Université de Sherbrooke

IGE487 – Modélisation de données

Enseignant : Luc Lavoie

Herbivorie, phase 2 : consolidation et alimentation de données

Jalon 2

DOCUMENTATION

Ahmed Khalil Lahmar

Seynabou Lena Kane

Saraha Harimamy Rakotoarinivo

Cheikh Ahmadou Tidiane Ly

Rafael Dhaene

14 novembre 2025

1. Introduction

Ce document présente la documentation du modèle de données « Herbivorie ». Ce modèle a pour objectif de gérer et d'analyser les données d'étude de la migration d'espèces végétales sous l'effet des changements climatiques et à son impact sur les écosystèmes fait par un groupe de chercheurs. L'étude est planifiée en prenant pour cas d'études le trille sur les flancs du mont Mégantic. Elle vise à structurer et organiser les premières données de terrain du mois de juin jusqu'en octobre recueillies au prototype développé lors de la phase 1.

2. Objectif de la base de données

La base Herbivore permet de centraliser les observations environnementales (température, humidité, pression, etc.) et biologiques (état des plantes, floraison, densité, etc.) recueillies sur différentes parcelles (placettes). Elle sert à suivre l'évolution des écosystèmes étudiés.

3. Les entités

- Plant : Représente une plante observée. Contient l'identifiant de la plante et la date d'observation.

  Attribut : id, placette, parcelle, date

- Placette : Zone ou parcelle où sont effectuées les observations.

  Attribut : placette, peuplement, date

- Etat : Définit l'état observé d'une plante.

  Attribut : etat, description

- Peuplement : Élément biologique observé

  Attribut : peuplement, description

- Taux : Valeur ou pourcentage associé à une mesure environnementale ou biologique.

  Attribut : taux_id , taux minimum, taux maximum

- Arbre : Représente un arbre observé dans une placette, avec son rang et sa description.

    Attribut : arbre, description

- Obsetat : Observation de l'état d'une plante à une date donnée.

    Attribut: plant_id, date, etat

- ObsDimension : Observation de la dimension d'une plante.

    Attribut : plant_id, date, longueur, largeur

- Obsfloraison : Observation de la floraison.

    Attribut: plant_id, date, fleur

- Obstruction : Observation d'un obstacle ou d'un facteur limitant.

    Attribut : placette_id, type observation, taux

A-Jalon 1

1. Les inadéquations

- Y1 : Ajout des types et des tables requis pour consigner les conditions météorologiques quotidiennes. La température minimale et maximale, le taux d'humidité minimale et maximale, le nombre de millimètres et la nature des précipitations.

- Y2 : Ajout des données météorologiques pour le mois de juin 2016 jusqu'en 2025. Les données brutes seront contenues dans la table *CarnetMeteo*. Afin de gérer les données ammulables, l'utilisation de la projection jointure est recommandé. Afin d'éviter d'insérer des données erronées ou non valides, des fonctions ont été ajouté pour effectuer des vérifications. Les vérifications vont se faire sur la présence d'une date valide, la cohérence des températures minimales et maximales, des données de l'humidité, la vitesse du vent, la pression minimale et maximale et la précipitation. Ajout d'un trigger qui insère automatiquement les données valides dans notre base de données. Si les valeurs ne sont pas valides, ils seront juste insérés dans le carnetMeteo.

- Y3 : la vue *VueConditionsMeteoSansPrecipitation* permet de donner les conditions météorologiques complètes hors précipitations.

Une fenêtre virtuelle qui affichera la date, la température minimale et maximale, le niveau d'humidité minimale et maximale, la vitesse du vent minimale et maximale, la pression minimales et maximale et la note.

- Y4 : la procédure *retirer_donnees_meteo* permet de retirer les données météorologiques pour une période qui sont en dessus d'une température donnée

Les entrées sont la date de début de la période, la date de fin de la période et la température minimale souhaité. En retour, la suppression des données de précipitations, de notes, de pression, de vents, de l'humidité, de température et une notification de réussite de mise à jour avec la période et le pourcentage utilisé.

- Y5 : la procédure *augmenter_temperatures* permet d'augmenter les températures minimales et maximales rapportés d'un pourcentage donné durant une période.

Les entrées sont la date de début de la période, date de fin de la période et le pourcentage à ajouter. En retour les températures mis à jour et une notification de réussite de mise à jour avec la période et le pourcentage utilisé.

2. IMM

Conception d'un IMM qui propose les fonctions minimales EMIR.

FUNCTION *consulter_meteo_par_date* : il permet de connaître les conditions météorologiques quotidiennes

FUNCTION *modifier_observation* : il permet d'effectuer des changements de valeur dans une table

FUNCTION *ajouter_observation* : il permet d'ajouter de nouvelles observations

FUNCTION *rechercher_anomalies* : il permet de vérifier les différentes anomalies au niveau de la température, de l'humidité, de vent et de la pression.

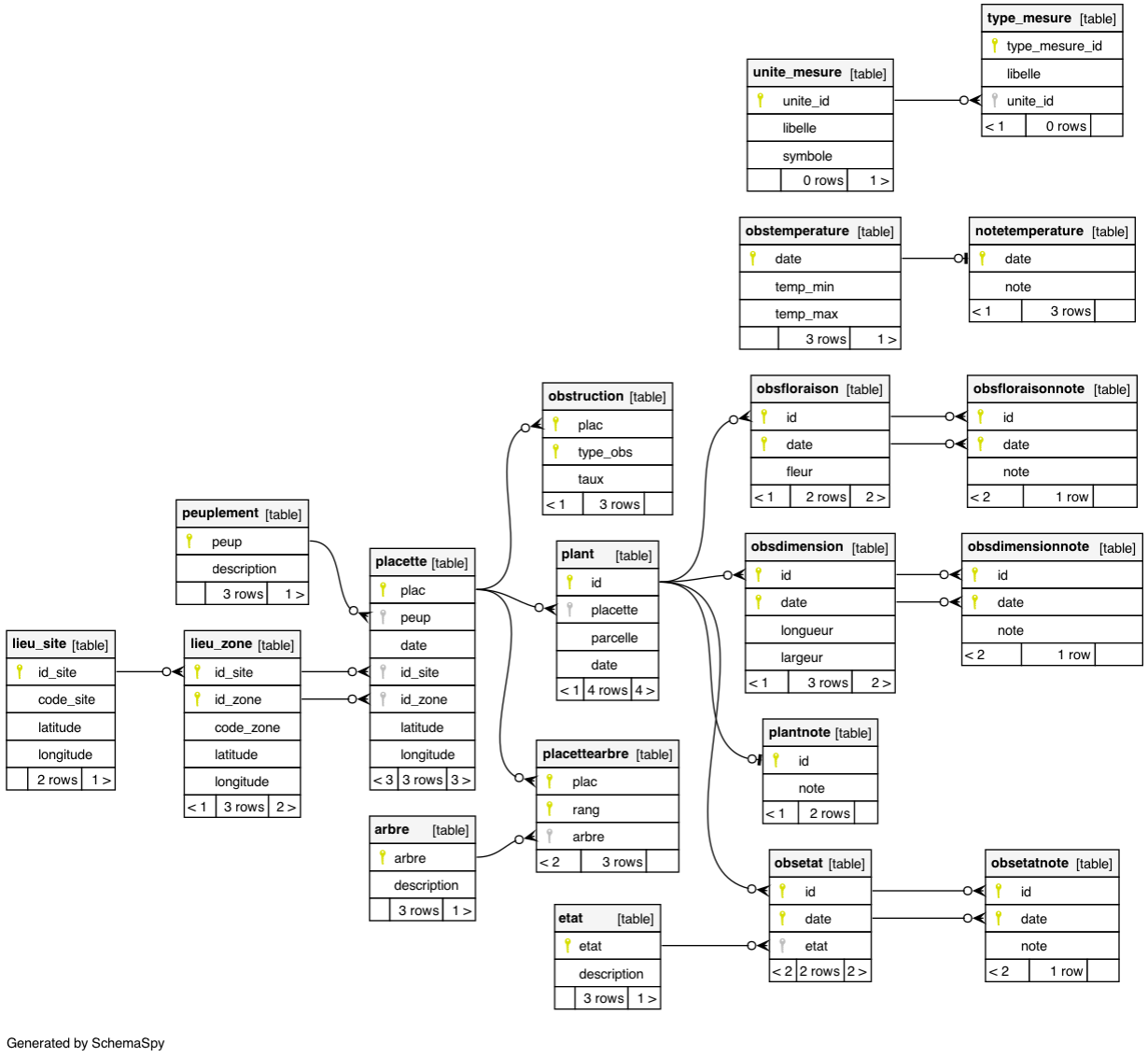
3. Priorisation et choix des inadéquations à corriger :

La priorisation des inadéquations à corriger a suivi une approche progressive, visant d'abord à enrichir le cœur fonctionnel du système avant de développer les fonctionnalités pour les utilisateurs finaux. Dans cette optique, les inadéquations **Y1 et Y2** ont été traitées en premier lieu, car elles sont directement liées à notre problème initial. Ces étapes ont permis d'établir une fondation de données météorologiques robuste, en ajoutant les structures nécessaires (Y1) et en mettant en place un mécanisme fiable pour l'alimentation et la validation des données brutes (Y2). Pour la gestion des informations

manquantes, le choix s'est porté sur la technique de la projection-jointure, considérée comme la solution optimale pour garantir la flexibilité et l'intégrité du modèle.

Une fois cette base de données consolidée, l'attention s'est tournée vers la création de fonctionnalités destinées aux utilisateurs. Y3 a permis de fournir un accès simplifié aux conditions météorologiques via une vue dédiée, tandis que Y4 et Y5 ont introduit des procédures permettant la manipulation ciblée de ces informations. Enfin, l'amorce de l'Interface Machine-Machine (IMM) a été initiée. Bien que son développement complet soit un des objectifs du jalon 2, cette étape était cruciale pour préparer une couche d'accès sécurisée. L'objectif est de permettre aux utilisateurs d'effectuer des opérations de type ÉMIR (Évaluation, Modification, Insertion, Retrait) via les fonctions de l'API, sans leur donner un accès direct aux requêtes natives.

4. Modèle conceptuel



Le modèle de conception structure les données écologiques selon une hiérarchie site–zone–placette et organise les observations dans des tables spécialisées. Les informations optionnelles sont gérées par des tables de projection pour éviter les valeurs Null. L'ensemble forme une base normalisée, claire et adaptée aux analyses.

5. Jeu de test

Test 1 : Reconstruction Complète de l'Historique d'un Plant

Ce test est le plus important. Il montre la possibilité de rassembler toutes les informations atomiques (dimensions, floraison, état, notes) pour un plant spécifique sur plusieurs dates, sans incohérence. Exemple : le plant MMA0001, qui a des observations à trois dates différentes.

La requête utilise des LEFT JOIN pour s'assurer que même si une observation (par exemple, ObsFloraison) n'existe pas pour une certaine date, nous voyons quand même les autres données de cette date. Le jeu du test se trouve dans le fichier test1.sql.

Ce que ce test prouve : Le résultat de cette requête rassemble toutes les informations de manière cohérente. Vous verrez que la note sur la dimension du 2023-07-15 est correctement associée à la bonne ligne, tout comme la note sur la floraison du 2023-06-15. Il n'y a pas de mélange d'informations entre les dates. C'est la preuve de la partie **"Join"** de la *Project-Join Normal Form*.

Test 2 : Isoler un Fait Spécifique à travers le Temps

Ce test démontre l'avantage de la décomposition : on peut interroger un seul aspect de l'évolution d'un plant très efficacement. Prenons le plant **MMA0003**, dont l'état se dégrade avec le temps. Le jeu du test se trouve dans le fichier test2.sql.

Ce que ce test prouve : Vous obtenez un historique clair et précis de l'état du plant, passant de 'Vivant' à 'Dommage partiel' puis 'Partiellement défolié'. Pour obtenir cette information, la base de données n'a pas eu besoin de lire ou de filtrer les données non pertinentes sur les dimensions ou la floraison. C'est la preuve de la partie **"Project"** (projection/décomposition) de la 5NF.

Test 3 : Absence d'Anomalies de Jointure (Tuples Parasites)

C'est un test conceptuel. Imaginez que **ObsDimension** et **ObsEtat** étaient dans la même table. Pour le plant **MMA0003**, vous auriez une ligne pour **2023-05-15** avec une dimension et un état. Mais pour la date **2023-06-15**, vous avez une nouvelle dimension ET un nouvel état. Comment lieriez-vous le fait que l'état a changé à cause d'un insecte au fait que la croissance a stagné ?

La structure évite ce problème. La clé primaire (**id, date**) dans chaque table d'observation garantit qu'un fait (comme **longueur = 43**) est lié de manière indissociable à une date précise, et ne peut pas être accidentellement associé à un autre fait d'une autre date.

B- Jalon 2

Dans le cadre de la phase 2 (J2), plusieurs éléments clés du modèle de données ont été révisés afin d'améliorer la cohérence, la qualité de l'information et la robustesse de l'architecture. Les sections suivantes présentent les choix conceptuels retenus ainsi que leurs justifications techniques.

1. Gestion du manque d'information

La gestion des valeurs manquantes constitue un enjeu important dans la modélisation des données. Quatre approches ont été analysées. La première consiste à utiliser explicitement la valeur « Null », ce qui permet de distinguer clairement une information absente ou inconnue, mais introduit une complexité supplémentaire dans la définition des contraintes et l'écriture des requêtes, qui doivent gérer explicitement ces cas. La deuxième approche attribue une valeur par défaut. Bien qu'elle simplifie certains traitements et évite les Null, elle reste problématique : une valeur comme « 0 » peut représenter une donnée réelle et non une absence d'information, ce qui entraîne des ambiguïtés sémantiques. La quatrième approche est une extension de la troisième, combinant projection-jointure et restriction-union, avec une table supplémentaire pour représenter la cause ou la raison de l'absence d'information.

La troisième approche, fondée sur la projection-jointure, a été retenue. Conceptuellement plus propre, elle permet d'éviter les champs Null dans la table principale en déplaçant l'information optionnelle dans des tables dédiées. Bien que cette stratégie nécessite davantage de jointures, elle offre une structuration plus rigoureuse des données et garantit l'accès uniquement à l'information pertinente. Elle réduit également le volume de données manipulées lors des requêtes, contribuant ainsi à une optimisation générale des performances.

2. Unités de mesure

Deux stratégies ont été évaluées pour la gestion des unités de mesure. La première consiste à imposer un système uniforme pour l'ensemble des données. Cette approche est privilégiée, car elle élimine la nécessité de gérer plusieurs conversions au niveau de la base de données. Les transformations éventuelles sont effectuées au niveau des interfaces personne-machine, ce qui simplifie la modélisation interne et limite les sources d'erreur.

La seconde approche consiste à modéliser explicitement les unités au moyen de tables supplémentaires définissant les types de mesures et les conversions associées. Bien qu'elle permette une grande flexibilité, elle introduit une complexité accrue dans les requêtes et dans la gestion des données.

L'utilisation d'un dictionnaire d'unités standardisées constitue donc la solution la plus appropriée. Elle réduit les incohérences, facilite les requêtes et assure une meilleure qualité des données, tout en confinant les opérations de conversion à l'interface utilisateur.

3. Modélisation des lieux

Une révision complète du script bd_create a été effectuée afin d'intégrer les nouvelles exigences relatives à la représentation spatiale. Deux niveaux de localisation ont été définis de manière explicite : le site et la zone. Ceux-ci sont implémentés à travers les tables lieu_site et lieu_zone, ce qui permet d'introduire une hiérarchie géographique structurée.

Les tables écologiques existantes ont été ajustées pour refléter cette organisation. En particulier, la table Placette inclut désormais les clés étrangères id_site et id_zone, assurant ainsi l'intégration cohérente de chaque placette dans la structure site → zone → placette. Lorsque pertinent, les coordonnées GPS (latitude et longitude) ont également été ajoutées afin d'améliorer la précision spatiale et d'appuyer les analyses ultérieures.

Le script remanié consolide les efforts du Jalon 1 : uniformisation des clés primaires, correction des références étrangères, intégration de tables de projection telles que PlantNote ou ObsDimensionNote et structuration modulaire des observations biologiques. L'ensemble constitue une base de données plus robuste, mieux normalisée et adaptée aux prochaines étapes du projet, notamment l'ELT, l'ÉMIR et les analyses statistiques.

4. Taux

Les données qualitatives associées aux taux ont été converties en mesures quantitatives afin d'améliorer la précision et d'obtenir une interprétation plus objective. Dans ce cadre, l'attribut type_obs de la table Obstruction a été remplacé par une valeur numérique comprise entre 0 et 100 (Taux_val). Cette transformation modifie le comportement des scripts d'insertion ainsi que des requêtes SELECT, puisqu'il n'est plus nécessaire de gérer une catégorie qualitative.

La relation vers la table de référence Taux par clé étrangère a été retirée, mais la table est conservée comme élément documentaire destiné à soutenir les analyses futures ou les besoins de traçabilité. Cette conservation n'entrave pas les opérations d'insertion et laisse ouverte la possibilité d'exploiter ces définitions lors de traitements analytiques ultérieurs.

5. IMM

Fonction site :

Cette fonction gère les sites dans la base de données :

1. **ajouter_site** : insère un nouveau site avec son code et ses coordonnées, ou retourne l'ID existant si le site est déjà présent.

2. **modifier_site** : met à jour le code et les coordonnées d'un site existant à partir de son ID.
3. **get_site_by_code** : récupère les informations d'un site à partir de son code.

Elle fournit des opérations spécifiques aux sites, permettant leur ajout, modification et consultation.

Fonction zone :

Cette fonction gère les zones associées à un site :

1. **ajouter_zone** : ajoute une zone avec son code et ses coordonnées à un site donné, ou retourne l'ID existant si la zone est déjà présente.
2. **modifier_zone** : met à jour le code et les coordonnées d'une zone existante à partir de son ID et du site auquel elle appartient.
3. **get_zone_by_code** : récupère les informations d'une zone spécifique à partir du code de la zone et de l'ID du site.

Elle fournit des opérations ciblées sur les zones, permettant leur ajout, modification et consultation dans le contexte d'un site.

Fonction mesures :

Cette fonction gère les unités et types de mesure :

1. **ajouter_unite_mesure** : ajoute une unité (mètre, litre, etc.) si elle n'existe pas déjà.
2. **ajouter_type_mesure** : ajoute un type de mesure (température, pression, volume…) lié à une unité existante, en évitant les doublons.
3. **lister_unites** : retourne la liste complète des unités disponibles.
4. **lister_types_mesure** : retourne la liste des types de mesure avec leurs unités associées.

Elle permet l'insertion et la consultation standardisées des unités et types de mesures pour le système.

Fonction observation :

Cette fonction gère les observations météorologiques :

1. **modifier_observation** : met à jour une valeur spécifique d'une colonne dans une table d'observations pour une date donnée.

2. **ajouter_observation** : insère une nouvelle observation complète (température, humidité, vent, pression et note) pour une date donnée, sans créer de doublons.

Elle permet l'ajout et la modification ciblée des données météorologiques dans le système, assurant cohérence et intégrité des enregistrements.

Fonction meteo :

consulter_meteo_par_date récupère toutes les observations météo pour une plage de dates donnée, en combinant température, humidité, vent, pression et notes, et retourne les résultats sous forme de tableau trié par date.

Elle fournit une vue consolidée et chronologique des données météorologiques pour une période spécifique, facilitant consultation et analyse.

6. Jeu de test

De nouveaux jeu de données ont été intégré afin de tester les nouvelles modifications.

Test Fonction zone :

Ce test valide les fonctions de gestion des sites et des zones dans le schéma *Herbivorie*. Il crée un site de test, ajoute plusieurs zones, vérifie la non-duplication lors de la réinsertion d'une zone existante, puis modifie une zone et confirme la mise à jour. Enfin, il interroge la fonction de récupération par code afin de s'assurer que les informations renvoyées correspondent bien aux données modifiées.

Test Procédures :

Le script insère des données météo fictives (température, humidité, vent, pression, précipitations) pour les quatre derniers jours, puis effectue deux tests :

1. Augmentation des températures de 10 % sur la période.
2. Suppression des enregistrements où la température minimale est inférieure à 8 °C.

C'est un test fonctionnel simple visant à vérifier les procédures de mise à jour et de filtrage des données météo.

Test Fonction site :

Le script teste les différentes opérations sur les sites :

1. Ajout d'un nouveau site et vérification qu'un doublon renvoie le même ID.

2. Consultation du site par son code.
3. Modification des informations d'un site existant et vérification de la mise à jour.
4. Ajout de plusieurs sites supplémentaires et vérification de leur présence dans la table.

C'est un test fonctionnel simple pour valider l'ajout, la lecture, la modification et la gestion des doublons des sites.

Test mesures :

Ce script initialise la base de données avec les unités de mesure et les types de mesures associées.

1. **Insertion des unités** : il ajoute les unités de longueur (m, km, cm, mm), de temps (s, min, h), de masse (kg, g, mg), de température (°C, K), de pression (Pa, hPa, bar), de vitesse (km/h, m/s), de volume (L, mL) et le pourcentage (%).
2. **Insertion des types de mesure** : chaque type est lié à une unité, par exemple température de l'air en °C, humidité relative en %, précipitations en mm ou litres, vitesse du vent en km/h ou m/s, et ainsi de suite pour la pression, le poids, la durée, la distance et le volume.

L'objectif est de structurer les mesures pour garantir cohérence et normalisation dans les saisies et calculs ultérieurs.

10. Conclusion

Ce modèle de données offre une base solide pour la collecte, le stockage et l'analyse des données écologiques. Il respecte les bonnes pratiques de modélisation, avec séparation des entités, normalisation et définition claire des relations entre les objets du domaine Herbivore.