



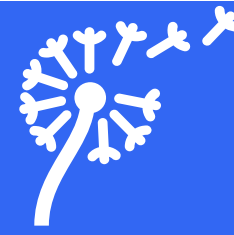
- PEREZ Ethan
- NASREDDINE BADRAN Faisal
- SUSINI Luigi
- NOYER Ilouan

# Sommaire



- Contexte / Enjeux
- Problématique
- Construction du MCD/MLD
- Arbres Algébriques
- Construction de BDD sur MySQL
- Peuplement
- Les Requêtes
- Compte Utilisateur
- Conclusion

# Contexte et Enjeux



- GIEC
- Ministère de la Transition Écologique
- Qualité de l'air
- Projet national
- Pollution de l'air
- Réchauffement climatique

# Problématique

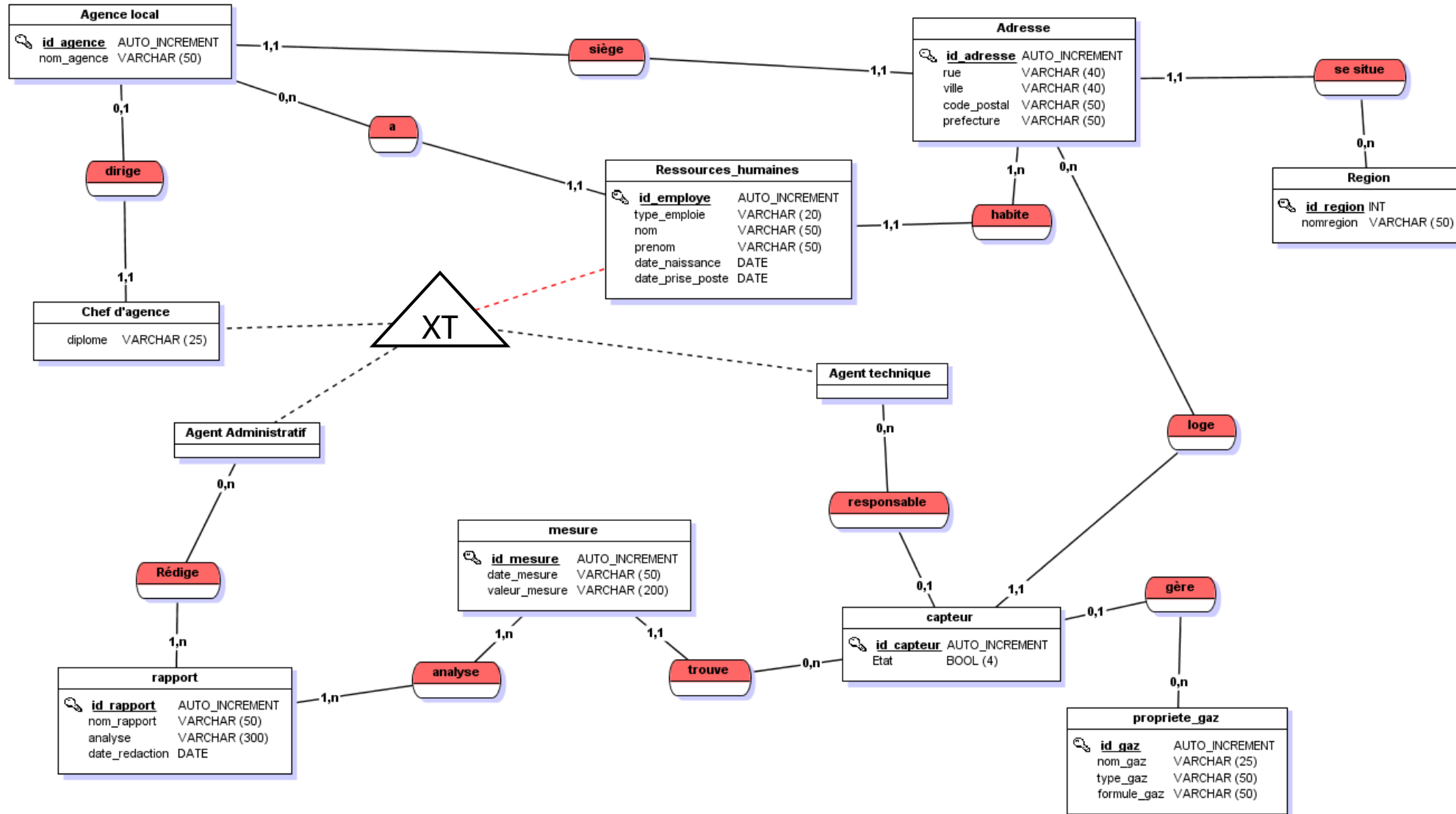


**Comment concevoir une base de données centralisée pour exploiter les données de qualité de l'air, tout en assurant la gestion des agences, des rapports et la securite des données ?**

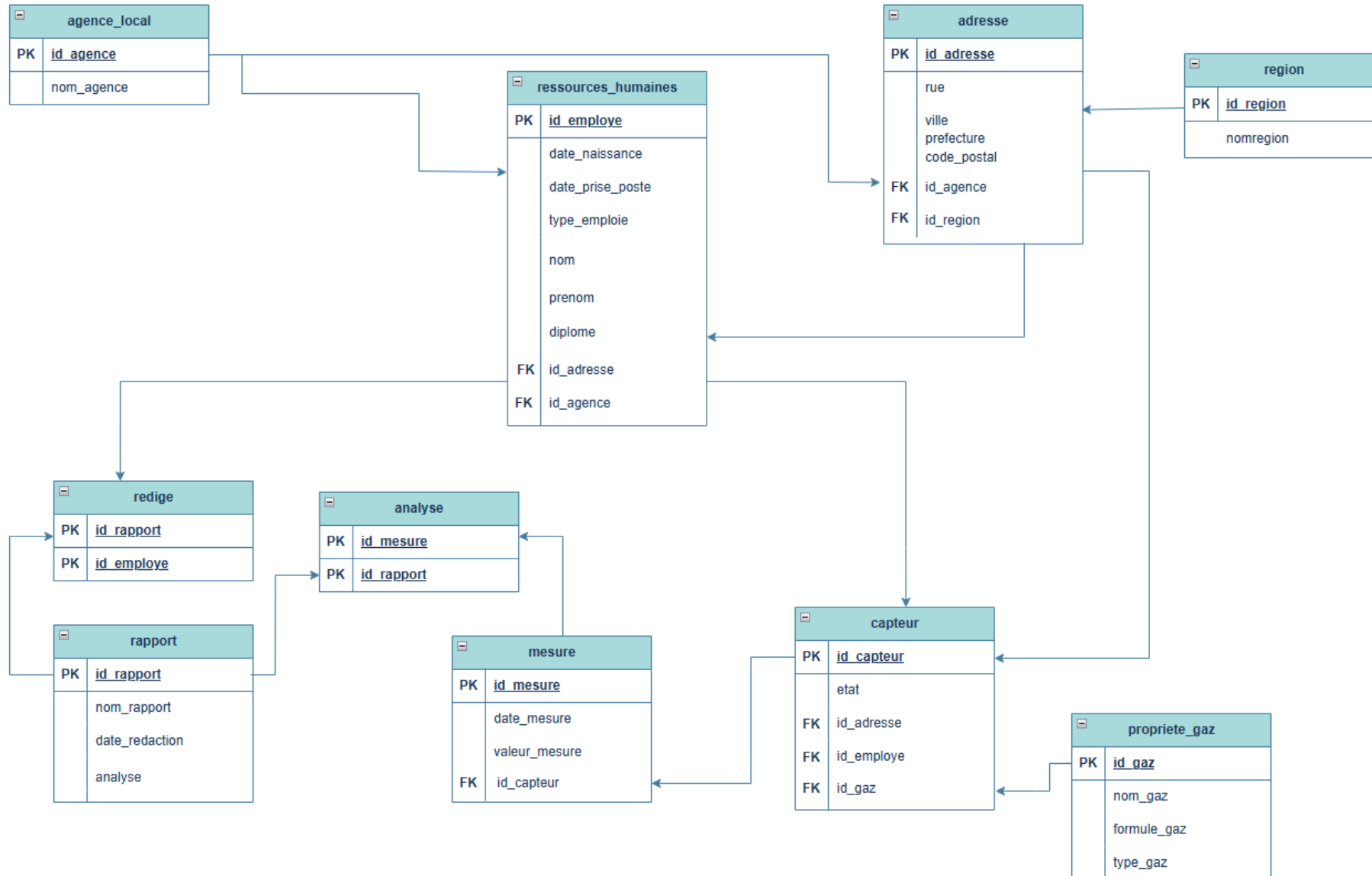


# Construction du MCD/MLD

SQL

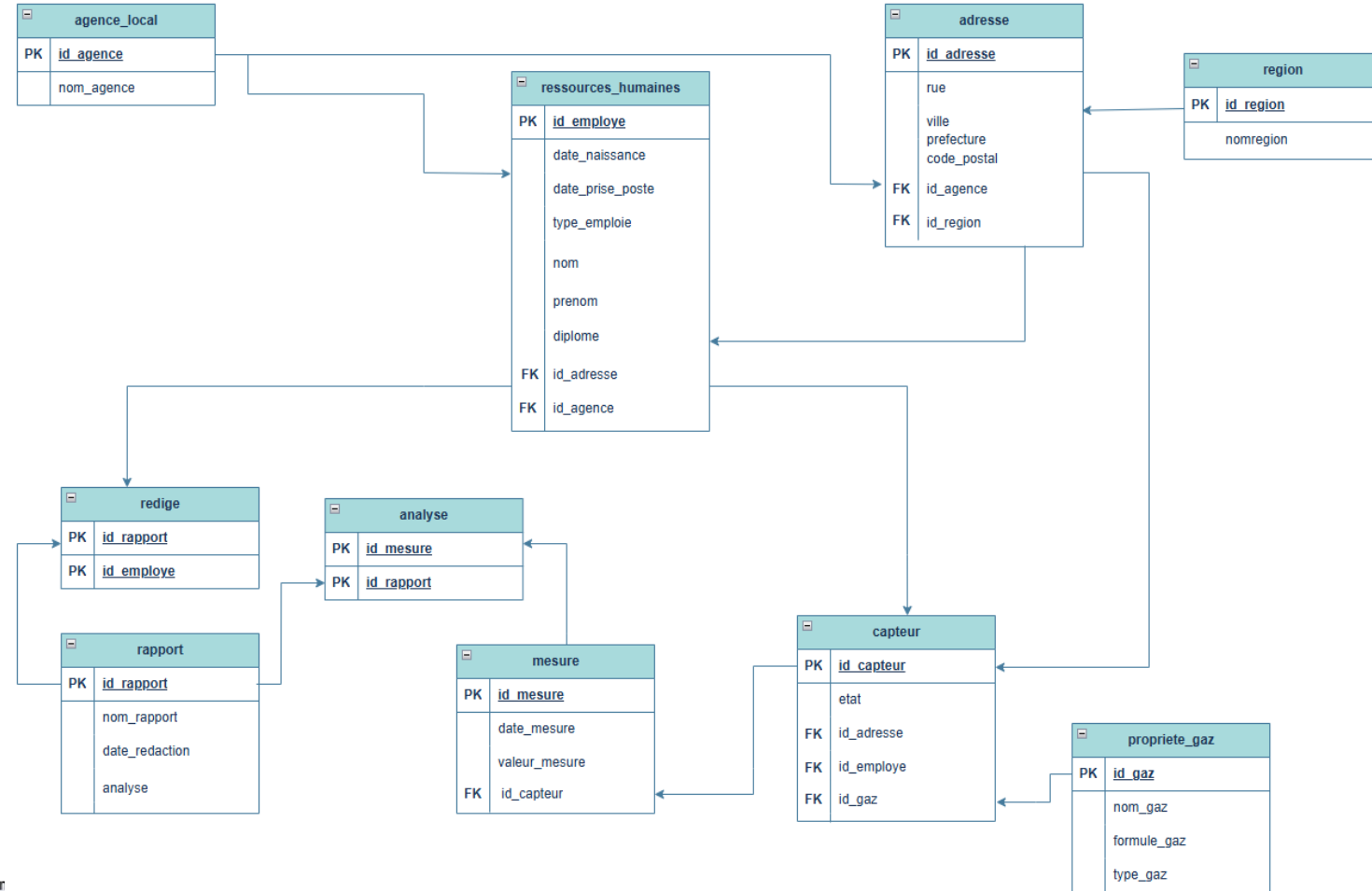
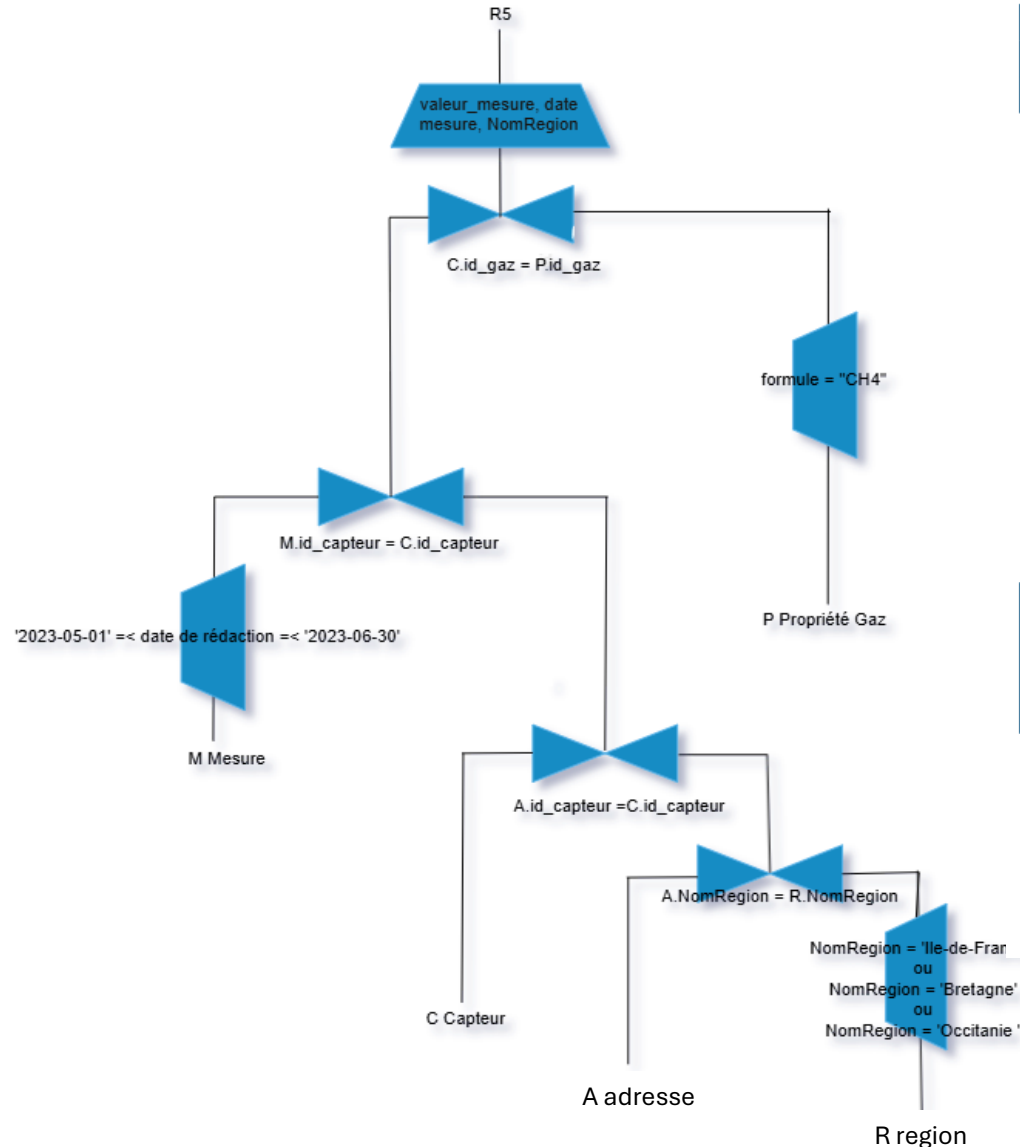


# Construction du MCD/MLD



# Arbres Algébriques

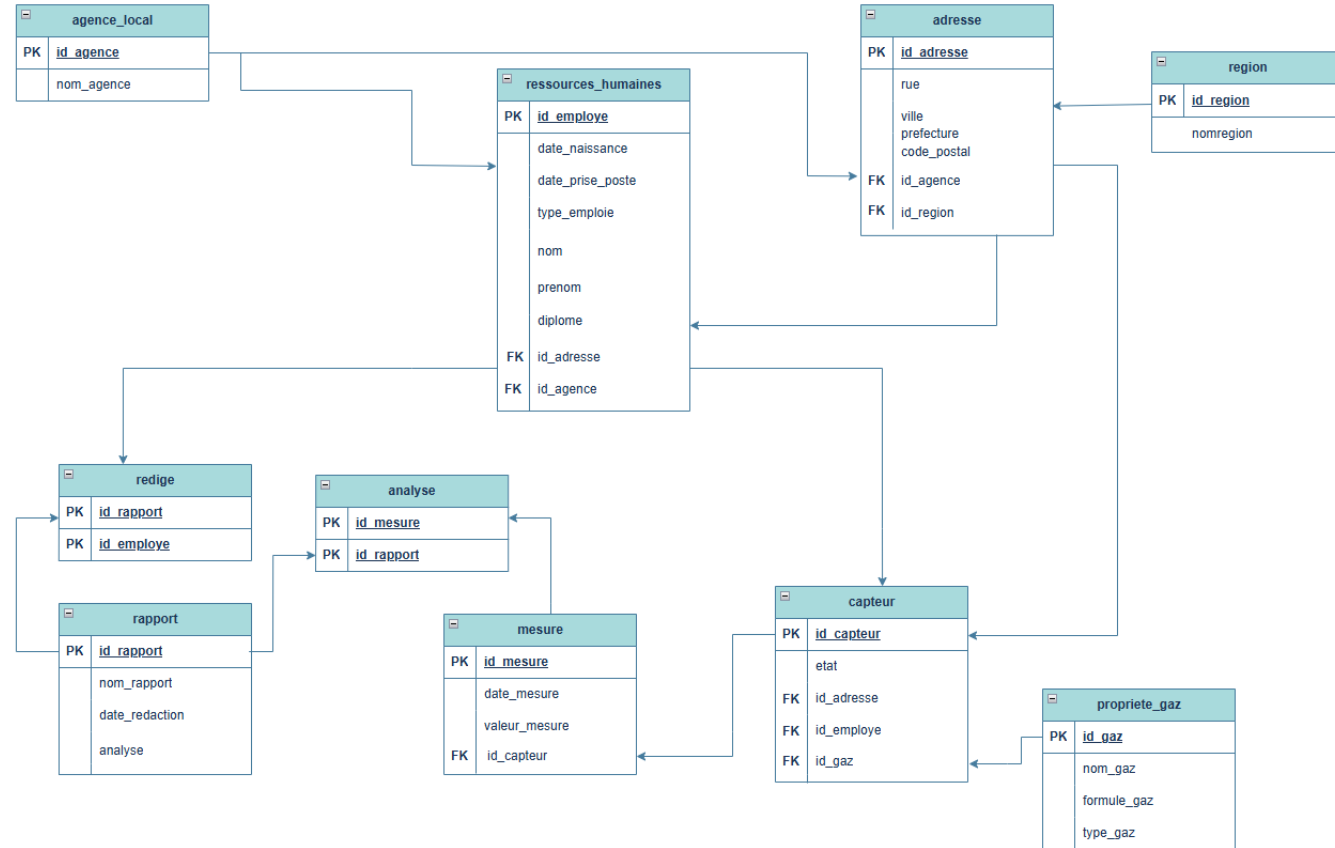
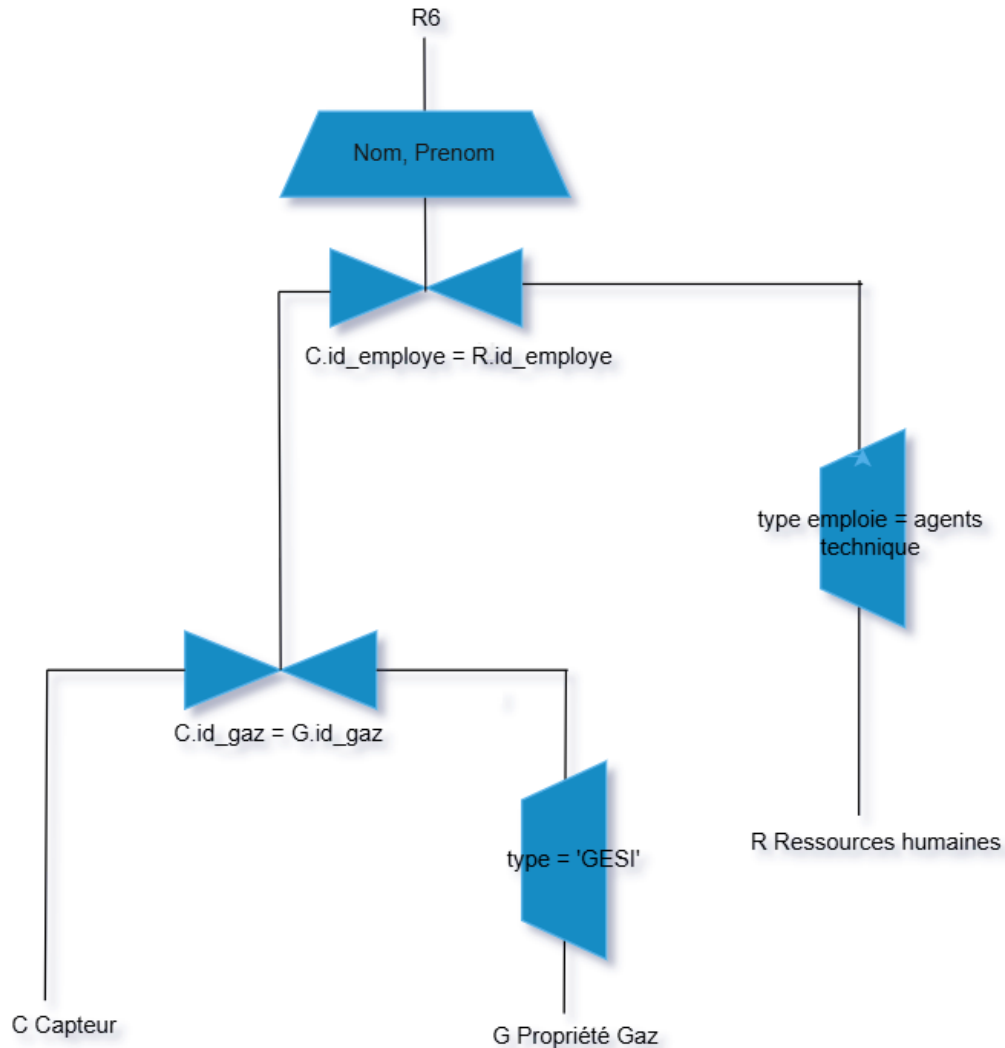
SQL



R5 = Afficher les concentrations de CH4 (en ppm) dans les régions « Ile-de-France », « Bretagne » et « Occitanie » en mai et juin 2023.<sup>7</sup>

# Arbres Algébriques

SQL



R6 = Liste des noms des agents techniques maintenant des capteurs concernant les gaz à effet de serre provenant de l'industrie (GESI).



# Construction de la BDD

The logo features a blue cylinder with the letters 'SQL' in white, and a blue cloud shape at the bottom right.

```
CREATE TABLE IF NOT EXISTS adresse(  
    id_adresse INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    rue VARCHAR(46),  
    ville VARCHAR(45),  
    code_postal VARCHAR(5),  
    prefecture VARCHAR(50),  
    id_agence INT,  
    FOREIGN KEY(id_agence) REFERENCES agence_local(id_agence),  
    id_region INT,  
    FOREIGN KEY(id_region) REFERENCES region(id_region)  
);
```

```
CREATE TABLE IF NOT EXISTS ressources_humaines(  
    id_employe INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    date_naissance DATE,  
    date_prise_poste DATE,  
    type_emploi ENUM("Chef_Agence", "Agent_Administratif", "Agent_Technique"),  
    nom VARCHAR(50),  
    prenom VARCHAR(50),  
    diplome TEXT,  
    id_adresse INT,  
    FOREIGN KEY (id_adresse) REFERENCES adresse(id_adresse),  
    id_agence INT,  
    FOREIGN KEY (id_agence) REFERENCES agence_local(id_agence)  
);
```

# Peuplement de la BDD



SQL

```
# Générer des adresses aléatoires
def generate_addresses(n=30):
    addresses = []
    for id_adresse in range(94, n + 1): # De 1 à n inclus # ici les id sont éditables

        agence = random.choice(list(agence_id.keys()))
        region = agence
        if agence == 14:
            region = 10

        departement = random.choice(regions_departements[region_id[region]])
        ville = random.choice(departement_villes[departement])
        prefecture = departement_prefectures[departement]
        rue = f'{random.choice(rues_exemples)} {random.randint(1, 300)}'
        code_postal = f'{departement}{random.randint(0, 999):03d}'

        #id_agence = region # même valeur que la région
        addresses.append((id_adresse, rue, ville, code_postal, agence, region, prefecture))

    return addresses

# Générer fichier .sql
def generate_sql_file(addresses, filename="insert_adresses2_2.sql"):
    with open(filename, "w", encoding="utf-8") as f:
        for addr in addresses:
            f.write(
                f'INSERT INTO adresse (id_adresse, Rue, Ville, Code_postal, id_agence, id_region, prefecture) VALUES '
                f'({addr[0]}, "{addr[1]}", "{addr[2]}", "{addr[3]}", "{addr[4]}", {addr[5]}, "{addr[6]}");\n'
            )
    return filename

# Exécuter la génération
addresses = generate_addresses(103)
sql_file_path = generate_sql_file(addresses)
sql_file_path
```

```
DELIMITER $$

CREATE PROCEDURE InsertGaz (
    IN p_nom_gaz VARCHAR(60),
    IN p_formule_gaz VARCHAR(60),
    IN p_type_gaz VARCHAR(60)
)
BEGIN
    INSERT INTO propriete_Gaz (nom_gaz, formule_gaz, type_gaz)
    VALUES (p_nom_gaz, p_formule_gaz, p_type_gaz);
END $$

DELIMITER ;

CALL InsertGaz("Protoxyde d'azote", 'N2O', 'GES');
CALL InsertGaz("Ozone troposphérique", 'O3', 'GES');
CALL InsertGaz("Méthane", 'CH4', 'GES');
CALL InsertGaz("Hydrofluorocarbures", 'HFC', 'GESI');
CALL InsertGaz("Hexafluorure de soufre", 'SF6', 'GESI');
CALL InsertGaz("Dioxyde de carbone", 'CO2', 'GES');
```



# Les Requêtes

The logo features a blue database cylinder with the letters 'SQL' in white, positioned next to a blue cloud icon.

R5

```
SELECT M.valeur_mesure, M.date_mesure, nomregion
FROM mesure M
JOIN Capteur C ON M.id_capteur = C.id_capteur
JOIN Adresse A ON A.id_adresse = C.id_adresse
JOIN Propriete_Gaz P ON C.id_gaz = P.id_gaz
JOIN region R ON R.id_region = A.id_region
WHERE formule_gaz = 'CH4'
      AND date_mesure BETWEEN '2023-05-01' AND '2023-06-30'
      AND nomregion IN ('Ile-de-France', 'Bretagne', 'Occitanie');
```

R11

```
DELIMITER //
CREATE PROCEDURE req11 (IN nomgaz VARCHAR(22))
BEGIN
SELECT nom_gaz, nom_rapport
FROM mesure
JOIN capteur ON capteur.id_capteur = mesure.id_capteur
JOIN propriete_gaz ON propriete_gaz.id_gaz = capteur.id_gaz
JOIN analyse ON analyse.id_mesure = mesure.id_mesure
JOIN rapport ON rapport.id_rapport = analyse.id_rapport
WHERE nom_gaz = nomgaz;
END //
DELIMITER ;

CALL req11("Hydrofluorocarbures");
```

R6

```
SELECT DISTINCT R.Nom, R.Prenom
FROM ressources_humaines R
JOIN capteur C ON C.id_employe = R.id_employe
JOIN propriete_gaz G ON G.id_gaz = C.id_gaz
WHERE R.type_emploi = 'Agent_technique' AND G.type_gaz = 'GESI';
```

# Compte utilisateur

The logo features a blue cylinder with the letters 'SQL' in white, and a smaller blue cloud-like shape below it.

```
CREATE USER 'user'@'localhost' IDENTIFIED BY '1234';  
GRANT SELECT ON cleardata.* TO 'user'@'localhost';  
  
SELECT user, host FROM mysql.user;  
SHOW GRANTS FOR 'user'@'localhost';
```

Création d'un utilisateur avec des permissions réduite pour la démonstration sur le Workbench.

# Conclusion



Les avantages de notre BDD :

- Optimisée
- Editable
- Cohérente
- Utilisable en tant qu'utilisateur

MERCI pour votre  
attention !