



PROJET DE FIN D'ETUDES SGBDR

Conception d'un système de gestion agricole pour le village d'Androndra

Rédigé par : HERIMIHARISOA Basilio Lala Martial

Filière : Informatique de gestion

Classe : 3e Année IG-42 FI-A

Année universitaire : 2024-2025

Année édition : Juin 2025

I. PRESENTATION DU PROBLEME

Dans notre village d'Androndra, presque tout le monde vit de l'agriculture. C'est ce que nous faisons depuis toujours. Mais à force d'observer autour de moi, je voyais que nous perdions beaucoup de temps et d'énergie. Nous notions tout à la main, parfois sur des papiers que nous perdions. Nous ne savions pas combien nous dépensions, ni ce qui rapportait le plus. Nous avançons un peu à l'aveugle.

Alors je me suis dit : et si nous avions un système pour mieux gérer tout ça ? C'est comme ça que ce projet est né. Pas juste pour faire joli, mais pour nous aider à mieux travailler.

Objectif du projet

Ce projet vise à mettre en place un système de gestion informatisé pour notre exploitation agricole. L'idée est de centraliser toutes les informations importantes pour mieux planifier, suivre et analyser nos activités. Plus précisément, ce système devra nous permettre de :

Suivre les cultures sur chaque parcelle, en fonction du sol et de l'eau.

Gérer les dépenses liées aux semences, aux engrais.

Faciliter la prise de décision pour améliorer notre travail au quotidien.

II. VOCABULAIRE DES TERMES UTILISES

Avant de construire quoi que ce soit, je devais être sûr que tout le monde comprenne les mêmes mots. Parce que quand on dit "intrant" ou "parcelle", chacun imagine peut-être autre chose. J'ai donc fait une petite liste, un "vocabulaire", pour qu'on parle tout le même langage. C'était simple, mais très important.

Parcelle : Petite portion de terrain cultivée, faisant partie d'un grand terrain

Intrant : Produit ajouté à la culture (semence, engrais, pesticide...)

Cycle de culture : Durée nécessaire pour cultiver une plante, du semis à la récolte

Variété : Type spécifique d'une culture (ex. : laitue batavia, Brède ananambo)

Tâche agricole : Action réalisée sur le champ (labour, arrosage, désherbage, etc.)

Saisonnalité : Période de l'année favorable à une culture

III. COLLECTE ET ANALYSE DES DONNEES

Pour faire une bonne base de données, il faut d'abord savoir ce que nous voulons y mettre. Mais au début, j'étais un peu perdu. Il y avait trop d'informations, un peu partout, et pas toujours claires. C'est pour cela que j'ai décidé de suivre ces étapes :

A. Dictionnaire des données

Le dictionnaire de données, c'est comme une liste qui explique toutes les informations que nous allons utiliser dans le projet.

Propriété	Description	Format	Taille	Type	Contrainte
ID_terrain	Identifiant unique du terrain	Entier	8	E	Auto-incrémenté
nom_terrain	Nom ou code du terrain	Alphabétique	50	E	Valeur Obligatoire
terrain_supprimee	Indique si le terrain a été supprimée	Booléen			
ID_localisation	Identifiant unique du localisation	Entier	8	E	Auto-incrémenté
localisation_terrain	Lieu, village ou coordonnées	Alphabétique	100	E	Valeur Obligatoire
ID_parcelle	Identifiant unique de la parcelle	Entier	8	E	Auto-incrémenté
nom_parcelle	Nom de la parcelle	Alphabétique	50	E	Valeur Obligatoire
superficie_parcelle	Superficie de la parcelle	Décimal	10	E	Valeur Obligatoire
Disponible	Disponibilité de la parcelle	Booléen		E	Valeur Obligatoire
parcelle_supprime	Indique si la parcelle a été supprimée	Booléen			
type_sol	Type de sol	Alphabétique	50	E	
ID_culture	Identifiant unique de la culture	Entier	8	E	Auto-incrémenté
nom_culture	Nom de la culture	Alphabétique	50	E	Valeur Obligatoire
variete_culture	Variété spécifique	Alphabétique	50	E	Valeur Obligatoire
cycle_culture	Durée du cycle (JOUR)	Entier	8	E	
culture_supprime	Indique si la culture a été supprimée.	Booléen			
saisonnalite_culture	Saison adaptée	Alphabétique	30	E	
ID_plantation	Identifiant de la plantation	Entier	8	E	Auto-incrémenté
date_plantation_terre	Date de semis	Date	-	E	Valeur Obligatoire
methode_culture	Type de méthode	Alphabétique	50	E	
quantite_plantee	Quantité plantée	Décimal	10	E	Valeur Obligatoire
plantation_supprime	Indique si la plantation a été supprimée.	Booléen			Valeur Obligatoire

Propriété	Description	Format	Taille	Type	Contrainte
ID_intrant	Identifiant de l'intrant	Entier	8	E	Auto-incrémenté
ID_type_intrant	Identifiant du type d'intrant	Entier	8	E	Auto-incrémenté
type_intrant	Type (semence, engrais)	Alphabétique	30	E	Valeur Obligatoire
nom_intrant	Libellé précis	Alphabétique	50	E	Valeur Obligatoire
quantite_utilisee	Quantité appliquée	Décimal	10	E	Valeur Obligatoire
provenance_intrant	Fournisseur ou origine	Alphabétique	100	E	
date_application	Date d'utilisation	Date	-	E	Valeur Obligatoire
intrant_supprime	Indique si l'intrant a été supprimée	Booléen			Valeur Obligatoire
cout_intrant	Prix payé pour cet intrant	Décimal	10	E	Valeur Obligatoire
ID_suivi	Identifiant de l'observation	Entier	8	E	Auto-incrémenté
date_suivi	Date de l'observation	Date	-	E	Valeur Obligatoire
suivi_supprime	Indique si le suivi a été supprimé	Booléen			Valeur Obligatoire
ID_type_suivi	Identifiant de type suivi	Entier	8	E	Auto-incrémenté
type_suivi	Maladie, attaque d'insectes, etc.	Alphabétique	30	E	Valeur Obligatoire
details_suivi	Description libre	Alphabétique	2000	E	
ID_recolte	Identifiant de la récolte	Entier	8	E	Auto-incrémenté
date_recolte	Date réelle de récolte	Date	-	E	Valeur Obligatoire
recolte_supprime	Indique si la récolte a été supprimée	Booléen			Valeur Obligatoire
quantite_recoltee	Poids ou nombre d'unités récoltées	Décimal	10	E	Valeur Obligatoire
ID_unite	Identifiant de l'unité	Entier	8	E	Auto-incrémenté
Unite	Kg, tonne, sac, botte, etc.	Alphabétique	20	E	Valeur Obligatoire

Tableau 1 : Dictionnaire des données

B. Dépendances fonctionnelles

Les dépendances fonctionnelles servent à montrer quelles données dépendent des autres, pour mieux organiser et éviter les doublons dans notre base de données.

Terrain

(ID_terrain) → (nom_terrain, terrain_supprime)

Localisation

(ID_localisation) → (localisation_terrain)

Parcelle

(ID_parcelle) → (nom_parcelle, superficie_parcelle, disponible, parcelle_supprime, type_sol)

Culture

(ID_culture) → (nom_culture, variete_culture, cycle_culture, saisonnalite_culture, culture_supprime)

Plantation

(ID_plantation) → (date_plantation_terre, methode_culture, quantite_plantee, plantation_supprime)

Intrant

(ID_intrant) → (nom_intrant, quantite_utilisee, provenance_intrant, date_application, cout_intrant, intrant_supprime)

Type intrant

(ID_type_intrant) → (type_intrant)

Suivi

(ID_suivi) → (date_suivi, details_suivi, suivi_supprime)

Type suivi

(ID_type_suivi) → (type_suivi)

Récolte

(ID_recolte) → (date_recolte, quantite_recoltee, recolte_supprime)

Unité

(ID_unite) → (unite)

IV. MODELISATION CONCEPTUELLE

À ce stade du projet, les idées étaient encore dispersées, souvent notées à la hâte sur des feuilles volantes. Il devenait essentiel de structurer l'ensemble pour en avoir une vision claire et partagée. La création du modèle conceptuel s'est donc imposée comme une étape clé.

A. Tableau de la matrice des dépendances fonctionnelles

Le tableau des dépendances fonctionnelles permet de résumer clairement quelles données dépendent d'autres, afin de mieux organiser les informations.

0	1	2	3	4	5	6	7	8	9	10	11
PTE	ID_terrain	ID_parcelle	ID_culture	ID_plantation	ID_intrant	ID_type_intrant	ID_suivi	ID_type_suivi	ID_recolte	ID_unite	ID_localisation
ID_terrain	*										
terrain_supprime	1										
nom_terrain	1										
ID_localisation											*
localisation_terrain											1
ID_parcelle		*									
parcelle_supprime		1									
nom_parcelle		1									
superficie_parcelle		1									
disponible		1									
type_sol		1									
ID_culture			*								
culture_supprime			1								
nom_culture			1								
variete_culture			1								
cycle_culture			1								
saisonnalite_culture			1								
ID_plantation				*							

[illegible]

0	1	2	3	4	5	6	7	8	9	10	11
PTE	ID_terrain	ID_parcelle	ID_culture	ID_plantation	ID_intrant	ID_type_intrant	ID_suivi	ID_type_suivi	ID_recolte	ID_unite	ID_localisation
date_recolte									1		
recolte_supprime									1		
quantite_recoltee									1		
ID_unite										*	
unite										1	

Tableau 2 : Tableau de la MDF

B. Graphe de la matrice des dépendances fonctionnelles

Le graphe de la matrice des dépendances fonctionnelles permet de visualiser les liens entre les données sous forme de schéma, pour mieux comprendre leur organisation

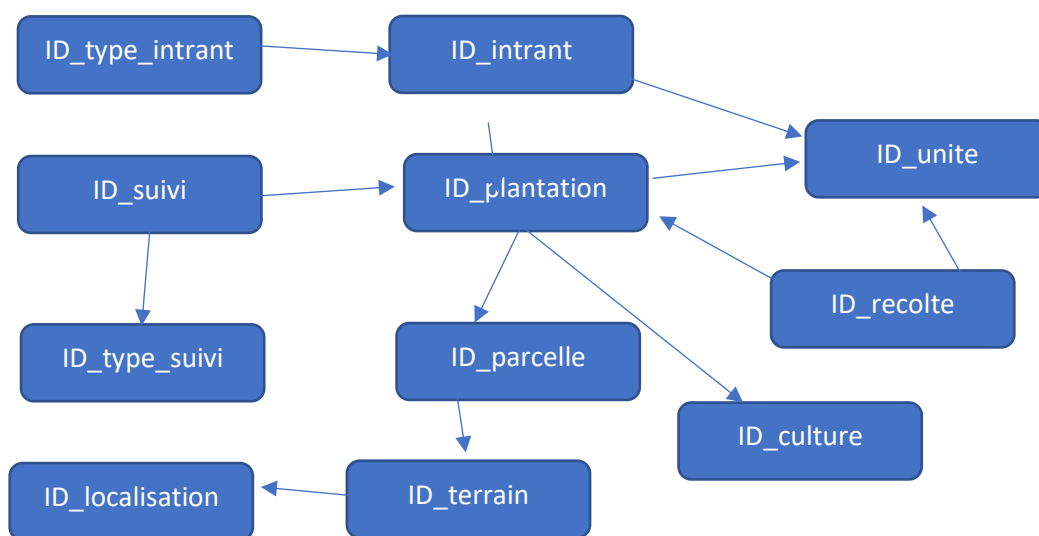


Figure 1 : Graphe de la MDF

C. Modèle Conceptuel des Données

Le Modèle Conceptuel des Données (MCD) sert à représenter de manière claire et simple les entités du projet, leurs propriétés, ainsi que les relations entre elles, avant de procéder à la création de la base de données.

Explication des relations entre les entités du MCD :

Dans le cas de la relation entre Terrain et Parcelle :

Un terrain peut contenir plusieurs parcelles, tandis qu'une parcelle est toujours rattachée à un seul terrain.

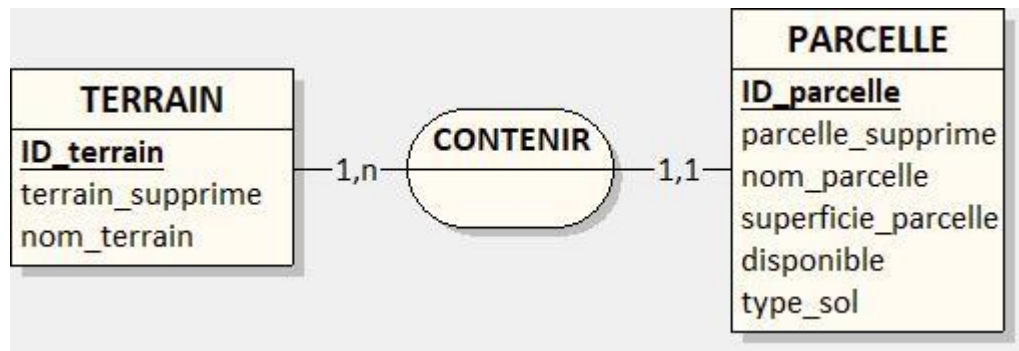


Figure 2 : Relation entre terrain et parcelle

Dans le cas de la relation entre Localisation et Terrain :

Un terrain est situé dans une localisation bien précis, tandis que cette localisation a plusieurs terrains



Figure 3 : Relation entre localisation et terrain

Dans le cas de la relation entre Plantation et Parcelle :

Une parcelle peut accueillir plusieurs plantations au cours du temps, tandis qu'une plantation est toujours réalisée sur une seule parcelle.

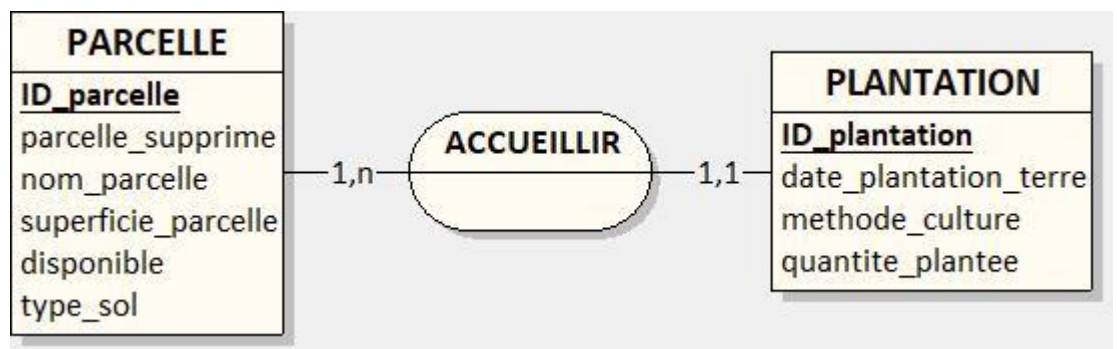


Figure 4 : Relation entre parcelle et plantation

Dans le cas de la relation entre Plantation et Culture :

Une culture peut être utilisée dans plusieurs plantations différentes, tandis qu'une plantation concerne toujours une seule culture.

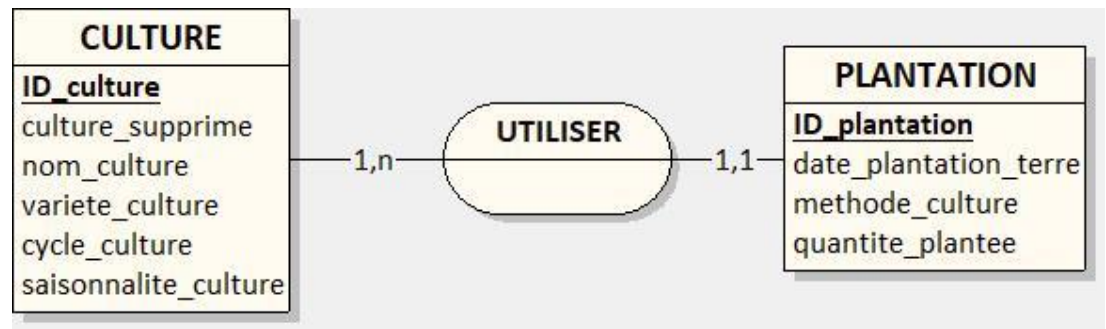


Figure 5 : Relation entre culture et plantation

Dans le cas de la relation entre Plantation et Intranant :

Plusieurs intrants peuvent être appliqués à une même plantation, tandis qu'un intrant n'est appliqué qu'à une seule plantation.

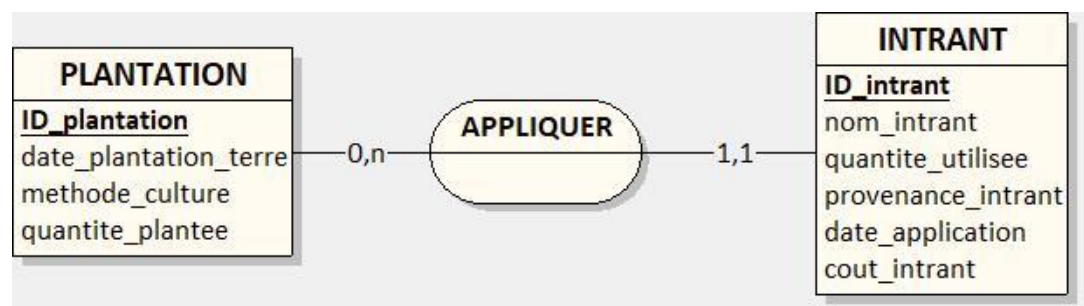


Figure 6 : Relation entre plantation et intrant

Dans le cas de la relation entre Intranant et Type_intranant :

Chaque intrant a un seul type, tandis qu'un type d'intrant peut être commun à plusieurs intrants.

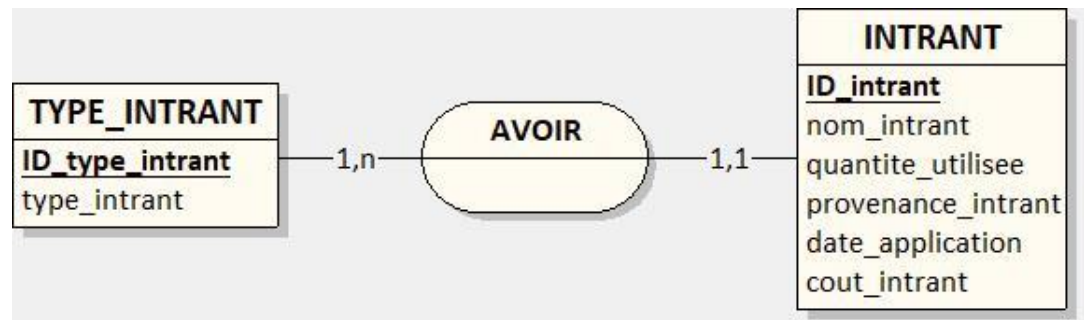


Figure 7 : Relation entre intrant et type intrant

Dans le cas des relations entre Récolte, Intranant et Unité la quantité récoltée est exprimée en une unité de mesure (comme kg ou sac), tout comme la quantité d'un intrant est mesurée en une unité précise. Une même unité peut ainsi être utilisée pour plusieurs récoltes et intrants.

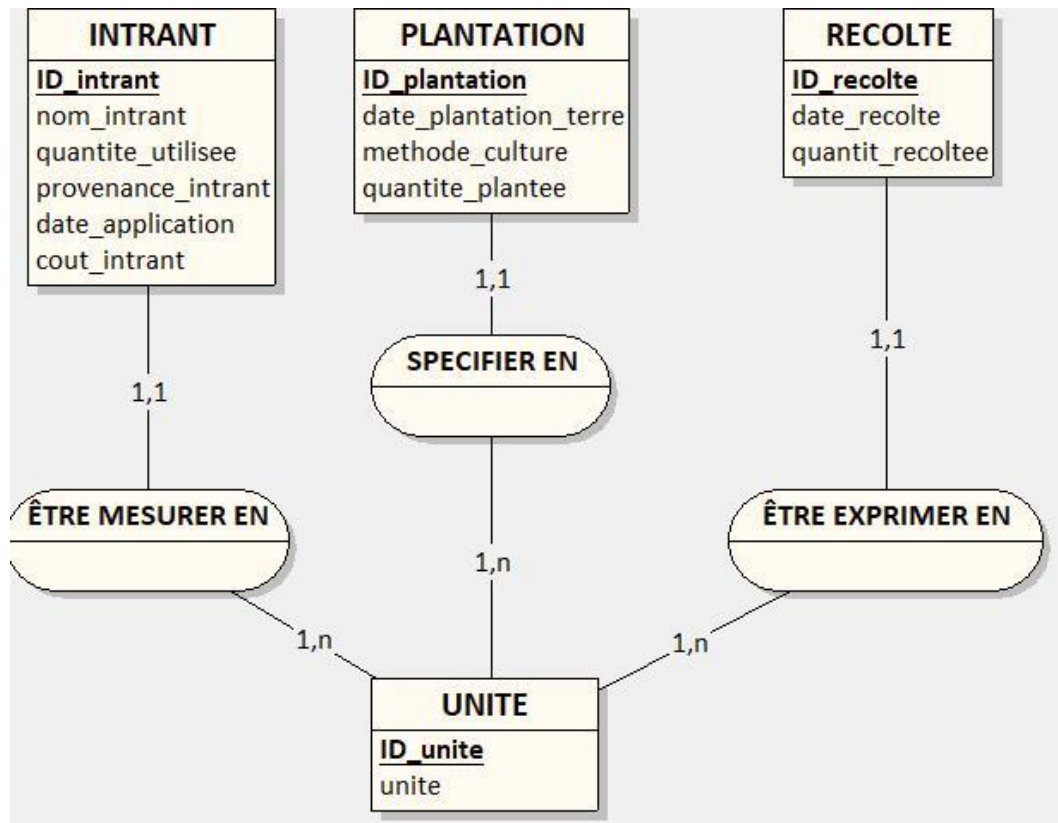


Figure 8 : Relation entre intrant, plantation, récolte et unité

Dans le cas de la relation entre Plantation et Suivi :

Une plantation peut avoir pour suivis plusieurs observations (comme des maladies ou des incidents), tandis qu'un suivi concerne toujours une seule plantation.



Figure 9 : Relation entre suivi et plantation

Dans le cas de la relation entre Suivi et Type_suivi :

Un suivi correspond à un seul type, tandis qu'un type de suivi (comme maladie ou attaque d'insectes) peut être associé à plusieurs suivis.



Figure 10 : Relation entre suivi et type suivi

Voici l'ensemble des relations entre les différentes entités du modèle.

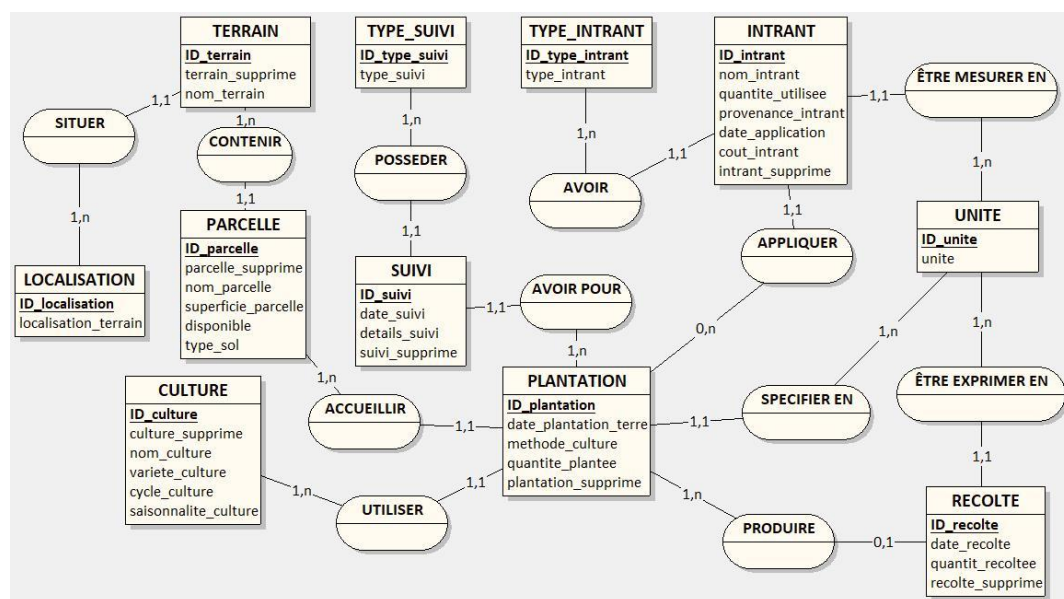


Figure 11 : Modèle Conceptuel des Données

V. NORMALISATION

En avançant dans la conception, j'ai vite remarqué que certaines informations revenaient plusieurs fois dans différentes tables. Cela rendait l'ensemble confus et difficile à maintenir. Pour améliorer la structure, j'ai décidé de passer par une étape de normalisation.

A. 1^{ère} forme normale (1FN)

Chaque attribut contient une valeur unique et atomique. Il ne doit pas y avoir de répétitions ou de groupes de valeurs dans une cellule.

B. 2^e forme normale (2FN)

Tous les attributs non-clés doivent dépendre entièrement de la clé primaire. Il ne doit pas y avoir de dépendance partielle.

C. 3^e forme normale (3FN)

Chaque donnée dépend uniquement de la clé primaire, pas d'une autre colonne.

Étapes de la normalisation

Entité	1FN	2FN	3FN
Terrain	Ok	Ok	Ok
Localisation	Ok	Ok	Ok
Parcelle	Ok	Ok	Ok
Culture	Ok	Ok	Ok
Plantation	Ok	Ok	Ok
Suivi	Ok	Ok	Ok

Entité	1FN	2FN	3FN
Intrant	Ok	Ok	Ok
Recolte	Ok	Ok	Ok
Type_intrant	Ok	Ok	Ok
Type_suivi	Ok	Ok	Ok
Unite	Ok	Ok	Ok

Tableau 3 : Étapes de la normalisation

VI. MODELISATION LOGIQUE DES DONNEES

Une fois que tout était clair conceptuellement, je devais traduire ça en langage de base de données. Cette étape m'a demandé de rigueur, car il fallait choisir les clés primaires, définir les relations, penser aux performances.

Voici donc les relations issues de cette modélisation logique :

Terrain

(ID_terrain, nom_terrain, #ID_localisation)

Localisation

(ID_localisation, localisation_terrain)

Parcelle

(ID_parcelle, nom_parcelle, superficie_parcelle, disponible, type_sol, #ID_terrain)

Culture

(ID_culture, nom_culture, variete_culture, cycle_culture, saisonnalite_culture)

Plantation

(ID_plantation, date_plantation_terre, methode_culture, quantite_plantee, #ID_culture, #ID_parcelle)

Intrant

(ID_intrant, nom_intrant, quantite_utilisee, provenance_intrant, date_application, cout_intrant, #ID_unite, #ID_type_intrant, #ID_plantation)

Type intrant

(ID_type_intrant, type_intrant)

Suivi

(ID_suivi, date_suivi, details_suivi, #ID_plantation, #ID_type_suivi)

Type suivi

(ID_type_suivi, type_suivi)

Recolte

(ID_recolte, date_recolte, quantite_recoltee, #ID_unite, #ID_plantation)

Unite

(ID_unite, unite)

VII. MODELISATION PHYSIQUE DES DONNEES

Après avoir bien réfléchi au modèle, je suis passé à la création des tables dans la base de données. J'ai suivi ce que j'avais défini avant, même si j'ai dû parfois faire quelques ajustements.

Ci-dessous figurent les tables physiques correspondant aux relations du MPD :

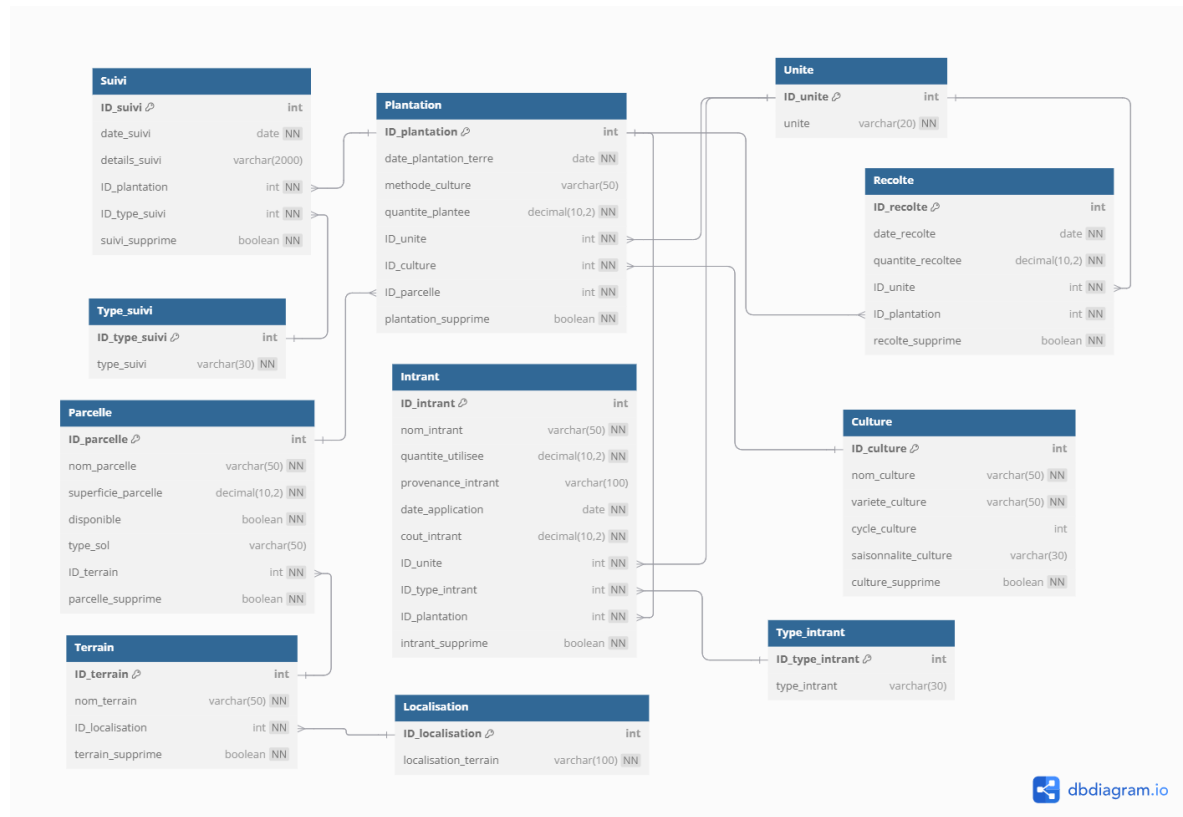


Figure 12 : Modèle Physique des Donnée

VIII. SECURITE ET SAUVEGARDE

Suite à une suppression accidentelle de données, j'ai mis en place un système de suppression logique. Les données ne sont plus effacées définitivement, mais déplacées dans une "corbeille" pour pouvoir être restaurées en cas d'erreur. Cette mesure renforce la sécurité et la fiabilité du système.

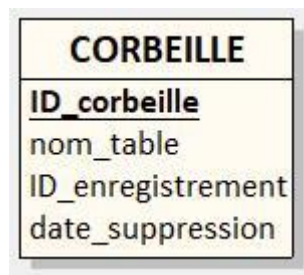


Figure 13 : Représentation de la table corbeille

IX. UTILISATION DU SYSTEME

Le système sera utilisé par les exploitants agricoles et les gestionnaires pour saisir, consulter, modifier, supprimer logiquement et restaurer des données agricoles. Voici quelques exemples concrets d'actions réalisables via les procédures stockées SQL :

1. Ajout de données (procédures add_...)

Ces procédures permettent d'insérer de nouvelles données dans la base.

Exemple : ajouter une nouvelle culture :

- `add_parcelle(nom, superficie, disponible, sol, ID_terrain)`
- `add_intrant(nom, quantité, provenance, date, coût, ID_unité, ID_type, ID_plantation)`
- `add_recolte(date, quantité, ID_unité, ID_plantation, disponible)`
- `add_suivi(date, détails, ID_plantation, ID_type_suivi)`
- `add_terrain(nom, ID_localisation)`
- `add_unite(unite)`
- `add_type_intrant(type)`
- `add_type_suivi(type)`

2. Lecture des données (procédures read_...)

Ces procédures permettent de consulter les données de la base.

Exemple : afficher toutes les plantations :

CALL `read_plantation();`

Autres :

- `read_parcelle()`
- `read_intrant()`
- `read_recolte()`
- `read_terrain()`

3. Suppression logique (procédures del_...)

Plutôt que de supprimer définitivement, les données sont marquées comme supprimées.

Exemple : supprimer une culture :

```
CALL del_culture(3);
```

Cela met culture_supprime = 1 et insère une ligne dans la table corbeille.

Autres :

- del_parcelle(ID)
- del_plantation(ID)
- del_intrant(ID)
- del_recolte(ID)
- del_suivi(ID)
- del_terrain(ID)

4. Consultation des éléments supprimés (corbeille)

Exemple : afficher toutes les cultures supprimées :

```
CALL filter_corbeille('culture', NULL);
```

Afficher tout ce qui est dans la corbeille :

```
CALL filter_corbeille(NULL, NULL);
```

5. Restauration des données supprimées (procédures restore_...)

Exemple : restaurer une culture :

```
CALL restore_culture(3);
```

Autres :

- restore_parcelle(ID)
- restore_plantation(ID)
- restore_intrant(ID)
- restore_recolte(ID)
- restore_suivi(ID)
- restore_terrain(ID)

Restaurer tout d'un coup :

```
CALL restore_all();
```

6. Documentation interne

Lister toutes les procédures disponibles :

```
CALL command_man();
```

Afficher les arguments attendus par une procédure :

```
CALL what_args('add_culture') ;
```


EXEMPLE CONCRET :

Pour illustrer concrètement l'utilisation du système, prenons un exemple vécu dans le cadre de notre projet. Nous avons récemment acquis un terrain situé à Ambatofotsy. Ce terrain a été divisé en deux parcelles, chacune ayant une superficie de 4 m² et un sol sablonneux. Nous les avons nommées : Parcelle A1 et Parcelle A2. Sur ces parcelles, nous avons décidé de planter deux cultures différentes :

- Sur la première parcelle, nous avons cultivé du Pakchoï, variété *Brassica chinensis*, avec un cycle de 45 jours adapté à la saison fraîche.
- Sur la seconde, nous avons planté du Petsaï (chou chinois), variété *Brassica pekinensis*, ayant un cycle de 60 jours, adapté à toutes les saisons.

La date de début des plantations a été fixée au lundi 3 juin 2024. À partir de cette date, nous avons procédé à la semence des graines.

Pendant les six premiers jours, un arrosage quotidien a été nécessaire. Ensuite, nous avons instauré un rythme plus espacé, avec un arrosage tous les deux jours : par exemple, lundi, mercredi, vendredi, puis samedi.

Au 10^e jour, nous avons appliqué de l'engrais chimique. Mais deux jours avant cette application, nous avons effectué un désherbage manuel et une aération du sol entre les rangs (appelée "mikitra"). Il est important de noter qu'aucun arrosage n'a été effectué juste avant l'application de l'engrais.

Au 25^e jour, nous avons enrichi le sol avec du fumier de volaille sec, en suivant la même procédure : désherbage et aération deux jours avant, sans arrosage avant l'apport du fumier.

Au cours de ces travaux, un traitement avec pesticide a également été réalisé pour protéger les cultures.

L'arrosage a ensuite repris normalement.

Le 32^e jour, nous avons procédé à une troisième ouverture de micro-canaux d'irrigation, suivie de deux jours de repos (sans arrosage), avant de reprendre l'irrigation.

Le 45^e jour, la culture de Pakchoï était prête pour la récolte. Pour le Petsaï, la culture a continué avec de l'arrosage, une nouvelle ouverture de micro-canaux au 50^e jour, puis deux jours de séchage avant la reprise de l'arrosage.

Finalement, la culture de Petsaï a été récoltée au 60^e jour.

Nous avons récolté deux sacs de Pakchoï et un demi-sac de Petsaï.

SCRIPT SQL

1. Ajouter une localisation

```
CALL add_localisation('Ambatofotsy');  
SELECT * FROM localisation ;
```

2. Ajouter un terrain

```
CALL add_terrain('Tanety kely', 4);  
SELECT * FROM TERRAIN;
```

3. Ajouter deux parcelles (A1 et A2) de 4m² sablonneux

```
CALL add_parcelle('A1', 4.00, 1, 'Sablonneux', 8);  
CALL add_parcelle('A2', 4.00, 1, 'Sablonneux', 8);
```

4. Ajouter les deux cultures (Pakchoï et Petsai)

```
CALL add_culture('Pakchoï', 'Brassica chinensis', 45, 'Saison  
fraîche');  
CALL add_culture('Petsai', 'Brassica pekinensis', 60, 'Toute saison');
```

5. Ajouter des unités (si pas déjà ajoutées)

```
CALL add_unite('sac');  
CALL add_unite('kg');
```

6. Plantation le 1er juin 2024 (Samedi)

```
SELECT * FROM PARCELLE;  
SELECT * FROM CULTURE;  
SELECT * FROM UNITE;
```

```
CALL add_plantation('2024-06-01', 'Semis direct', 1.00, 17, 11, 22);  
CALL add_plantation('2024-06-01', 'Semis direct', 1.00, 17, 12, 23);
```

7. Suivi : arrosage quotidien pendant 6 jours

```
CALL add_type_suivi('Arrosage');
```

```
CALL read_plantation();  
SELECT * FROM type_suivi ;
```

```
CALL add_suivi('2024-06-01', 'Arrosage quotidien pendant 6 jours après  
semis', 25, 3);
```

```
CALL add_suivi('2024-06-01', 'Arrosage quotidien pendant 6 jours après  
semis', 26, 3);
```

8. Suivi : aération manuelle (mikitra) avant engrais chimique (10e jour)

```
CALL add_type_suivi('Aération manuelle');  
CALL add_suivi('2024-06-09', 'Mikitra avant engrais chimique', 25, 4);  
CALL add_suivi('2024-06-09', 'Mikitra avant engrais chimique', 26, 4);
```

9. Ajout engrais chimique (10e jour)

```
CALL add_type_intrant('Engrais chimique');  
SELECT * FROM type_intrant;  
CALL add_intrant('Engrais NPK', 0.50, 'Magasin local', '2024-06-11',  
5000, 17, 5, 25);  
CALL add_intrant('Engrais NPK', 0.50, 'Magasin local', '2024-06-11',  
5000, 17, 5, 26);
```

10. Aération avant fumier (23e jour)

```
CALL add_suivi('2024-06-23', 'Mikitra avant fumier de volaille',  
25, 4);  
CALL add_suivi('2024-06-23', 'Mikitra avant fumier de volaille',  
26, 4);
```

11. Ajout fumier de volaille (25e jour)

```
CALL add_type_intrant('Fumier');  
CALL add_intrant('Fumier de volaille', 1.0, 'Ferme locale', '2024-  
06-25', 0, 16, 6, 25);  
CALL add_intrant('Fumier de volaille', 1.0, 'Ferme locale', '2024-  
06-25', 0, 16, 6, 26);
```

12. Ajout pesticide (entre 25 et 32e jour)

```
CALL add_type_intrant('Pesticide');  
CALL add_intrant('Pesticide naturel', 0.10, 'Magasin local',  
'2024-06-28', 1000, 17, 7, 25);  
CALL add_intrant('Pesticide naturel', 0.10, 'Magasin local',  
'2024-06-28', 1000, 17, 7, 26);
```

13. Suivi : ouverture micro-canaux (32e jour)

```
CALL add_type_suivi('Ouverture micro-canaux');  
CALL add_suivi('2024-07-03', 'Ouverture micro-canaux + pause 2j',  
25, 5);  
CALL add_suivi('2024-07-03', 'Ouverture micro-canaux + pause 2j',  
26, 5);
```

14. Récolte Pakchoï (45e jour)

```
CALL add_recolte('2024-07-16', 1.00, 16, 25, 1);
```

15. Nouvelle ouverture de micro-canaux pour Petsai (50e jour)

```
CALL add_suivi('2024-07-21', 'Ouverture micro-canaux + pause 2j',  
26, 5);
```

16. Récolte Petsai (60e jour)

```
CALL add_recolte('2024-07-31', 1.50, 16, 26, 1);
```

17. Mise en corbeille de la plantation du Pakchoï

```
CALL read_plantation();  
CALL del_plantation(25);
```

18. Mise en corbeille de la parcelle A1

```
SELECT * FROM PARCELLE;  
CALL del_parcelle(22);
```

19. Vérifier la corbeille

```
CALL filter_corbeille(NULL, NULL);  
CALL read_plantation();  
CALL restore_all() ;  
CALL filter_corbeille(NULL, NULL);  
CALL read_plantation();
```

X. IDÉE D'INTERFACE DE GESTION

Pour finir, j'ai pensé à une interface facile à utiliser, avec un tableau de bord clair. On y verrait tout : les parcelles, les plantations, les intrants, les récoltes... Pas besoin de chercher dans 100 menus. C'est encore une idée, mais j'y crois. Et je sais que ça peut changer notre façon de travailler.

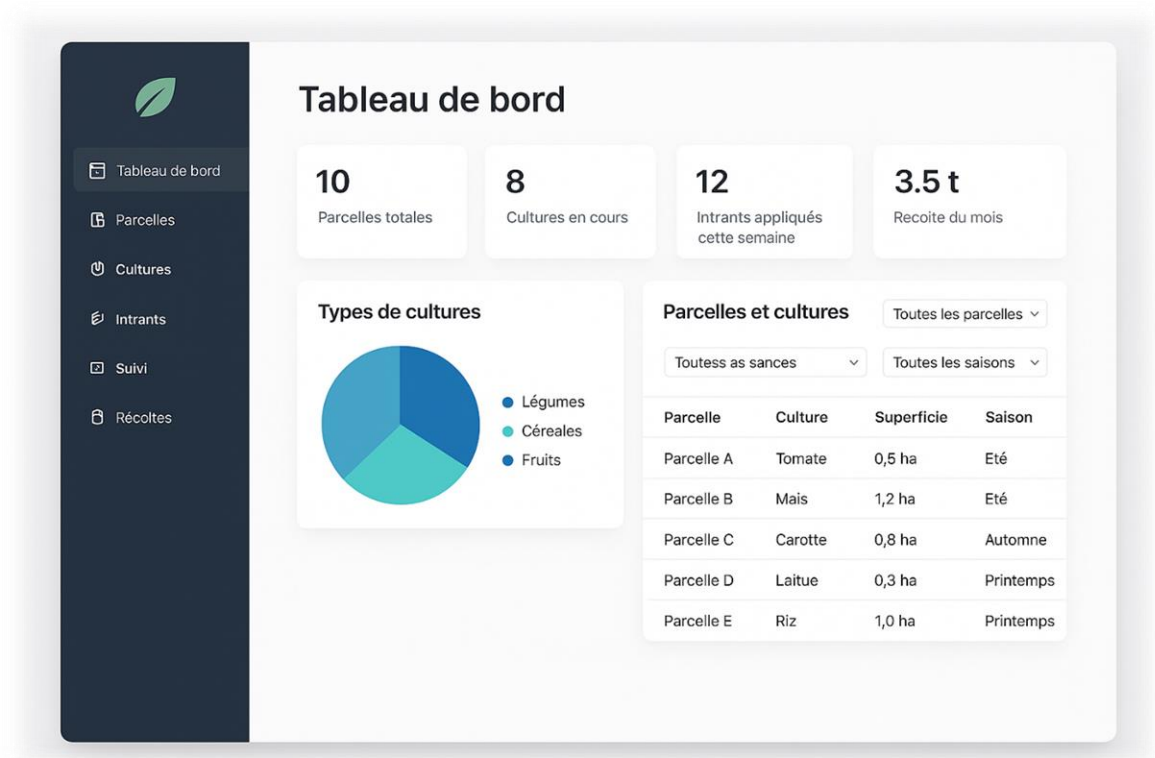


Figure 14 : Exemple de tableau de bord

XI. CONCLUSION DU PROJET

Ce projet est né d'un besoin réel vécu dans mon village d'Androndra. Face aux pertes de temps et au manque d'organisation dans nos activités agricoles, j'ai voulu créer un système simple mais efficace pour centraliser les données, mieux suivre les plantations et prendre de meilleures décisions.

Chaque étape, des erreurs aux améliorations, m'a permis d'apprendre et de rendre ce projet plus solide. Aujourd'hui, ce système est plus qu'un devoir d'étude : c'est un outil concret qui peut vraiment aider les agriculteurs de notre village.