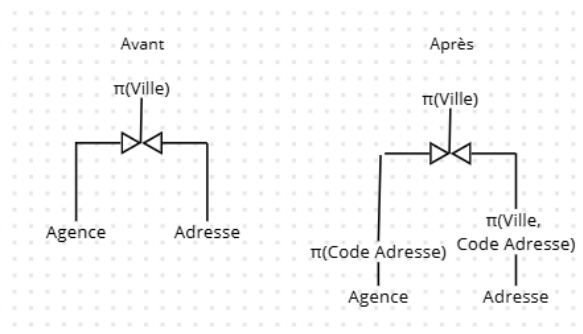


Ici l'ordre anti-chronologique se traduira en SQL par un ORDER BY Date\_Rapport DESC.

## OPTIMISATION :

Nous pouvons optimiser tous ces arbres algébriques en ne sélectionnant que les colonnes réellement nécessaires (clé des jointures et données souhaité) et en faisant les sélections le plus tôt possible :

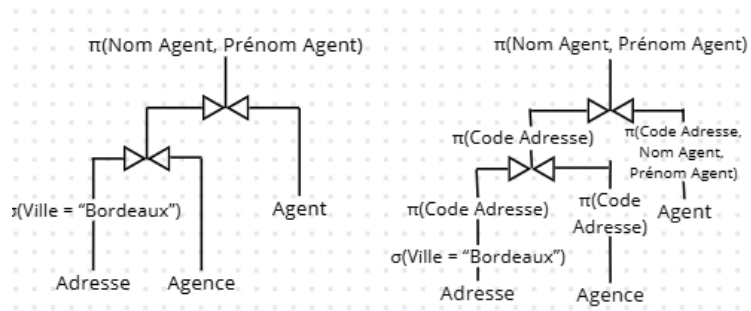
1)



On projette seulement les colonnes utiles (Ville, et les clefs primaires/étrangères pour les jointures)

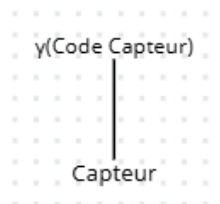
Cela est plus optimisé car on ne va pas prendre toute la table pour la jointure, le travail nécessaire sera donc moins grand.

2)

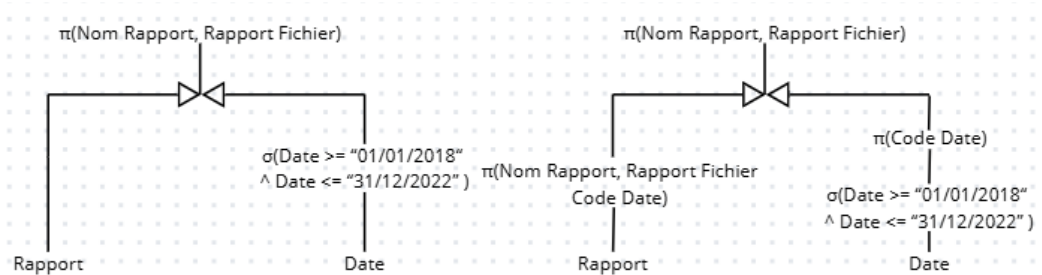


Le Code Adresse est utilisé plusieurs fois pour relier ces tables

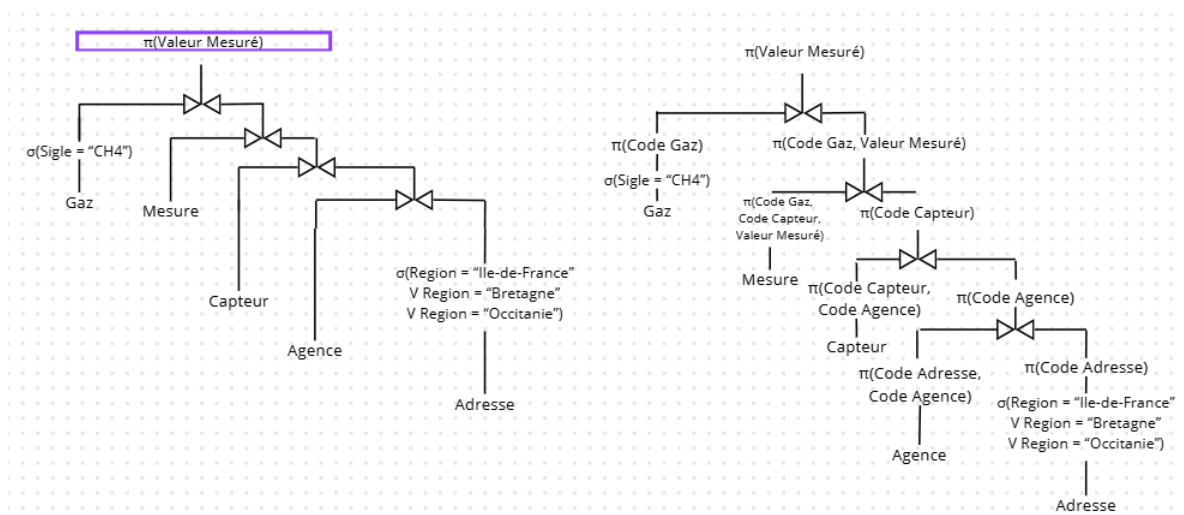
3)



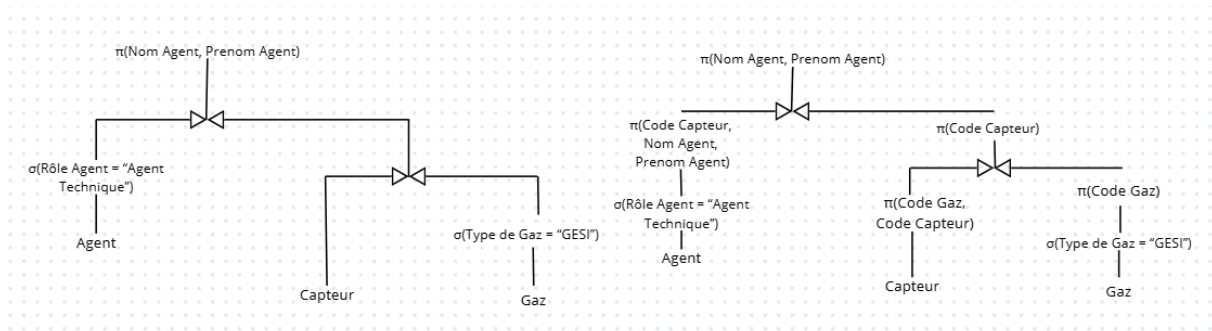
4)



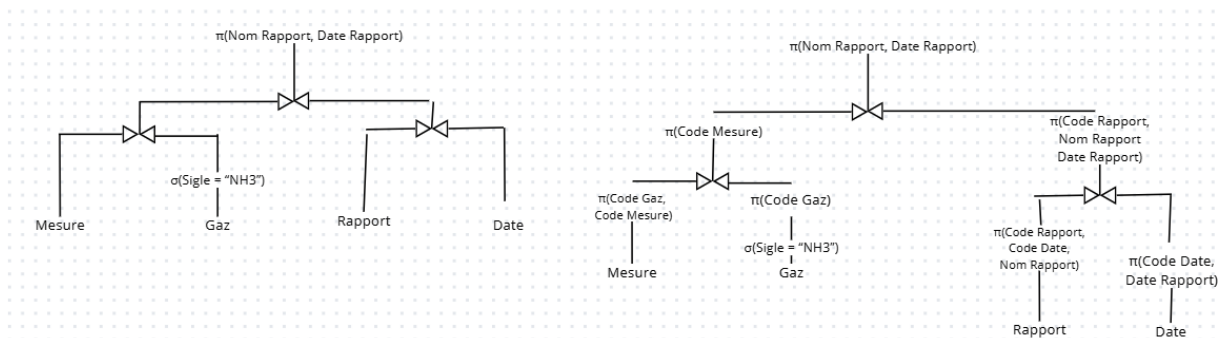
5)



6)



7)



Voici les arbres les plus optimisé et qui demanderont le moins de puissance possible, cela sera notamment utile lorsque la base de données sera immense, et que les données recherchées seront énormes.

## En conclusion :

Nous avons choisi d'axer la base de données sur un modèle avec le moins de redondance possible.

Les données que vous nous avez demandées sont renseigné, et trier de manière à rendre plus fluide la gestion de la base de données par la suite.