

BDD

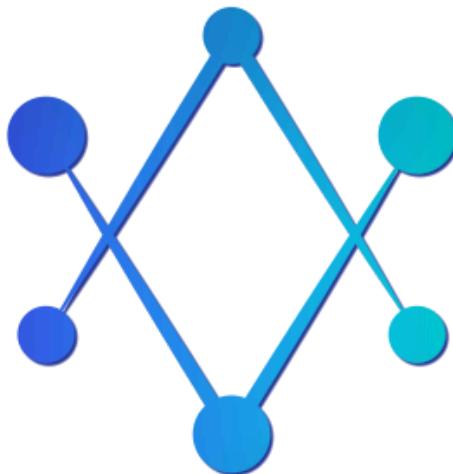
GROUPE 1

LIVRABLE 2



SOMMAIRE

1/Présentation du groupe :	3
2/Reformulation du besoin :	3
3/Requêtes demandées converti en SQL:	4
4/Génération BDD:	10
5/Peuplement de la BDD:	10
6/Fonctionnalités présentes:	11
7/Conclusion:	13



ClearData



1/Présentation du groupe :

Membres du groupe:

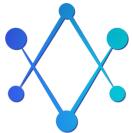
///
///
///
///

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 :

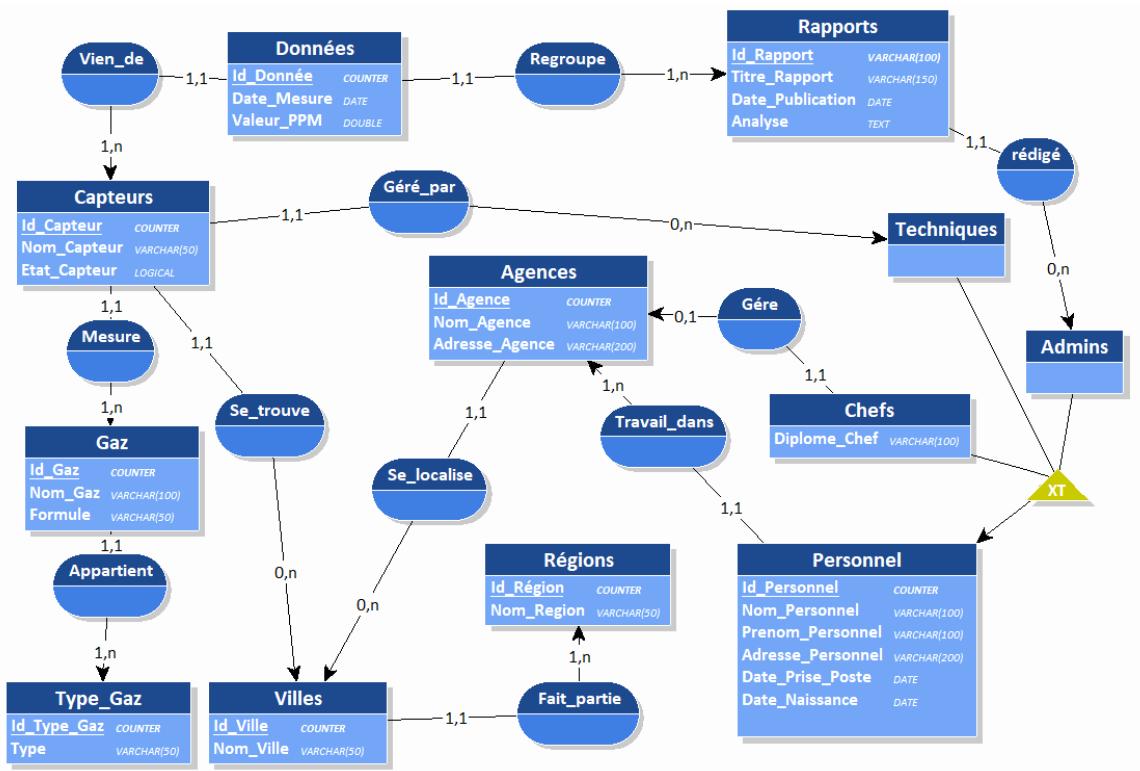
Le ministère de la Transition Écologique souhaite centraliser les données sur la qualité de l'air actuellement dispersées dans les agences régionales. L'entreprise ClearData est missionnée pour concevoir une base de données relationnelle prototype permettant de :

- 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.

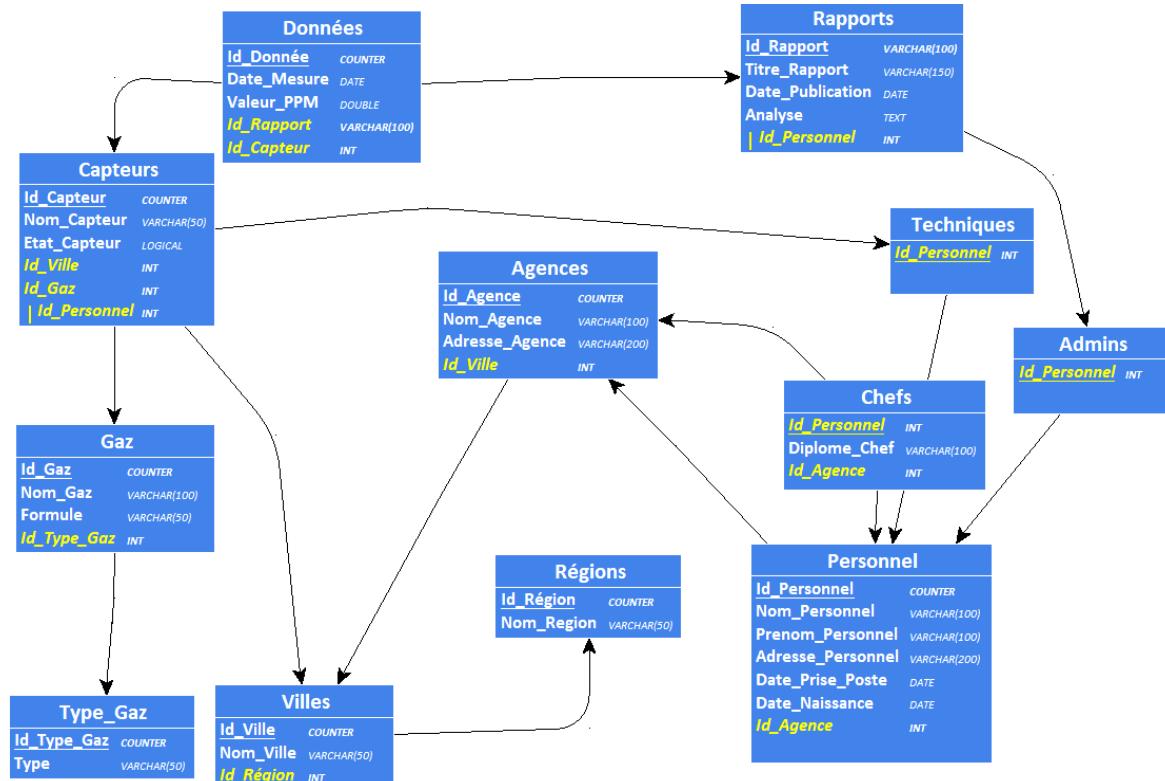


3/Requêtes demandées converties en SQL:

RAPPEL DU MCD



RAPPEL DU MLD





Liste des requêtes demandées:

1. Liste de l'ensemble des agences

SQL: **SELECT Nom_Agence, Adresse_Agence FROM Agences;**

```
mysql> SELECT Nom_Agence, Adresse_Agence FROM Agences;
+-----+-----+
| Nom_Agence | Adresse_Agence |
+-----+-----+
| Agence Nord | 12 Rue de la Paix, Paris
| Agence Sud | 88 Avenue Lumière, Marseille
| Agence Ouest | 5 Boulevard Atlantique, Bordeaux
| Agence Bretagne | 15 Rue de la Mer, Rennes
| Agence Occitanie | 22 Avenue du Soleil, Toulouse
| Agence Rhône-Alpes | 8 Montée de la Colline, Lyon
| Agence Limousin | 5 Rue du Centre, Limoges
+-----+-----+
7 rows in set (0.00 sec)
```

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

SQL: **SELECT Personnel.* FROM Techniques NATURAL JOIN Personnel
NATURAL JOIN Agences NATURAL JOIN Villes WHERE Villes.Nom_Ville =
'Bordeaux';**

```
mysql> SELECT Personnel.* FROM Techniques NATURAL JOIN Personnel NATURAL JOIN Agences NATURAL JOIN Villes WHERE Villes.Nom_Ville = 'Bordeaux';
+-----+-----+-----+-----+-----+-----+
| Id_Personnel | Nom_Personnel | Prenom_Personnel | Adresse_Personnel | Date_Prise_Poste | Date_Naissance | Id_Agence |
+-----+-----+-----+-----+-----+-----+
|      30 | Fabre       | Élise          | 6 Rue de la Gare   | 2023-02-02        | 1992-05-20     |      3 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

3. Nombre total de capteurs déployés.

SQL: **SELECT COUNT(*) FROM Capteurs;**

```
mysql> SELECT COUNT(*) FROM Capteurs;
+-----+
| COUNT(*) |
+-----+
|      7 |
+-----+
1 row in set (0.00 sec)
```



4. Liste des rapports publiés entre 2018 et 2022

SQL: `SELECT * FROM Rapports WHERE YEAR(Date_Publication) BETWEEN 2018 AND 2022;`

Id_Rapport	Titre_Rapport	Date_Publication	Analyse	Id_Personnel
RPT001	Rapport 1	2019-04-01	Niveaux de gaz normaux	1
RPT002	Rapport 2	2019-04-02	Pic de pollution détecté	4
RPT003	Rapport 3	2019-04-03	Intervention nécessaire	7
RPT004	Rapport 4	2019-04-04	Équipement à calibrer	10

4 rows in set (0.00 sec)

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

SQL: `SELECT Date_Mesure, Valeur_PPM, Nom_Région FROM Données NATURAL JOIN Capteurs NATURAL JOIN Villes NATURAL JOIN Régions WHERE Nom_Région IN ('Île-de-France', 'Bretagne', 'Occitanie') AND MONTH(Date_Mesure) BETWEEN 5 AND 6 AND YEAR(Date_Mesure) = 2022;`

Date_Mesure	Valeur_PPM	Nom_Région
2022-05-01	44.2	Île-de-France

1 row in set (0.00 sec)

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

SQL: `SELECT Nom_Personnel FROM Type_Gaz NATURAL JOIN Gaz NATURAL JOIN Capteurs NATURAL JOIN Techniques NATURAL JOIN Personnel WHERE Type = 'GESI';`

Nom_Personnel
Faure
Fernandez

2 rows in set (0.00 sec)



7. Titres et dates des rapports concernant des concentrations de CH4 (sans classement)

SQL: `SELECT Titre_Rapport, Date_Publication FROM Rapports INNER JOIN Données ON Rapports.Id_Rapport = Données.Id_Rapport INNER JOIN Capteurs ON Capteurs.Id_Capteur = Données.Id_Capteur NATURAL JOIN Gaz WHERE Gaz.Formule = 'CH4';`

Titre_Rapport	Date_Publication
Rapport 2	2019-04-02
Rapport 3	2019-04-03
Rapport 4	2019-04-04
Rapport 5	2024-04-05
Rapport 6	2024-04-06
Rapport 7	2024-04-07
Rapport 8	2024-04-07
Rapport 9	2024-04-07
Rapport 10	2024-04-07
Rapport 4	2019-04-04
Rapport 1	2019-04-01
Rapport 4	2019-04-04
Rapport 1	2019-04-01
Rapport 2	2019-04-02
Rapport 3	2019-04-03
Rapport 5	2024-04-05
Rapport 4	2019-04-04
Rapport 5	2024-04-05
Rapport 6	2024-04-06
Rapport 7	2024-04-07
Rapport 8	2024-04-07
Rapport 9	2024-04-07
Rapport 10	2024-04-07
Rapport 1	2019-04-01
Rapport 2	2019-04-02
Rapport 3	2019-04-03
Rapport 4	2019-04-04
Rapport 5	2024-04-05
Rapport 6	2024-04-06
Rapport 7	2024-04-07
Rapport 8	2024-04-07
Rapport 9	2024-04-07
Rapport 10	2024-04-07

8. Afficher le mois où la concentration de HFC a été la moins importante pour chaque région.

SQL: `SELECT r.Nom_Région, MIN(d.Valeur_PPM) AS Valeur_Min FROM Données d JOIN Capteurs c ON d.Id_Capteur = c.Id_Capteur JOIN Villes v ON c.Id_Ville = v.Id_Ville JOIN Régions r ON v.Id_Région = r.Id_Région JOIN Gaz g ON g.Id_Gaz = c.Id_Gaz WHERE g.Formule = 'HFC' GROUP BY r.Nom_Région;`

Nom_Région	Valeur_Min
Hauts-de-France	36.8



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

SQL: `SELECT AVG(Valeur_PPM) FROM Données NATURAL JOIN Capteurs NATURAL JOIN Villes NATURAL JOIN Régions NATURAL JOIN Gaz WHERE Nom_Région = 'Ile-de-France' AND YEAR(Date_Mesure) = 2020 GROUP BY Nom_Gaz;`

```
mysql> SELECT AVG(Valeur_PPM) FROM Données NATURAL JOIN Capteurs NATURAL JOIN Villes NATURAL JOIN Régions NATURAL JOIN Gaz WHERE Nom_Région = 'Ile-de-France' AND YEAR(Date_Mesure) = 2020 GROUP BY Nom_Gaz;
+-----+
| AVG(Valeur_PPM) |
+-----+
| 29.23333333333333 |
+-----+
1 row in set (0.00 sec)
```

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

SQL: `SELECT Personnel.Nom_Personnel, Personnel.Prenom_Personnel, COUNT(Id_Rapport) * 1.0 / TIMESTAMPDIFF(MONTH, Date_Prise_Poste, CURDATE()) AS Productivity FROM Admins NATURAL JOIN Rapports NATURAL JOIN Personnel NATURAL JOIN Agences NATURAL JOIN Villes WHERE Nom_Ville = 'Toulouse' GROUP BY Id_Personnel;`

```
mysql>
mysql> SELECT Personnel.Nom_Personnel, Personnel.Prenom_Personnel, COUNT(Id_Rapport) * 1.0 / TIMESTAMPDIFF(MONTH, Date_Prise_Poste, CURDATE()) AS Productivity FROM Admins NATURAL JOIN Rapports NATURAL JOIN Personnel NATURAL JOIN Agences NATURAL JOIN Villes WHERE Nom_Ville = 'Toulouse' GROUP BY Id_Personnel;
+-----+-----+-----+
| Nom_Personnel | Prenom_Personnel | Productivity |
+-----+-----+-----+
| Martin        | Frederic       |      0.02778 |
| Cartman       | Michel          |      0.03846 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

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

SQL: DELIMITER //

```
CREATE PROCEDURE GetRapportsByGaz(IN gazName VARCHAR(50))
```

```
BEGIN
```

```
    SELECT r.*
```

```
    FROM Rapports r
```

```
    INNER JOIN Données d ON r.Id_Rapport = d.Id_Rapport
```

```
    INNER JOIN Capteurs c ON c.Id_Capteur = d.Id_Capteur
```

```
    NATURAL JOIN Gaz
```



```
WHERE Gaz.Formule = gazName;
```

```
END //
```

```
DELIMITER ;
```

```
CALL GetRapportsByGaz('CO2');
```

```
mysql>
mysql> DELIMITER ;
mysql> DELIMITER //
```

```
mysql>
mysql> CREATE PROCEDURE GetRapportsByGaz(IN gazName VARCHAR(50))
-> BEGIN
->   SELECT r.* 
->   FROM Rapports r
->   INNER JOIN Données d ON r.Id_Rapport = d.Id_Rapport
->   INNER JOIN Capteurs c ON c.Id_Capteur = d.Id_Capteur
->   NATURAL JOIN Gaz
->   WHERE Gaz.Nom_Gaz = gazName;
-> END //
Query OK, 0 rows affected (0.01 sec)
```

```
mysql>
mysql> DELIMITER ;
mysql> CALL GetRapportsByGaz('CO2');
```

Id_Rapport Titre_Rapport Date_Publication Analyse Id_Personnel
RPT001 Rapport 1 2019-04-01 Niveaux de gaz normaux 1
RPT002 Rapport 2 2019-04-02 Pic de pollution détecté 4
RPT003 Rapport 3 2019-04-03 Intervention nécessaire 36
RPT004 Rapport 4 2019-04-04 Équipement à calibrer 10
RPT005 Rapport 5 2024-04-05 Données stables 35
RPT006 Rapport 6 2024-04-06 Maintenance recommandée 35
RPT007 Rapport 7 2024-04-07 Aucune anomalie détectée 19
RPT008 Rapport 8 2024-04-07 anomalie détectée 10
RPT009 Rapport 9 2024-04-07 Augmentation des valeurs 16
RPT010 Rapport 10 2024-04-07 Capteur fonctionne 36
RPT002 Rapport 2 2019-04-02 Pic de pollution détecté 4
RPT003 Rapport 3 2019-04-03 Intervention nécessaire 36
RPT004 Rapport 4 2019-04-04 Équipement à calibrer 10
RPT005 Rapport 5 2024-04-05 Données stables 35
RPT006 Rapport 6 2024-04-06 Maintenance recommandée 35
RPT007 Rapport 7 2024-04-07 Aucune anomalie détectée 19
RPT008 Rapport 8 2024-04-07 anomalie détectée 10
RPT009 Rapport 9 2024-04-07 Augmentation des valeurs 16
RPT010 Rapport 10 2024-04-07 Capteur fonctionne 36
RPT001 Rapport 1 2019-04-01 Niveaux de gaz normaux 1
RPT002 Rapport 2 2019-04-02 Pic de pollution détecté 4
RPT003 Rapport 3 2019-04-03 Intervention nécessaire 36
RPT004 Rapport 4 2019-04-04 Équipement à calibrer 10
RPT005 Rapport 5 2024-04-05 Données stables 35
RPT006 Rapport 6 2024-04-06 Maintenance recommandée 35
RPT007 Rapport 7 2024-04-07 Aucune anomalie détectée 19
RPT008 Rapport 8 2024-04-07 anomalie détectée 10
RPT009 Rapport 9 2024-04-07 Augmentation des valeurs 16

```
28 rows in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.26 sec)
```



IMPORTANT: DELIMITER SERT À DÉCLARER OU LA PROCÉDURE COMMENCE / SE TERMINE.

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

SQL: `SELECT R.* FROM Régions R JOIN Villes V ON R.Id_Région = V.Id_Région LEFT JOIN Capteurs C ON V.Id_Ville = C.Id_Ville LEFT JOIN Agences A ON V.Id_Ville = A.Id_Ville LEFT JOIN Personnel P ON A.Id_Agence = P.Id_Agence GROUP BY R.Id_Région HAVING COUNT(DISTINCT C.Id_Capteur) > COUNT(DISTINCT P.Id_Personnel);`

```
mysql> SELECT R.* FROM Régions R JOIN Villes V ON R.Id_Région = V.Id_Région LEFT JOIN Capteurs C ON V.Id_Ville = C.Id_Ville LEFT JOIN Agences A ON V.Id_Ville = A.Id_Ville LEFT JOIN Personnel P ON A.Id_Agence = P.Id_Agence GROUP BY R.Id_Région HAVING COUNT(DISTINCT C.Id_Capteur) > COUNT(DISTINCT P.Id_Personnel);
+-----+-----+
| Id_Région | Nom_Région |
+-----+-----+
| 7 | Hauts-de-France |
| 8 | Normandie |
+-----+-----+
2 rows in set (0.00 sec)
```

4/Génération BDD:

La base de donnée est générée grâce à un script SQL qui d'abord la supprime si elle existe déjà puis la recréer en mettant chaque tables, attributs et associations(Clés primaires, Clés étrangères) vis à vis du MLD

SQL:

Consulter le scripte SQL fournis avec le document:

(BDD_PROJET_STRUCTURE.sql)

Difficultées rencontrées:

Réussir à concevoir un MCD/MLD correcte qui répond au demandes du projet

5/Peuplement de la BDD:

La base de données est peuplé grâce à un scripte SQL qui injecte des données en respectant toute les association clé étrangères et le spécifications demandées

spécifications demandées:

au moins 200 relevés de capteurs, au moins 20 employés, au moins 10 rapports

Les données doivent être réalistes quant aux régions et aux villes.

Pour les dates, se baser sur un intervalle entre janvier 2017 et décembre 2024.



Les quantités mesurées sont en ppm (partie par million) peuvent prendre des valeurs entre 0.01 et 500.

SQL:

Consulter le scripte SQL fournis avec le document:

(BDD_PROJET_DATA_INSERT.sql)

Difficultés rencontrées:

Réussir à faire des associations correctes entre les tables avec les clés étrangères

6/Fonctionnalités présentes:

- La base de données respecte au maximum les principes de qualité.
- Elle contient des données non redondantes et cohérentes, consultables, ajoutables et supprimables facilement (dans les limites des demandes).
- Elle présente un fort degré d'indépendance entre les données et le matériel (modèle/SGBD utilisé).
- Elle présente un fort degré d'indépendance entre les données et les programmes, notamment en matière de sécurité (création et accès).
- Les contraintes d'intégrité sont définies sur l'ensemble des colonnes et des tables.
- La base est peuplée avec des données complètes et pertinentes.
- Elle contient au moins 200 relevés de capteurs.
- Elle contient au moins 20 employés.
- Elle contient au moins 10 rapports.
- Les données qu'elle contient sont réalistes quant aux régions et aux villes.
- Les dates présentes sont comprises entre janvier 2017 et décembre 2024.
- Les quantités mesurées sont exprimées en ppm (parties par million).
- Les valeurs enregistrées varient entre 0,01 et 500 ppm.

Exploitation de la base :



La base contient au moins deux comptes utilisateurs : un compte 'admin' et un compte 'user':

Le compte admin peut lire/ecrire dans chaque table de la BDD

SQL:

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin1234';
GRANT ALL PRIVILEGES ON cleargaz.* TO 'admin'@'localhost';
FLUSH PRIVILEGES;
```

Le compte user peut lire uniquement dans les tables
(Capteurs,Rapports,Données,Villes,Données,Régions) sans pouvoir les modifier

SQL:

-- User

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'user1234';
GRANT SELECT ON cleargaz.Capteurs TO 'user'@'localhost';
GRANT SELECT ON cleargaz.Rapports TO 'user'@'localhost';
GRANT SELECT ON cleargaz.Données TO 'user'@'localhost';
GRANT SELECT ON cleargaz.Villes TO 'user'@'localhost';
GRANT SELECT ON cleargaz.Données TO 'user'@'localhost';
GRANT SELECT ON cleargaz.Régions TO 'user'@'localhost';
FLUSH PRIVILEGES;
```

Le moteur de stockage est choisi en fonction des requêtes-types du projet pour optimiser l'écriture, la lecture et la suppression de données:

InnoDB est le moteur de stockage par défaut MySQL et on l'utilise pour notre BDD car il convient le mieux.

La base prévoit des commandes SQL pour ajouter, supprimer ou modifier les données lorsque cela est possible:



En utilisant des **PROCÉDURES MySQL** il possible de faciliter l'insertion/suppression/modification de données:

Consulter le script SQL fournis avec le document:

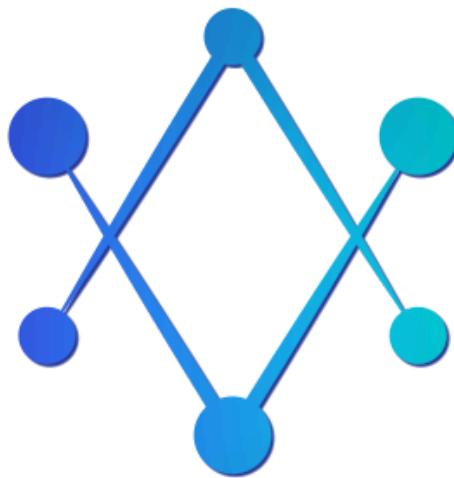
(BDD_PROJET_Smart_DATA_Manip.sql)

Ce script SQL permet de faciliter la manipulation des données car il associe les clés étrangères automatiquement il suffit de donner le nom de l'item de la table voisine

Création d'un script SQL dump de la BDD: BDD_GROUPE1_DUMP.sql

7/Conclusion:

Ce livrable permet de voir les différentes manières d'optimisation pour faciliter la manipulation des données une fois la BDD remplie, et l'injection de données permettra aussi d'assurer la démonstration des requêtes SQL lors de la présentation finale



ClearData

FIN DU DOCUMENT