

PROJET 3 : Conception d'une base de donnees

Plateforme de gestion, de suivi et d'analyse du secteur informel

ENSAE Dakar - ISE 2

16 January 2026

Table des matières

	5
Introduction	7
1 Analyse fonctionnelle	9
1.1 Contexte et problematique	9
1.2 Objectif general	9
1.3 Acteurs du systeme	9
1.4 Processus metier	10
2 Diagramme de contexte	11
2.1 Entrees	11
2.2 Sorties	11
3 Modele Conceptuel de Donnees	13
3.1 Entites principales	14
3.2 Associations et cardinalites	15
4 Modele Logique de Donnees	17
4.1 Tables issues des entites	17
4.2 Tables d'association	18
5 Modele Physique de Donnees	19
5.1 Conventions de nommage	19
5.2 Types de donnees utilises	19

5.3	Structure des tables principales	19
5.4	Index de performance	21
6	Dictionnaire des donnees	23
6.1	Tables principales	23
6.2	Tables d association	23
7	Contraintes d integrite	27
7.1	Contraintes d entite	27
7.2	Contraintes referentielles	27
7.3	Contraintes de domaine	27
7.4	Contraintes d unicite	27
8	Scenarios utilisateurs	29
8.1	Cas d utilisation principaux	29
9	Propositions d extension	31
9.1	Ameliorations fonctionnelles	31
9.2	Ameliorations techniques	31
10	Requetes SQL	33
10.1	Requetes simples	33
10.2	Requetes intermediaires	34
10.3	Requetes avancees	35
	Conclusion	37
A	Schema MCD Merise	39
B	Script SQL complet	41

Agence Nationale de la Statistique et de la Démographie (ANSD)
École Nationale de la Statistique et de l'Analyse Économique (ENSAE)
PRÉSENTATION DU PROJET – ANNÉE ACADÉMIQUE 2025-2026
Plateforme de gestion, de suivi et d'analyse du secteur informel
BASE DE DONNEES 1 — Rapport du projet de creation d'une base de donnees
pour le secteur informel au Sénégal
Rédigé par
Paul BALAFAI DIALLO Cheick Oumar Mamady I BERETE
Élèves ingénieurs statisticiens économistes (ISE 2)
Sous la supervision de
M. Demba DIACK
Freelance Senior Data Scientist / ML Engineer
Projet réalisé dans le cadre du cours de base de données 1
Document généré via R Bookdown .

Introduction

Ce rapport presente la conception complete d'une base de donnees dediee a la **gestion, au suivi et a l'analyse du secteur informel au Senegal**. Il s'inscrit dans une logique de production d'indicateurs fiables, territorialisés et exploitables pour l'aide a la decision publique.

Le systeme de gestion de base de donnees relationnelle (SGBDR) retenu est **PostgreSQL**, choisi pour sa robustesse, sa conformite aux standards SQL et ses fonctionnalites avancees.

Chapitre 1

Analyse fonctionnelle

1.1 Contexte et problematique

Au Senegal, le secteur informel occupe une place centrale dans l'emploi et la production, mais il demeure insuffisamment documente. Les donnees existantes sont souvent fragmentees, non standardisees et parfois obsoletes, ce qui limite :

- la production d'indicateurs fiables ;
- la territorialisation des diagnostics ;
- le ciblage efficace des politiques publiques.

1.2 Objectif general

L'objectif est de concevoir une base de donnees permettant de :

- recenser les unites et travailleurs du secteur informel ;
- caracteriser les activites et leur organisation spatiale ;
- suivre l'emploi, les revenus et les charges avec historisation ;
- documenter vulnerabilites, contraintes et besoins ;
- relier les unites aux programmes d'appui et aux resultats obtenus ;
- gerer les documents associes (fiches, photos, autorisations, rapports).

1.3 Acteurs du systeme

Le tableau 1.1 presente les differents acteurs du systeme et leurs responsabilites.

TABLE 1.1 – Acteurs du systeme et leurs responsabilites

Acteur	Responsabilites
Administrateur national	Parametrage des referentiels, gestion des comptes utilisateurs
Agent de recensement	Collecte et mise a jour des donnees terrain
Superviseur regional	Contrôle qualite et validation des donnees
Analyste statistique	Requetes, indicateurs et exports de donnees
Decideur public	Consultation des synthèses et indicateurs

1.4 Processus metier

Les processus metier principaux sont :

1. **Parametrage des referentiels** : regions, departements, communes, sites, activites economiques
2. **Recensement des unites informelles** : identification et caracterisation
3. **Enregistrement des travailleurs** : profil socio-demographique et lien avec l'unite
4. **Saisie d'observations economiques datees** : suivi longitudinal
5. **Declaration des vulnerabilites et besoins** : difficultes, risques, besoins exprimes
6. **Gestion des programmes d'appui** : sessions, participations, resultats
7. **Production d'indicateurs et analyses** : requetes et tableaux de bord

Chapitre 2

Diagramme de contexte

Systeme : Plateforme BD Secteur Informel

2.1 Entrees

- Donnees terrain (recensement, enquetes)
- Referentiels territoriaux et sectoriels
- Programmes publics d'appui
- Documents justificatifs

2.2 Sorties

- Indicateurs economiques et sociaux
- Listes operationnelles (unites, travailleurs, beneficiaires)
- Rapports et extractions statistiques
- Tableaux de suivi des programmes

Chapitre 3

Modele Conceptuel de Donnees

Le Modele Conceptuel de Donnees (Merise) decrit les principales entites, associations et cardinalites necessaires a la gestion du secteur informel.

La figure 3.1 presente le diagramme MCD complet de la base de donnees.

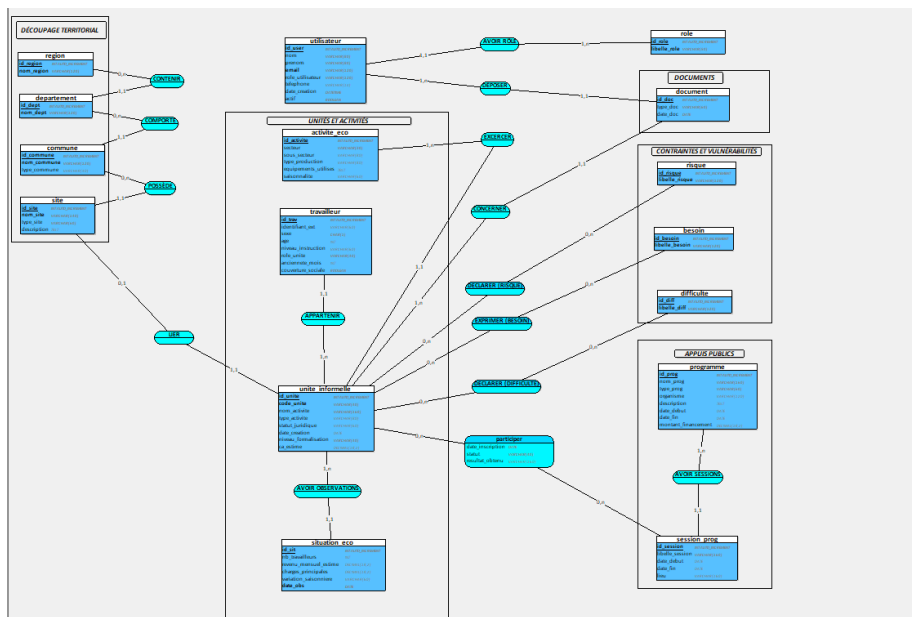


FIGURE 3.1 – Diagramme du Modele Conceptuel de Donnees (MCD)

3.1 Entites principales

Le MCD comprend les entites suivantes organisees en domaines fonctionnels :

3.1.1 Decoupage territorial

- **REGION** : decoupage administratif de niveau 1
- **DEPARTEMENT** : decoupage administratif de niveau 2
- **COMMUNE** : decoupage administratif de niveau 3
- **SITE** : lieu physique d'exercice des activites (marche, zone artisanale, etc.)

3.1.2 Unites et activites

- **ACTIVITE_ECO** : classification des activites economiques (secteur, sous-secteur, type de production)
- **UNITE_INFORMELLE** : entite centrale representant une unite de production informelle
- **TRAVAILLEUR** : personnes travaillant dans les unites informelles
- **EQUIPEMENT** : materiel utilise pour les activites

3.1.3 Suivi economique

- **SITUATION_ECO** : observations economiques datees (revenus, charges, effectifs)

3.1.4 Vulnerabilites

- **DIFFICULTE** : types de difficultes rencontrees
- **RISQUE** : types de risques encourus
- **BESOIN** : types de besoins exprimes

3.1.5 Programmes d'appui

- **PROGRAMME** : programmes publics d'accompagnement
- **SESSION_PROG** : sessions de formation ou d'appui

TABLE 3.1 – Associations et cardinalites du MCD

Association	Entite 1	Card. 1	Entite 2	Card. 2
CONTENIR	REGION	(1,n)	DEPARTEMENT	(1,1)
COMPORTER	DEPARTEMENT	(1,n)	COMMUNE	(1,1)
POSSEDER	COMMUNE	(0,n)	SITE	(1,1)
LIER	SITE	(1,n)	UNITE_INFORMELLE	(1,1)
EXERCER	ACTIVITE_ECO	(1,n)	UNITE_INFORMELLE	(1,1)
APPARTENIR	UNITE_INFORMELLE	(1,n)	TRAVAILLEUR	(1,1)
AVOIR_OBSERVATIONS	UNITE_INFORMELLE	(1,n)	SITUATION_ECO	(1,1)
UTILISER	ACTIVITE_ECO	(0,n)	EQUIPEMENT	(0,n)
DECLARER_DIFFICULTE	UNITE_INFORMELLE	(0,n)	DIFFICULTE	(0,n)
DECLARER_RISQUE	UNITE_INFORMELLE	(0,n)	RISQUE	(0,n)
EXPRIMER_BESOIN	UNITE_INFORMELLE	(0,n)	BESOIN	(0,n)
AVOIR_SESSIONS	PROGRAMME	(1,n)	SESSION_PROG	(1,1)
PARTICIPER	UNITE_INFORMELLE	(0,n)	SESSION_PROG	(0,n)
DEPOSER	UTILISATEUR	(0,n)	DOCUMENT	(1,1)
CONCERNER	UNITE_INFORMELLE	(0,n)	DOCUMENT	(1,1)
AVOIR_ROLE	ROLE	(1,n)	UTILISATEUR	(1,1)

3.1.6 Gestion documentaire

- **DOCUMENT** : pieces justificatives et rapports
- **UTILISATEUR** : agents du systeme
- **ROLE** : profils d'accès

3.2 Associations et cardinalites

Le tableau 3.1 presente les associations entre entites et leurs cardinalites.

Chapitre 4

Modele Logique de Donnees

La traduction du MCD vers le modele relationnel suit les regles classiques de transformation Merise.

4.1 Tables issues des entites

4.1.1 Referentiels utilisateurs

- **ROLE** (#id_role, libelle_role)
- **UTILISATEUR** (#id_user, nom, prenom, email, telephone, date_creation, actif, @id_role)

4.1.2 Decoupage territorial

- **REGION** (#id_region, nom_region)
- **DEPARTEMENT** (#id_dept, nom_dept, @id_region)
- **COMMUNE** (#id_commune, nom_commune, type_commune, @id_dept)
- **SITE** (#id_site, nom_site, type_site, description, @id_commune)

4.1.3 Unites et activites

- **ACTIVITE_ECO** (#id_activite, secteur, sous_secteur, type_production, saisonnalite, description)
- **UNITE_INFORMELLE** (#id_unite, code_unite, nom_activite, type_activite, statut_juridique, date_creation, niveau_formalisation, ca_estime, @id_site, @id_activite)

- **TRAVAILLEUR** (#id_trav, identifiant_ext, sexe, age, niveau_instruction, role_unite, anciennete_mois, couverture_sociale, @id_unite)

4.1.4 Suivi economique

- **SITUATION_ECO** (#id_sit, nb_travailleurs_decl, revenu_mensuel_estime, charges_principales, variation_saisonniere, date_obs, @id_unite)

4.1.5 Vulnerabilites (referentiels)

- **DIFFICULTE** (#id_diff, libelle_diff)
- **RISQUE** (#id_risque, libelle_risque)
- **BESOIN** (#id_besoin, libelle_besoin)
- **EQUIPEMENT** (#id_equipement, libelle_equipement)

4.1.6 Programmes

- **PROGRAMME** (#id_prog, nom_prog, type_prog, organisme, description, date_debut, date_fin)
- **SESSION_PROG** (#id_session, libelle_session, date_debut, date_fin, lieu, @id_prog)

4.1.7 Documents

- **DOCUMENT** (#id_doc, type_doc, date_doc, chemin_fichier, commentaire, @id_user, @id_unite)

4.2 Tables d'association

- **UTILISER** (@id_activite, @id_equipement, remarque)
- **DECLARER_DIFFICULTE** (@id_unite, @id_diff, date_decl, niveau_severite)
- **DECLARER_RISQUE** (@id_unite, @id_risque, date_decl, probabilite_percue)
- **EXPRIMER_BESOIN** (@id_unite, @id_besoin, date_decl, priorite)
- **PARTICIPER** (@id_unite, @id_session, date_inscription, statut, resultat_obtenu, montant_financement)

Chapitre 5

Modele Physique de Donnees

Le MPD est implemente en **PostgreSQL** et respecte les bonnes pratiques de conception.

5.1 Conventions de nommage

- Noms de tables en minuscules avec underscores
- Prefixe **pk_** pour les cles primaires
- Prefixe **fk_** pour les cles etrangeres
- Prefixe **uq_** pour les contraintes d’unicite
- Prefixe **ck_** pour les contraintes de verification
- Prefixe **idx_** pour les index

5.2 Types de donnees utilises

Le tableau 5.1 presente les types de donnees PostgreSQL utilises.

5.3 Structure des tables principales

5.3.1 Table **UNITE_INFORMELLE**

TABLE 5.1 – Types de donnees PostgreSQL utilises

Type PostgreSQL	Utilisation
SERIAL	Identifiants auto-incrementes
VARCHAR(n)	Chaines de caracteres de taille limitee
TEXT	Textes longs (descriptions, commentaires)
INT	Entiers (age, effectifs, durees)
NUMERIC(18,2)	Montants monetaires
DATE	Dates sans heure
TIMESTAMP	Dates avec heure
BOOLEAN	Valeurs booleennes
CHAR(1)	Codes courts (sexe M/F)

```
CREATE TABLE unite_informelle (
  id_unite SERIAL CONSTRAINT pk_unite PRIMARY KEY,
  code_unite VARCHAR(40) CONSTRAINT uq_unite_code UNIQUE NOT NULL,
  nom_activite VARCHAR(160) NOT NULL,
  type_activite VARCHAR(80),
  statut_juridique VARCHAR(60),
  date_creation DATE,
  niveau_formalisation VARCHAR(40) NOT NULL,
  ca_estime NUMERIC(18,2),
  id_site INT NOT NULL,
  id_activite INT NOT NULL,
  CONSTRAINT fk_unite_site FOREIGN KEY (id_site) REFERENCES site(id_site),
  CONSTRAINT fk_unite_activite FOREIGN KEY (id_activite)
    REFERENCES activite_eco(id_activite),
  CONSTRAINT ck_unite_ca CHECK (ca_estime IS NULL OR ca_estime >= 0)
);
```

5.3.2 Table TRAVAILLEUR

```
CREATE TABLE travailleur (
  id_trav SERIAL CONSTRAINT pk_travailleur PRIMARY KEY,
  identifiant_ext VARCHAR(60),
  sexe CHAR(1) NOT NULL,
  age INT,
  niveau_instruction VARCHAR(60),
  role_unite VARCHAR(40) NOT NULL,
  anciennete_mois INT,
  couverture_sociale BOOLEAN NOT NULL DEFAULT FALSE,
```

```
id_unite INT NOT NULL,  
CONSTRAINT ck_trav_sexe CHECK (sexe IN ('M','F')),  
CONSTRAINT ck_trav_age CHECK (age IS NULL OR age BETWEEN 10 AND 100),  
CONSTRAINT ck_trav_anciennete CHECK (anciennete_mois IS NULL  
    OR anciennete_mois >= 0),  
CONSTRAINT fk_trav_unite FOREIGN KEY (id_unite)  
    REFERENCES unite_informelle(id_unite)  
);
```

5.4 Index de performance

```
-- Index sur les cles etrangeres frequemment utilisees  
CREATE INDEX idx_unite_site ON unite_informelle(id_site);  
CREATE INDEX idx_unite_activite ON unite_informelle(id_activite);  
CREATE INDEX idx_trav_unite ON travailleur(id_unite);  
CREATE INDEX idx_sit_unite_date ON situation_eco(id_unite, date_obs);  
  
-- Index sur les tables d'association  
CREATE INDEX idx_utiliser_act ON utiliser(id_activite);  
CREATE INDEX idx_dd_unite ON declarer_difficulte(id_unite);  
CREATE INDEX idx_dr_unite ON declarer_risque(id_unite);  
CREATE INDEX idx_eb_unite ON exprimer_besoin(id_unite);  
CREATE INDEX idx_participer_session ON participer(id_session);  
CREATE INDEX idx_doc_unite ON document(id_unite);
```


Chapitre 6

Dictionnaire des donnees

6.1 Tables principales

6.1.1 Table ROLE

6.1.2 Table UTILISATEUR

6.1.3 Table UNITE_INFORMELLE

6.1.4 Table TRAVAILLEUR

6.1.5 Table SITUATION_ECO

6.2 Tables d association

6.2.1 Table PARTICIPER

6.2.2 Table DECLARER_DIFFICULTE

TABLE 6.1 – Structure de la table ROLE

Champ	Type	Contraintes	Description
id_role	SERIAL	PK	Identifiant unique du role
libelle_role	VARCHAR(50)	UNIQUE, NOT NULL	Libelle du role

TABLE 6.2 – Structure de la table UTILISATEUR

Champ	Type	Contraintes	Description
id_user	SERIAL	PK	Identifiant unique
nom	VARCHAR(80)	NOT NULL	Nom de l'utilisateur
prenom	VARCHAR(80)	NOT NULL	Prenom de l'utilisateur
email	VARCHAR(120)	UNIQUE, NOT NULL	Adresse email
telephone	VARCHAR(30)	-	Numero de telephone
date_creation	TIMESTAMP	NOT NULL, DEFAULT NOW()	Date de creation du compte
actif	BOOLEAN	NOT NULL, DEFAULT TRUE	Statut du compte
id_role	INT	FK vers role	Role de l'utilisateur

TABLE 6.3 – Structure de la table UNITE_INFORMELLE

Champ	Type	Contraintes	Description
id_unite	SERIAL	PK	Identifiant unique
code_unite	VARCHAR(40)	UNIQUE, NOT NULL	Code d'identification
nom_activite	VARCHAR(160)	NOT NULL	Nom de l'activite
type_activite	VARCHAR(80)	-	Type d'activite
statut_juridique	VARCHAR(60)	-	Statut juridique
date_creation	DATE	-	Date de creation
niveau_formalisation	VARCHAR(40)	NOT NULL	Niveau de formalisation
ca_estime	NUMERIC(18,2)	>= 0	Chiffre d'affaires estimé
id_site	INT	FK vers site, NOT NULL	Site de localisation
id_activite	INT	FK vers activite_eco, NOT NULL	Activite economique

TABLE 6.4 – Structure de la table TRAVAILLEUR

Champ	Type	Contraintes	Description
id_trav	SERIAL	PK	Identifiant unique
identifiant_ext	VARCHAR(60)	-	Identifiant externe
sexe	CHAR(1)	NOT NULL, IN (M,F)	Sexe du travailleur
age	INT	BETWEEN 10 AND 100	Age
niveau_instruction	VARCHAR(60)	-	Niveau d'instruction
role_unite	VARCHAR(40)	NOT NULL	Role dans l'unite
anciennete_mois	INT	>= 0	Anciennete en mois
couverture_sociale	BOOLEAN	NOT NULL, DEFAULT FALSE	Protection sociale
id_unite	INT	FK vers unite_informelle	Unite d'appartenance

TABLE 6.5 – Structure de la table SITUATION_ECO

Champ	Type	Contraintes	Description
id_sit	SERIAL	PK	Identifiant unique
nb_travailleurs_decl	INT	≥ 0	Nombre de travailleurs declares
revenu_mensuel_estime	NUMERIC(18,2)	≥ 0	Revenu mensuel estime
charges_principales	NUMERIC(18,2)	≥ 0	Charges principales
variation_saisonniere	VARCHAR(60)	-	Type de variation saisonniere
date_obs	DATE	NOT NULL	Date d observation
id_unite	INT	FK vers unite_informelle	Unite concerne

TABLE 6.6 – Structure de la table PARTICIPER

Champ	Type	Contraintes	Description
id_unite	INT	PK, FK vers unite_informelle	Unite participante
id_session	INT	PK, FK vers session_prog	Session concerne
date_inscription	DATE	NOT NULL	Date d inscription
statut	VARCHAR(40)	NOT NULL	Statut de participation
resultat_obtenu	VARCHAR(160)	-	Resultat obtenu
montant_financement	NUMERIC(18,2)	≥ 0	Montant du financement

TABLE 6.7 – Structure de la table DECLARER_DIFFICULTE

Champ	Type	Contraintes	Description
id_unite	INT	PK, FK vers unite_informelle	Unite declarante
id_diff	INT	PK, FK vers difficulte	Type de difficulte
date_decl	DATE	PK, NOT NULL	Date de declaration
niveau_severite	INT	BETWEEN 1 AND 5	Niveau de severite

Chapitre 7

Contraintes d integrite

7.1 Contraintes d entite

Toutes les tables possedent une cle primaire (SERIAL ou composite) garantissant l unicite des enregistrements.

7.2 Contraintes referentielles

Le tableau 7.1 presente les contraintes referentielles du schema.

7.3 Contraintes de domaine

7.4 Contraintes d unicite

TABLE 7.1 – Contraintes referentielles

Table source	Cle etrangere	Table cible	ON DELETE
utilisateur	id_role	role	RESTRICT
departement	id_region	region	RESTRICT
commune	id_dept	departement	RESTRICT
site	id_commune	commune	RESTRICT
unite_informelle	id_site, id_activite	site, activite_eco	RESTRICT
travailleur	id_unite	unite_informelle	RESTRICT
situation_eco	id_unite	unite_informelle	RESTRICT
utiliser	id_activite, id_equipement	activite_eco, equipement	CASCADE
declarer_difficulte	id_unite, id_diff	unite_informelle, difficulte	CASCADE
session_prog	id_prog	programme	CASCADE
participer	id_unite, id_session	unite_informelle, session_prog	CASCADE
document	id_user, id_unite	utilisateur, unite_informelle	RESTRICT

TABLE 7.2 – Contraintes de domaine

Table	Contrainte	Expression
travailleur	ck_trav_sexe	sexe IN (M,F)
travailleur	ck_trav_age	age BETWEEN 10 AND 100
travailleur	ck_trav_anciennete	anciennete_mois >= 0
unite_informelle	ck_unite_ca	ca_estime >= 0
situation_eco	ck_sit_*	valeurs >= 0
programme	ck_prog_dates	date_fin >= date_debut
declarer_difficulte	ck_dd_niveau	niveau_severite BETWEEN 1 AND 5
declarer_risque	ck_dr_prob	probabilite_percue BETWEEN 1 AND 5
exprimer_besoin	ck_eb_priorite	priorite BETWEEN 1 AND 5

TABLE 7.3 – Contraintes d unicite

Table	Contrainte	Colonnes
role	uq_role_libelle	libelle_role
utilisateur	uq_utilisateur_email	email
region	uq_region_nom	nom_region
departement	uq_departement_nom_region	(nom_dept, id_region)
commune	uq_commune_nom_dept	(nom_commune, id_dept)
site	uq_site_nom_commune	(nom_site, id_commune)
unite_informelle	uq_unite_code	code_unite
situation_eco	uq_sit_unite_date	(id_unite, date_obs)
equipement	uq_equipement_libelle	libelle_equipement
difficulte	uq_difficulte_libelle	libelle_diff
risque	uq_risque_libelle	libelle_risque
besoin	uq_besoin_libelle	libelle_besoin

Chapitre 8

Scenarios utilisateurs

8.1 Cas d utilisation principaux

8.1.1 UC1 : Gestion des comptes et referentiels

- **Acteur** : Administrateur national
- **Actions** : Creer/modifier/desactiver des comptes utilisateurs, gerer les referentiels (activites, equipements, difficultes, risques, besoins)

8.1.2 UC2 : Recensement des unites informelles

- **Acteur** : Agent de recensement
- **Actions** : Creer une nouvelle unite, renseigner les caracteristiques, associer a un site et une activite

8.1.3 UC3 : Enregistrement des travailleurs

- **Acteur** : Agent de recensement
- **Actions** : Ajouter un travailleur a une unite, renseigner le profil socio-demographique

8.1.4 UC4 : Suivi economique

- **Acteur** : Agent de recensement
- **Actions** : Saisir une observation economique datee (revenus, charges, effectifs)

8.1.5 UC5 : Declaration des vulnerabilites

- **Acteur** : Agent de recensement
- **Actions** : Declarer difficultes, risques et besoins avec niveau de seve-rite/priorite

8.1.6 UC6 : Gestion des programmes d appui

- **Acteur** : Superviseur regional
- **Actions** : Creer un programme, planifier des sessions, inscrire des unites participantes

8.1.7 UC7 : Analyse et indicateurs

- **Acteur** : Analyste statistique
- **Actions** : Executer des requetes, produire des indicateurs, exporter des donnees

Chapitre 9

Propositions d extension

9.1 Ameliorations fonctionnelles

1. **Geolocalisation** : Ajouter des coordonnees GPS (latitude, longitude) a la table SITE pour la cartographie
2. **Historisation du niveau de formalisation** : Creer une table HISTORIQUE_FORMALISATION pour suivre l evolution
3. **Anonymisation et vues statistiques** : Creer des vues agregees sans donnees nominatives pour les decideurs
4. **Detection de doublons** : Mettre en place des contraintes et triggers pour identifier les unites potentiellement dupliquees
5. **Indice synthetique de vulnerabilite** : Calculer un score composite base sur difficultes, risques et besoins

9.2 Ameliorations techniques

1. **Partitionnement** : Partitionner SITUATION_ECO par annee pour ameliorer les performances
2. **Audit trail** : Ajouter des colonnes created_at, updated_at, created_by, updated_by
3. **Soft delete** : Implementer la suppression logique plutot que physique
4. **Full-text search** : Activer la recherche plein texte sur les descriptions

Chapitre 10

Requetes SQL

10.1 Requetes simples

10.1.1 R1 : Liste des regions

```
SELECT id_region, nom_region
FROM region
ORDER BY nom_region;
```

10.1.2 R2 : Nombre d unites par site

```
SELECT s.nom_site, COUNT(u.id_unite) AS nb_unites
FROM site s
LEFT JOIN unite_informelle u ON s.id_site = u.id_site
GROUP BY s.id_site, s.nom_site
ORDER BY nb_unites DESC;
```

10.1.3 R3 : Travailleurs sans couverture sociale

```
SELECT t.id_trav, t.sexe, t.age, t.role_unite
FROM travailleur t
WHERE t.couverture_sociale = FALSE;
```

10.2 Requetes intermediaires

10.2.1 R4 : Revenu moyen par secteur d activite

```
SELECT a.secteur,  
       AVG(se.revenu_mensuel_estime) AS revenu_moyen,  
       COUNT(DISTINCT u.id_unite) AS nb_unites  
FROM activite_eco a  
JOIN unite_informelle u ON a.id_activite = u.id_activite  
JOIN situation_eco se ON u.id_unite = se.id_unite  
GROUP BY a.secteur  
ORDER BY revenu_moyen DESC;
```

10.2.2 R5 : Unites ayant declare plus de 3 difficultes

```
SELECT u.code_unite, u.nom_activite, COUNT(dd.id_diff) AS nb_difficultes  
FROM unite_informelle u  
JOIN declarer_difficulte dd ON u.id_unite = dd.id_unite  
GROUP BY u.id_unite, u.code_unite, u.nom_activite  
HAVING COUNT(dd.id_diff) > 3  
ORDER BY nb_difficultes DESC;
```

10.2.3 R6 : Taux de couverture sociale par region

```
SELECT r.nom_region,  
       COUNT(t.id_trav) AS nb_travailleurs,  
       SUM(CASE WHEN t.couverture_sociale THEN 1 ELSE 0 END) AS avec_couverture,  
       ROUND(100.0 * SUM(CASE WHEN t.couverture_sociale THEN 1 ELSE 0 END)  
            / COUNT(t.id_trav), 2) AS taux_couverture  
FROM region r  
JOIN departement d ON r.id_region = d.id_region  
JOIN commune c ON d.id_dept = c.id_dept  
JOIN site s ON c.id_commune = s.id_commune  
JOIN unite_informelle u ON s.id_site = u.id_site  
JOIN travailleur t ON u.id_unite = t.id_unite  
GROUP BY r.id_region, r.nom_region  
ORDER BY taux_couverture DESC;
```

10.3 Requetes avancees

10.3.1 R7 : Evolution du revenu moyen par trimestre

```
SELECT
    EXTRACT(YEAR FROM se.date_obs) AS annee,
    EXTRACT(QUARTER FROM se.date_obs) AS trimestre,
    AVG(se.revenu_mensuel_estime) AS revenu_moyen,
    COUNT(DISTINCT se.id_unite) AS nb_observations
FROM situation_eco se
GROUP BY EXTRACT(YEAR FROM se.date_obs), EXTRACT(QUARTER FROM se.date_obs)
ORDER BY annee, trimestre;
```

10.3.2 R8 : Top 10 des programmes avec le plus de participants

```
SELECT p.nom_prog, p.type_prog, p.organisme,
    COUNT(DISTINCT par.id_unite) AS nb_participants,
    SUM(par.montant_financement) AS total_financement
FROM programme p
JOIN session_prog sp ON p.id_prog = sp.id_prog
JOIN participer par ON sp.id_session = par.id_session
GROUP BY p.id_prog, p.nom_prog, p.type_prog, p.organisme
ORDER BY nb_participants DESC
LIMIT 10;
```

10.3.3 R9 : Indice de vulnerabilite par unite

```
WITH scores AS (
    SELECT u.id_unite, u.code_unite, u.nom_activite,
        COALESCE(AVG(dd.niveau_severite), 0) AS score_difficulte,
        COALESCE(AVG(dr.probabilite_percue), 0) AS score_risque,
        COALESCE(AVG(eb.priorite), 0) AS score_besoin
    FROM unite_informelle u
    LEFT JOIN declarer_difficulte dd ON u.id_unite = dd.id_unite
    LEFT JOIN declarer_risque dr ON u.id_unite = dr.id_unite
    LEFT JOIN exprimer_besoin eb ON u.id_unite = eb.id_unite
    GROUP BY u.id_unite, u.code_unite, u.nom_activite
)
SELECT code_unite, nom_activite,
```

```
ROUND(score_difficulte, 2) AS score_difficulte,  
ROUND(score_risque, 2) AS score_risque,  
ROUND(score_besoin, 2) AS score_besoin,  
ROUND((score_difficulte + score_risque + score_besoin) / 3, 2)  
    AS indice_vulnerabilite  
FROM scores  
WHERE (score_difficulte + score_risque + score_besoin) > 0  
ORDER BY indice_vulnerabilite DESC;
```

10.3.4 R10 : Tableau croise activites x difficultes

```
SELECT a.secteur, d.libelle_diff, COUNT(*) AS nb_declarations  
FROM activite_eco a  
JOIN unite_informelle u ON a.id_activite = u.id_activite  
JOIN declarer_difficulte dd ON u.id_unite = dd.id_unite  
JOIN difficulte d ON dd.id_diff = d.id_diff  
GROUP BY a.secteur, d.libelle_diff  
ORDER BY a.secteur, nb_declarations DESC;
```

Conclusion

Ce projet a permis de concevoir une base de données relationnelle complète pour la gestion du secteur informel au Sénégal. La méthodologie Merise a guidé la modélisation depuis l'analyse des besoins jusqu'à l'implémentation PostgreSQL.

Les points clés de cette conception sont :

1. **Couverture fonctionnelle complète** : référentiels, unités, travailleurs, suivi économique, vulnérabilités, programmes d'appui et documents
2. **Intégrité des données** : contraintes référentielles, de domaine et d'unicité rigoureuses
3. **Historisation** : observations économiques et déclarations de vulnérabilités datées
4. **Extensibilité** : structure modulaire permettant des évolutions futures
5. **Performance** : index optimisés pour les requêtes fréquentes

Cette base de données constitue un socle solide pour la production d'indicateurs fiables et l'aide à la décision en matière de politiques publiques d'accompagnement du secteur informel.

Annexe A

Schema MCD Merise

Le diagramme MCD complet est fourni dans le fichier `Projet_looping.luo`.

Annexe B

Script SQL complet

Le script DDL PostgreSQL complet est fourni dans le fichier `MPD.sql`.