

Livrable 2 - Bloc 06 :

Equipe 14 :

- BLANC Thomas
- DUTERTRE-THOUAN Lou
- KADIRI Shahinèze
- OUACH Bilal

1. Introduction

Dans le cadre du projet national sur la qualité de l'air confié à ClearData par le ministère de la transition écologique, nous avons conçu une base de données relationnelle destinée à centraliser les données issues des agences locales.

Dans ce dernier contre-rendu, vous trouverez le peuplement fictif de la base de donnée ainsi que toutes les demandes que vous nous aviez fait.

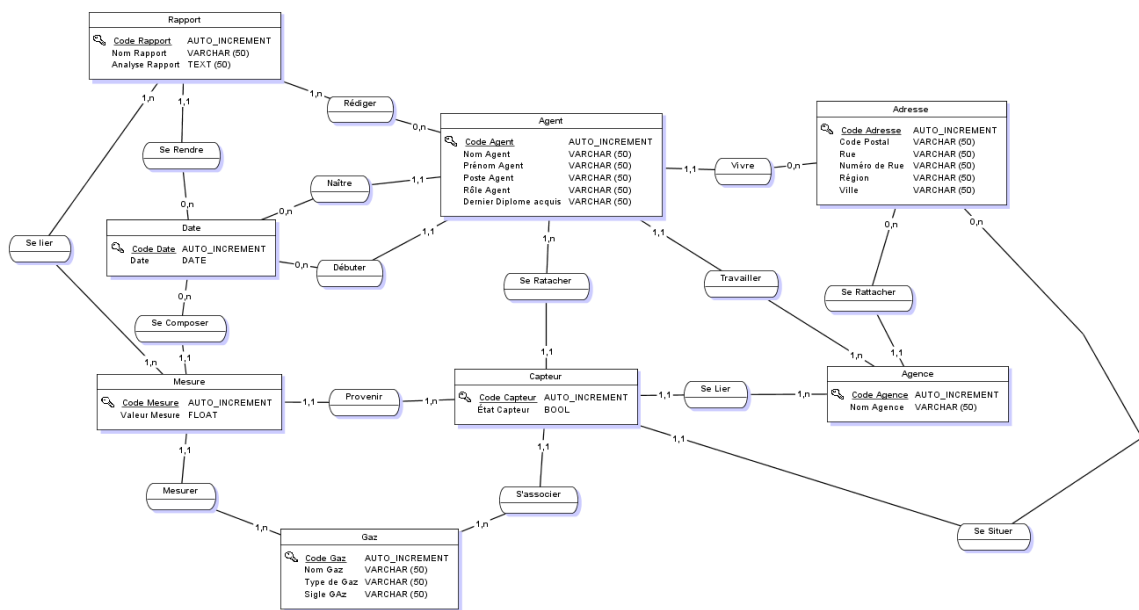
2. Sommaire

- (Une mise à jour du MCD/MPD/MLD de notre équipe)
- Notions théoriques
- Les requêtes demandées (en langage SQL)
- Des explications sur la génération et le peuplement de la base de données finale (origine des données, difficultés rencontrées,
- Les fonctionnalités présentes ou absentes par rapport à la demande et aux spécifications techniques.
- Conclusion

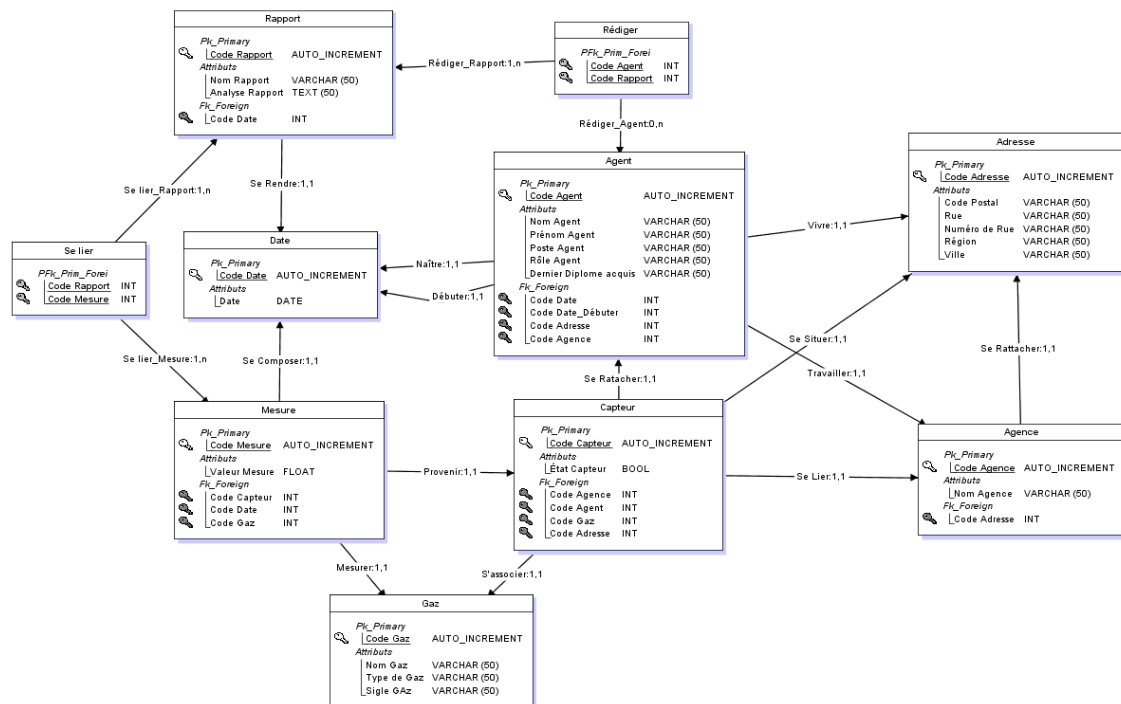
3. Nouveau MCD

Après regroupement de notre équipe, certaines chose dans la base de données était manquante ou manquaient de clarté c'est pour cela que nous l'avons refait, d'une manière à respecter vos demandes :

MCD :



MLD :



4. Notions théoriques

Compte utilisateur : Dans le cadre de ce projet, 2 types de compte sont demandés :

- Compte admin :
 - Accès complet à la BDD
 - Peut créer, modifier et supprimer des données/tables/utilisateurs.
 - Réservé à l'équipe technique ou aux administrateurs systèmes
- Compte user :
 - Accès restreint
 - Ne peut ni modifier la structure de la base, ni accéder à certaines informations sensibles (comme les données RH)

Moteur de stockage : Tout d'abord, un moteur de stockage est le composant qui gère la manière dont les données sont enregistrées, lues et manipulées dans une base MYSQL, il en existe 2 principaux :

- InnoDB

- MyISAM

Moteur	Avantages	Inconvénients	Adapté si...
InnoDB	Supporte les transactions, les contraintes d'intégrité (clé étrangère), la concurrence entre utilisateurs	Un peu plus lourd en écriture	On a besoin de relations fortes, de sécurité et de modifications fréquentes
MyISAM	Très rapide en lecture, léger	Pas de support des clés étrangères ni des transactions	On fait surtout de la lecture (peu d'inserts/updates)

Optimisation des requêtes :

Pour améliorer la rapidité et l'efficacité des requêtes, plusieurs bonnes pratiques ont été appliquées :

- Utilisation des jointures bien structurées pour relier les tables
- Mise en place d'index sur les colonnes les plus utilisées dans les filtres (WHERE) ou tris (ORDER BY)
- Vérification régulière des performances via des test sur des jeux de données réalistes

Aspects juridiques : Pour ce projet sur la qualité de l'air, certains aspects juridiques doivent être respectés, notamment en lien avec la protection des données personnelles

Obligation	Application dans le projet
Respect du RGPD (Règlement Général sur la Protection des Données)	Masquer les données sensibles, pas de numéros de sécu, identifiants internes
Sécurisation de l'accès	Comptes limités, contrôle des droits (comptes admin/user)
Données immuables	Verrouillage des rapports une fois créés

Manipulation des données :

La base de données a été conçue pour permettre facilement les actions suivantes :

- Ajouter, modifier ou supprimer :
 - des agences
 - des employés
 - des capteurs
- Pour les relevés de capteurs : On peut ajouter ou supprimer un relevé mais pas le modifier pour respecter la fiabilité des mesures
- Pour les rapports : On peut en créer mais pas les modifier ni les supprimer une fois qu'ils sont enregistrés
- Des commandes SQL simples sont prévues pour toutes ces opérations (ex : INSERT, DELETE, UPDATE)

5. Contraintes

- Données non redondantes et cohérentes qu'on peut consulter, ajouter et supprimer facilement (dans les limites des demandes).
- Fort degré d'indépendance entre données et matériel (modèle/SGBD utilisé).

- Fort degré d'indépendance entre données et programmes sécurité (pour la création et l'accès à la base de données).
- Il faut définir les contraintes d'intégrité de l'ensemble de la base de données, sur les colonnes et les tables.
- Données réalistes quant aux régions et aux villes.
- Pour les dates: se baser sur un intervalle entre janvier 2017 et décembre 2024.

6. Requêtes demandées

Commençons par faire un rappel des requêtes qui nous sont demandées :
Résumés de données souhaités

- Liste de l'ensemble des agences

```
SELECT Nom_Agence FROM agence;
```

- Liste de l'ensemble du personnel technique de l'agence de Bordeaux

```
SELECT
    Nom_Agent,
    Prenom_Agent
FROM agent
JOIN agence ON agent.Code_Adresse = agence.Code_Adresse
JOIN adresse ON agence.Code_Adresse = adresse.Code_Adresse
WHERE Ville = "Bordeaux";
```

- Nombre total de capteurs déployés.

```
SELECT COUNT(Code_Capteur) AS Nbr_Capteur
FROM capteur;
```

- Liste des rapports publiés entre 2018 et 2022

```
SELECT Nom_Rapport,date
FROM rapport
JOIN date ON rapport.Code_Date = date.Code_Date
WHERE date BETWEEN "2018-01-01" AND "2022-12-31";
```

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

```
SELECT Valeur_Mesure,Date /*,Sigle_Gaz,Region*/
FROM mesure
JOIN gaz ON mesure.Code_Gaz = gaz.Code_Gaz
JOIN capteur ON mesure.Code_Capteur = capteur.Code_Capteur
JOIN adresse ON capteur.Code_Adresse = adresse.Code_Adresse
JOIN date ON mesure.Code_Date = date.Code_Date
WHERE Sigle_GAz = "CH4"
AND (Region = "Île-de-France"
    OR Region = "Bretagne"
    OR Region = "Occitanie")
AND Date BETWEEN '2023-05-01' AND '2023-06-30';
```

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

```
SELECT DISTINCT Nom_Agent,Prenom_Agent
FROM agent
JOIN capteur ON agent.Code_Agent = capteur.Code_Agent
JOIN gaz ON capteur.Code_Gaz = gaz.Code_Gaz
WHERE Type_de_Gaz = "GESI"
AND Poste_Agent = "Agent Technique";
```

- Titres et dates des rapports concernant des concentrations de NH3, classés par ordre anti-chronologique.

```
SELECT Nom_Rapport,date.Date/*,Sigle_Gaz*/
FROM rapport
JOIN date ON rapport.Code_Date = date.Code_Date
JOIN se_liier ON rapport.Code_Rapport = se_liier.Code_Rapport
JOIN mesure ON se_liier.Code_Mesure = mesure.Code_Mesure
JOIN gaz ON mesure.Code_Gaz = gaz.Code_Gaz
WHERE Sigle_Gaz = "NH3"
ORDER BY date.Date DESC;
```

- Afficher le mois où la concentration de PFC a été la moins importante pour chaque région.

```
SELECT
    adresse.Region,
    MONTH(date.Date) AS Mois,
/* MONTH() --> Fonction pour obtenir le mois d'une date */
    mesure.Valeur_Mesure
FROM mesure
JOIN gaz ON mesure.Code_Gaz = gaz.Code_Gaz
JOIN capteur ON mesure.Code_Capteur = capteur.Code_Capteur
JOIN adresse ON capteur.Code_Adresse = adresse.Code_Adresse
JOIN date ON mesure.Code_Date = date.Code_Date
WHERE Sigle_Gaz = "PFC"
AND mesure.Valeur_Mesure IN (
/*On récupère la valeur minimale mesuré des régions pour ensuite les comparer avec le PFC*/
    SELECT MIN(mesure.Valeur_Mesure)
    FROM mesure
    JOIN gaz ON mesure.Code_Gaz = gaz.Code_Gaz
    JOIN capteur ON mesure.Code_Capteur = capteur.Code_Capteur
    JOIN adresse ad ON capteur.Code_Adresse = ad.Code_Adresse
    WHERE gaz.Sigle_Gaz = 'PFC'
    AND ad.Region = adresse.Region /* On récupère uniquement les adresses de la requête principale */
);
```

- Moyenne des concentrations (en ppm) dans la région « Ile-de-France » en 2020, pour chaque gaz étudié.

```
SELECT gaz.Nom_Gaz,ROUND(AVG(mesure.Valeur_Mesure),2) AS Moyenne_Gaz_Mesure
FROM mesure
JOIN gaz ON mesure.Code_Gaz = gaz.Code_Gaz
JOIN capteur ON mesure.Code_Capteur = capteur.Code_Capteur
JOIN adresse ON capteur.Code_Adresse = adresse.Code_Adresse
JOIN date ON mesure.Code_Date = date.Code_Date
WHERE adresse.Region = "Île-de-France"
AND EXTRACT(YEAR FROM date.Date) = 2020
GROUP BY gaz.Nom_Gaz,adresse.Region;
```

- Taux de productivité des agents administratifs de l'agence de Rouen (le taux est calculé en nombre de rapports écrits par mois en moyenne, sur la durée de leur contrat)

```

SELECT
agent.Nom_Agent,
agent.Prenom_Agent,
COUNT(DISTINCT rapport.Code_Rapport) AS Nb_Rapports,
TIMESTAMPDIFF(MONTH, dd.Date, MAX(dr.Date)) AS Duree_Contrat_Mois,
ROUND(COUNT(DISTINCT rapport.Code_Rapport) / TIMESTAMPDIFF(MONTH, dd.Date, MAX(dr.Date)), 2) AS Taux_Productivite
FROM agent
JOIN agence ON agent.Code_Agence = agence.Code_Agence
JOIN adresse ON agence.Code_Adresse = adresse.Code_Adresse
JOIN date dd ON agent.Code_Date_Debuter = dd.Code_Date /*On doit séparer les deux table date pour pouvoir ré
JOIN rediger ON agent.Code_Agent = rediger.Code_Agent
JOIN rapport ON rediger.Code_Rapport = rapport.Code_Rapport
JOIN date dr ON rapport.Code_Date = dr.Code_Date
WHERE adresse.Ville = 'Rouen'
AND agent.Poste_Agent = 'Agent administratif'
GROUP BY agent.Code_Agent
ORDER BY Taux_Productivite DESC;

```

- Pour un gaz donné, liste des rapports contenant des données qui le concernent (on doit pouvoir donner le nom du gaz en paramètre)

```

DELIMITER $

CREATE PROCEDURE GetRapportsByGaz(Sigle_Gaz VARCHAR(50))
BEGIN
    SELECT rapport.Nom_Rapport
    FROM rapport
    JOIN se_liier ON rapport.Code_Rapport = se_liier.Code_Rapport
    JOIN mesure ON se_liier.Code_Mesure = mesure.Code_Mesure
    JOIN gaz ON mesure.Code_Gaz = gaz.Code_Gaz
    WHERE gaz.Sigle_Gaz = Sigle_Gaz;
END $

DELIMITER ;

CALL GetRapportsByGaz('CO2');
CALL GetRapportsByGaz('NH3');

```

La procédure est à créer une seule et unique fois dans la base de donnée puis nous avons juste à appeler cette procédure (comme un fonction en python).

- Liste des régions dans lesquelles il y a plus de capteurs que de personnel d'agence.

```

SELECT
adresse.Region,
COUNT(DISTINCT capteur.Code_Capteur) AS Nbr_Capteur,
COUNT(DISTINCT agent.Code_Agent) AS Nbr_Agent
FROM agent
JOIN capteur ON agent.Code_Agent = capteur.Code_Agent
JOIN adresse ON capteur.Code_Adresse = adresse.Code_Adresse
GROUP BY adresse.Region
HAVING Nbr_Capteur > Nbr_Agent

```

7. Peuplement de la base de données

Pour notre base de données test, nous avons choisi de générer des données cohérentes au hasard à l'aide d'une intelligence artificielle. Nous avons fait certaines demande à la main et certaines où nous avons demandé à l'IA de nous générer différentes données en fonction de ce que nous recherchons.

D'abord, nous avons généré ces données sous forme d'Excel.

Par exemple, pour les adresses de nos agences, nous avons récupéré les villes importantes et les données associées à la main.

Voici le résultat :

N° Rue	Ville	Région	Pays
104 Rue du President Edouard Herriot	69002 Lyon	Auvergne-Rhône-Alpes	France
27 Boulevard Marechal de Lattre de Tassigny	21000 Dijon	Bourgogne-Franche-Comté	France
84 Centre Commercial Colombia	18603 Rennes	Bretagne	France
12 Rue Notre Dame de Recouvrance	45000 Orléans	Centre-Val de Loire	France
48 Rue Lorenzo Véro	20000 Ajaccio	Corse	France
3 Rue d'Erstein	67100 Strasbourg	Grand Est	France
18 Rue Monge	59000 Lille	Hauts-de-France	France
3 Rue du 19 Mars 1962	94800 Villejuif	Île-de-France	France
79 Rue d'Ernemont	76000 Rouen	Normandie	France
20 Rue Dauzats	33000 Bordeaux	Nouvelle-Aquitaine	France
6 Rue Clémence Isaure	31000 Toulouse	Occitanie	France
8 Rue du Refuge	44000 Nantes	Pays de la Loire	France
52 Rue Jean Mermoz	13008 Marseille	Provence-Alpes-Côte d'Azur	France
8 Rue Jean de la Varende	14000 Caen	Normandie	France
12 Place du Coderc	24000 Périgueux	Nouvelle-Aquitaine	France
8 Rue François Chénieux	87000 Limoges	Nouvelle-Aquitaine	France

Une fois que nous avons ces données Excel, on peut, ensuite, soit demander à l'IA de nous générer la requête SQL INSERT INTO correspondant aux données Excel lorsqu'il y a trop de données, soit écrire cette requête manuellement lorsque c'est possible ou bien **insérer les valeurs dans MySQL à travers un fichiers CSV**

- Méthode 1 :

Réutilisons les adresses comme exemple, nous avons effectué la requête SQL des adresses manuellement :

```
INSERT INTO adresses (numero_rue, rue, ville, code_postal, region, pays)

VALUES

(104, "Rue du President Edouard Herriot", "Lyon", "69002", "Auvergne-Rhône-Alpes", "France"),
(27, "Boulevard Marechal de Lattre de Tassigny", "Dijon", "21000", "Bourgogne-Franche-Comté", "France"),
(84, "Centre Commercial Colombia", "Rennes", "18603", "Bretagne", "France"),
(12, "Rue Notre Dame de Recouvrance", "Orléans", "45000", "Centre-Val de Loire", "France"),
(48, "Rue Lorenzo Véro", "Ajaccio", "20000", "Corse", "France"),
(3, "Rue d'Erstein", "Strasbourg", "67100", "Grand Est", "France"),
(18, "Rue Monge", "Lille", "59000", "Hauts-de-France", "France"),
(3, "Rue du 19 Mars 1962", "Villejuif", "94800", "Île-de-France", "France"),
(79, "Rue d'Ernemont", "Rouen", "76000", "Normandie", "France"),
(20, "Rue Dauzats", "Bordeaux", "33000", "Nouvelle-Aquitaine", "France"),
(6, "Rue Clémence Isaure", "Toulouse", "31000", "Occitanie", "France"),
(8, "Rue du Refuge", "Nantes", "44000", "Pays de la Loire", "France"),
(52, "Rue Jean Mermoz", "Marseille", "13008", "Provence-Alpes-Côte d'Azur", "France"),
(8, "Rue Jean de la Varende", "Caen", "14000", "Normandie", "France"),
(12, "Place du Coderc", "Périgueux", "24000", "Nouvelle-Aquitaine", "France"),
(8, "Rue François Chénieux", "Limoges", "87000", "Nouvelle-Aquitaine", "France");
```

On effectue cette démarche pour chaque table de notre base de données afin de pouvoir peupler entièrement celle-ci.

- Méthode 2 :

Configure Import Settings

Detected file format: csv 

Encoding:

Columns:

<input checked="" type="checkbox"/> Source Column	Field Type
<input checked="" type="checkbox"/> Valeur_Mesure	<input type="text" value="double"/>
<input checked="" type="checkbox"/> Code_Capteur	<input type="text" value="int"/>
<input checked="" type="checkbox"/> Code_Date	<input type="text" value="int"/>
<input checked="" type="checkbox"/> Code_Gaz	<input type="text" value="int"/>

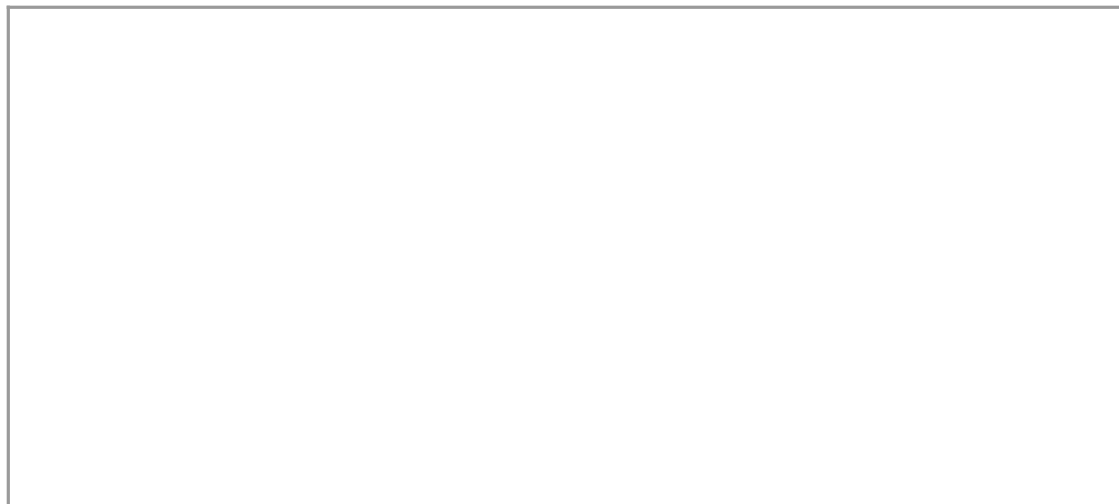
Valeur_Me...	Code_Cap...	Code_Date	Code_Gaz
367.41	9	26311	2
101.64	26	26042	6
69.58	29	26183	6
298.84	20	25491	10
364.31	29	27121	2

On importe des données que nous avons fait sur un Excel à part et importe tout cela en csv !

Nous avons essentiellement utilisé la Méthode 2, les liens entre les tables doivent donc être fait à la main...

Base de données : Peuplement Test

Voici donc notre base de données avec toutes les données remplies de manière logique, pratique, pour tester toutes vos fonctionnalités :



8. Fonctionnalité supplémentaires

Voici un récapitulatif de ce que que vous nous aviez demandé :

On doit pouvoir mettre à jour ces données facilement : ajout/suppression d'une agence ou d'un agent, modification des informations d'une agence ou d'un agent...

→ Cette requête SQL permet en effet de mettre à jour n'importe quelle donnée de la table :


```
UPDATE table1
SET
  colonne1 = 'valeur1',
  colonne2 = 'valeur2',
  ...
(WHERE condition)
```

Pour les données elles-mêmes (des capteurs), il doit être possible d'en ajouter et d'en supprimer (par exemple s'il y a eu un souci sur un capteur) mais pas d'en modifier.

Les colonnes dans la table Mesure doivent être donc inchangeable mais peuvent être supprimées et ajoutées (pas de possibilité d'utiliser des UPDATE en requête pour cette table)

→ Cette requête permet à un certain utilisateur de pouvoir modifier cela, si l'on souhaite que tous les utilisateurs ne puissent pas UPDATE une certaine table 2 choix s'offrent à nous :

- On met les permissions pour chaque utilisateur
- On met les permissions pour un rôle et on assigne ce rôle à tous les utilisateurs souhaités

Nous utiliserons la méthode 1 car il n'y aura que 2 profils utilisateurs (voir plus loin la raison)

```
GRANT INSERT, DELETE ON nom_base.nom_table TO 'user1'@'localhost';

REVOKE UPDATE ON nom_base.nom_table FROM 'user1'@'localhost';
```

Il faut bien éviter qu'un rapport soit modifié ou supprimé une fois qu'il a été produit !

→ Même idée que la demande précédente :

```
GRANT INSERT ON nom_base.nom_table TO 'user1'@'localhost';

REVOKE UPDATE, DELETE ON nom_base.nom_table FROM 'user1'@'localhost';
```

Comptes utilisateurs : Création de (au moins) 2 comptes utilisateurs pour les accès et les droits sur la base de données : un compte 'admin' et un compte 'user'. Il faudra bien définir les différents droits octroyés à chacun de ces comptes.

Pour ce faire, nous devons définir ce que peut faire chacun de ces rôles :

- **admin** --> Toutes les permissions
- **user** --> Permissions de visiter (pas de modification, pas de suppression et pas d'ajout seulement de quoi faire des requêtes de visualisation)

— ADMIN —

Role	Description	Global Privileges
<input checked="" type="checkbox"/> DBA	grants the rights to perform all tasks	<input checked="" type="checkbox"/> ALTER
<input checked="" type="checkbox"/> MaintenanceAdmin	grants rights needed to maintain server	<input checked="" type="checkbox"/> ALTER ROUTINE
<input checked="" type="checkbox"/> ProcessAdmin	rights needed to assess, monitor, and kill any user proce...	<input checked="" type="checkbox"/> CREATE
<input checked="" type="checkbox"/> UserAdmin	grants rights to create users logins and reset passwords	<input checked="" type="checkbox"/> CREATE ROUTINE
<input checked="" type="checkbox"/> SecurityAdmin	rights to manage logins and grant and revoke server an...	<input checked="" type="checkbox"/> CREATE TABLESPACE
<input checked="" type="checkbox"/> MonitorAdmin	minimum set of rights needed to monitor server	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES
<input checked="" type="checkbox"/> DBManager	grants full rights on all databases	<input checked="" type="checkbox"/> CREATE USER
<input checked="" type="checkbox"/> DBDesigner	rights to create and reverse engineer any database sche...	<input checked="" type="checkbox"/> CREATE VIEW
<input checked="" type="checkbox"/> ReplicationAdmin	rights needed to setup and manage replication	<input checked="" type="checkbox"/> DELETE
<input checked="" type="checkbox"/> BackupAdmin	minimal rights needed to backup any database	<input checked="" type="checkbox"/> DROP
<input checked="" type="checkbox"/> Custom	custom role	<input checked="" type="checkbox"/> EVENT
		<input checked="" type="checkbox"/> EXECUTE
		<input checked="" type="checkbox"/> FILE
		<input checked="" type="checkbox"/> GRANT OPTION
		<input checked="" type="checkbox"/> INDEX
		<input checked="" type="checkbox"/> INSERT
		<input checked="" type="checkbox"/> LOCK TABLES
		<input checked="" type="checkbox"/> PROCESS

— USER —

Role	Description	Global Privileges
<input type="checkbox"/> DBA	grants the rights to perform all tasks	<input type="checkbox"/> EXECUTE
<input type="checkbox"/> MaintenanceAdmin	grants rights needed to maintain server	<input type="checkbox"/> FILE
<input type="checkbox"/> ProcessAdmin	rights needed to assess, monitor, and kill any user proce...	<input type="checkbox"/> GRANT OPTION
<input type="checkbox"/> UserAdmin	grants rights to create users logins and reset passwords	<input type="checkbox"/> INDEX
<input type="checkbox"/> SecurityAdmin	rights to manage logins and grant and revoke server an...	<input type="checkbox"/> INSERT
<input type="checkbox"/> MonitorAdmin	minimum set of rights needed to monitor server	<input type="checkbox"/> LOCK TABLES
<input type="checkbox"/> DBManager	grants full rights on all databases	<input type="checkbox"/> PROCESS
<input type="checkbox"/> DBDesigner	rights to create and reverse engineer any database sche...	<input type="checkbox"/> REFERENCES
<input type="checkbox"/> ReplicationAdmin	rights needed to setup and manage replication	<input type="checkbox"/> RELOAD
<input type="checkbox"/> BackupAdmin	minimal rights needed to backup any database	<input type="checkbox"/> REPLICATION CLIENT
<input checked="" type="checkbox"/> Custom	custom role	<input type="checkbox"/> REPLICATION SLAVE
		<input checked="" type="checkbox"/> SELECT
		<input checked="" type="checkbox"/> SHOW DATABASES
		<input checked="" type="checkbox"/> SHOW VIEW
		<input type="checkbox"/> SHUTDOWN
		<input type="checkbox"/> SUPER
		<input type="checkbox"/> TRIGGER
		<input type="checkbox"/> UPDATE

9. Conclusion

En résumé, nous vous proposons cette base de données (**fourni dans le zip fourni, à lancer avec votre SGBD**), avec toutes les demandes annexes et l'implémentation dans vos systèmes de gestion de base de donnée.

Cette base de données, fiable et optimisé, vous sera utile pour gérer efficacement vos données et d'améliorer la qualité de l'air en France.

Les requêtes que vous nous avez demandés en SQL sont aussi disponible dans le fichier ZIP, juste à lancer quand la base de données est créée.

Si vous le souhaitez nous proposons un formation sur MySQL, pour vos futurs agents administratif.

Pour toute demande annexe, n'hésitez pas à nous contacter !

L'équipe ClearData