**Structure de la base de données :**

**Tables principales :**

* utilisateurs : Gestion des membres et administrateurs avec leurs profils
* donnees\_principales : Stockage des données importées ou saisies manuellement
* informations\_complementaires : Données additionnelles liées aux enregistrements principaux
* departements et communes : Référentiels géographiques du Bénin

**Tables de gestion :**

* importations : Historique des fichiers Excel importés
* logs\_activite : Traçabilité de toutes les actions
* sessions\_utilisateur : Gestion des tokens JWT
* documents\_generes : Historique des PDF générés

**Fonctionnalités intégrées :**

* ✅ Système de rôles (administrateur/membre)
* ✅ Restriction d'accès par département/commune
* ✅ Traçabilité complète des actions
* ✅ Gestion du statut de traitement des données
* ✅ Support pour l'importation Excel
* ✅ Indexation optimisée pour les performances
* ✅ Données géographiques du Bénin pré-remplies

**Sécurité :**

* Contraintes de clés étrangères
* Index pour optimiser les requêtes
* Support UTF8 pour les caractères spéciaux
* Structure prête pour le hachage des mots de passe

-- Base de données pour la plateforme ADECOB - Gestion des infrastructures socio-communautaires

-- Création de la base de données

CREATE DATABASE IF NOT EXISTS adecob\_platform

CHARACTER SET utf8mb4

COLLATE utf8mb4\_unicode\_ci;

USE adecob\_platform;

-- Table des départements du Bénin

CREATE TABLE departements (

id INT AUTO\_INCREMENT PRIMARY KEY,

nom VARCHAR(100) NOT NULL UNIQUE,

code VARCHAR(10),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Table des communes

CREATE TABLE communes (

id INT AUTO\_INCREMENT PRIMARY KEY,

nom VARCHAR(100) NOT NULL,

departement\_id INT NOT NULL,

code VARCHAR(10),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (departement\_id) REFERENCES departements(id) ON DELETE CASCADE,

INDEX idx\_departement (departement\_id)

);

-- Table des utilisateurs

CREATE TABLE utilisateurs (

id INT AUTO\_INCREMENT PRIMARY KEY,

nom VARCHAR(100) NOT NULL,

prenom VARCHAR(100) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

telephone VARCHAR(20),

mot\_de\_passe VARCHAR(255) NOT NULL,

role ENUM('administrateur', 'membre') DEFAULT 'membre',

departement\_id INT,

commune\_id INT,

statut ENUM('actif', 'inactif', 'suspendu') DEFAULT 'actif',

date\_inscription TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

derniere\_connexion TIMESTAMP NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (departement\_id) REFERENCES departements(id) ON DELETE SET NULL,

FOREIGN KEY (commune\_id) REFERENCES communes(id) ON DELETE SET NULL,

INDEX idx\_email (email),

INDEX idx\_role (role),

INDEX idx\_departement\_commune (departement\_id, commune\_id)

);

-- Table principale des infrastructures (adaptée aux données Excel)

CREATE TABLE infrastructures (

id INT AUTO\_INCREMENT PRIMARY KEY,

-- Informations d'enquête

start VARCHAR(150),

end VARCHAR(150),

date\_enquete DATE,

nom\_enqueteur VARCHAR(150),

numero\_enquete VARCHAR(50),

-- Localisation administrative

commune VARCHAR(100),

parakou BOOLEAN DEFAULT FALSE,

tchaourou BOOLEAN DEFAULT FALSE,

ndali BOOLEAN DEFAULT FALSE,

nikki BOOLEAN DEFAULT FALSE,

bembereke BOOLEAN DEFAULT FALSE,

kalale BOOLEAN DEFAULT FALSE,

sinende BOOLEAN DEFAULT FALSE,

perere BOOLEAN DEFAULT FALSE,

-- Localisation géographique détaillée

village\_quartier VARCHAR(200),

hameau VARCHAR(200),

secteur\_domaine VARCHAR(200),

-- Informations sur l'infrastructure

type\_infrastructure VARCHAR(200),

nom\_infrastructure VARCHAR(300),

annee\_realisation YEAR,

bailleur VARCHAR(200),

type\_materiaux TEXT,

-- État et gestion

etat\_fonctionnement ENUM('Bon', 'Moyen', 'Mauvais', 'Hors service', 'Non fonctionnel') DEFAULT 'Bon',

niveau\_degradation ENUM('Aucun', 'Faible', 'Moyen', 'Élevé', 'Très élevé') DEFAULT 'Aucun',

mode\_gestion VARCHAR(200),

precise TEXT,

-- Observations et recommandations

defectuosites\_relevees TEXT,

mesures\_proposees TEXT,

observation\_generale TEXT,

-- Photos et documents

photo1\_url VARCHAR(500),

photo2\_url VARCHAR(500),

photo3\_url VARCHAR(500),

photo4\_url VARCHAR(500),

-- Besoins de réhabilitation

rehabilitation BOOLEAN DEFAULT FALSE,

-- Géolocalisation

localisation DECIMAL(10, 8),

latitude DECIMAL(10, 8),

longitude DECIMAL(11, 8),

altitude DECIMAL(8, 2),

precision\_gps DECIMAL(8, 2),

-- Métadonnées de collecte

external\_id VARCHAR(100),

uuid VARCHAR(100) UNIQUE,

submission\_time TIMESTAMP NULL,

validation\_status ENUM('draft', 'submitted', 'validated', 'rejected') DEFAULT 'draft',

notes TEXT,

status\_collecte VARCHAR(50),

submitted\_by VARCHAR(100),

version\_form VARCHAR(20),

tags TEXT,

index\_ordre INT,

-- Gestion interne

statut\_traitement ENUM('nouveau', 'en\_cours', 'traite', 'valide') DEFAULT 'nouveau',

source\_donnee ENUM('import', 'saisie\_manuelle') NOT NULL,

utilisateur\_id INT NOT NULL,

date\_creation TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

date\_traitement TIMESTAMP NULL,

traite\_par\_utilisateur\_id INT NULL,

observations\_internes TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

-- Clés étrangères

FOREIGN KEY (utilisateur\_id) REFERENCES utilisateurs(id) ON DELETE CASCADE,

FOREIGN KEY (traite\_par\_utilisateur\_id) REFERENCES utilisateurs(id) ON DELETE SET NULL,

-- Index pour optimisation

INDEX idx\_statut\_traitement (statut\_traitement),

INDEX idx\_commune (commune),

INDEX idx\_type\_infrastructure (type\_infrastructure),

INDEX idx\_etat\_fonctionnement (etat\_fonctionnement),

INDEX idx\_utilisateur (utilisateur\_id),

INDEX idx\_date\_creation (date\_creation),

INDEX idx\_geolocalisation (latitude, longitude),

INDEX idx\_uuid (uuid),

INDEX idx\_validation\_status (validation\_status)

);

-- Table des informations complémentaires

CREATE TABLE informations\_complementaires (

id INT AUTO\_INCREMENT PRIMARY KEY,

infrastructure\_id INT NOT NULL,

type\_information VARCHAR(100) NOT NULL,

libelle VARCHAR(200) NOT NULL,

valeur TEXT,

date\_ajout TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

ajoute\_par\_utilisateur\_id INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (infrastructure\_id) REFERENCES infrastructures(id) ON DELETE CASCADE,

FOREIGN KEY (ajoute\_par\_utilisateur\_id) REFERENCES utilisateurs(id) ON DELETE SET NULL,

INDEX idx\_infrastructure (infrastructure\_id),

INDEX idx\_type\_information (type\_information)

);

-- Table des importations de fichiers

CREATE TABLE importations (

id INT AUTO\_INCREMENT PRIMARY KEY,

nom\_fichier VARCHAR(255) NOT NULL,

taille\_fichier INT,

chemin\_fichier VARCHAR(500),

nombre\_lignes\_total INT DEFAULT 0,

nombre\_lignes\_importees INT DEFAULT 0,

nombre\_erreurs INT DEFAULT 0,

statut ENUM('en\_cours', 'termine', 'erreur') DEFAULT 'en\_cours',

utilisateur\_id INT NOT NULL,

date\_importation TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

date\_fin\_importation TIMESTAMP NULL,

rapport\_erreurs JSON,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (utilisateur\_id) REFERENCES utilisateurs(id) ON DELETE CASCADE,

INDEX idx\_utilisateur (utilisateur\_id),

INDEX idx\_date\_importation (date\_importation),

INDEX idx\_statut (statut)

);

-- Table des logs d'activité

CREATE TABLE logs\_activite (

id INT AUTO\_INCREMENT PRIMARY KEY,

utilisateur\_id INT,

action VARCHAR(100) NOT NULL,

description TEXT,

table\_concernee VARCHAR(100),

id\_enregistrement INT,

donnees\_avant JSON,

donnees\_apres JSON,

adresse\_ip VARCHAR(45),

user\_agent TEXT,

date\_action TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (utilisateur\_id) REFERENCES utilisateurs(id) ON DELETE SET NULL,

INDEX idx\_utilisateur (utilisateur\_id),

INDEX idx\_action (action),

INDEX idx\_date\_action (date\_action),

INDEX idx\_table\_concernee (table\_concernee)

);

-- Table pour les sessions utilisateurs (pour JWT refresh tokens)

CREATE TABLE sessions\_utilisateur (

id INT AUTO\_INCREMENT PRIMARY KEY,

utilisateur\_id INT NOT NULL,

refresh\_token VARCHAR(500) NOT NULL,

expire\_le TIMESTAMP NOT NULL,

adresse\_ip VARCHAR(45),

user\_agent TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (utilisateur\_id) REFERENCES utilisateurs(id) ON DELETE CASCADE,

INDEX idx\_utilisateur (utilisateur\_id),

INDEX idx\_refresh\_token (refresh\_token(255)),

INDEX idx\_expire\_le (expire\_le)

);

-- Table des documents générés (PDF)

CREATE TABLE documents\_generes (

id INT AUTO\_INCREMENT PRIMARY KEY,

nom\_document VARCHAR(255) NOT NULL,

type\_document VARCHAR(100),

chemin\_fichier VARCHAR(500),

taille\_fichier INT,

nombre\_enregistrements INT,

utilisateur\_id INT NOT NULL,

filtre\_applique JSON,

date\_generation TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

expire\_le TIMESTAMP,

statut ENUM('genere', 'expire', 'supprime') DEFAULT 'genere',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (utilisateur\_id) REFERENCES utilisateurs(id) ON DELETE CASCADE,

INDEX idx\_utilisateur (utilisateur\_id),

INDEX idx\_date\_generation (date\_generation),

INDEX idx\_statut (statut)

);

-- Insertion des départements du Bénin

INSERT INTO departements (nom, code) VALUES

('Alibori', 'AL'),

('Atacora', 'AT'),

('Atlantique', 'AQ'),

('Borgou', 'BO'),

('Collines', 'CO'),

('Couffo', 'CF'),

('Donga', 'DO'),

('Littoral', 'LI'),

('Mono', 'MO'),

('Ouémé', 'OU'),

('Plateau', 'PL'),

('Zou', 'ZO');

-- Insertion des communes (focus sur celles mentionnées dans les données)

INSERT INTO communes (nom, departement\_id) VALUES

-- Borgou (département principal mentionné)

('Parakou', 4),

('Tchaourou', 4),

('N''Dali', 4),

('Nikki', 4),

('Bembèrèké', 4),

('Kalalé', 4),

('Sinendé', 4),

('Pèrèrè', 4),

-- Autres communes importantes

-- Littoral

('Cotonou', 8),

-- Atlantique

('Abomey-Calavi', 3),

('Allada', 3),

('Ouidah', 3),

('Sô-Ava', 3),

-- Ouémé

('Porto-Novo', 10),

('Sèmè-Kpodji', 10),

('Adjarra', 10),

-- Zou

('Abomey', 12),

('Bohicon', 12),

('Covè', 12),

-- Mono

('Lokossa', 9),

('Comè', 9),

-- Atacora

('Natitingou', 2),

('Tanguiéta', 2),

-- Alibori

('Kandi', 1),

('Malanville', 1),

-- Donga

('Djougou', 7),

('Bassila', 7),

-- Couffo

('Aplahoué', 6),

('Dogbo', 6),

-- Collines

('Savalou', 5),

('Bantè', 5),

-- Plateau

('Pobè', 11),

('Kétou', 11);

-- Création d'un utilisateur administrateur par défaut

INSERT INTO utilisateurs (nom, prenom, email, mot\_de\_passe, role, departement\_id, commune\_id)

VALUES ('Admin', 'ADECOB', 'admin@adecob.org', '$2b$10$example\_hashed\_password', 'administrateur', 4, 1);

-- Vues utiles pour les requêtes fréquentes

-- Vue pour les utilisateurs avec leurs départements et communes

CREATE VIEW vue\_utilisateurs\_complet AS

SELECT

u.id,

u.nom,

u.prenom,

u.email,

u.telephone,

u.role,

u.statut,

d.nom as departement,

c.nom as commune,

u.date\_inscription,

u.derniere\_connexion

FROM utilisateurs u

LEFT JOIN departements d ON u.departement\_id = d.id

LEFT JOIN communes c ON u.commune\_id = c.id;

-- Vue pour les infrastructures avec localisation complète

CREATE VIEW vue\_infrastructures\_complete AS

SELECT

i.id,

i.nom\_infrastructure,

i.type\_infrastructure,

i.commune,

i.village\_quartier,

i.etat\_fonctionnement,

i.niveau\_degradation,

i.annee\_realisation,

i.bailleur,

i.statut\_traitement,

i.source\_donnee,

i.date\_creation,

i.latitude,

i.longitude,

u.nom as createur\_nom,

u.prenom as createur\_prenom,

ut.nom as traiteur\_nom,

ut.prenom as traiteur\_prenom

FROM infrastructures i

LEFT JOIN utilisateurs u ON i.utilisateur\_id = u.id

LEFT JOIN utilisateurs ut ON i.traite\_par\_utilisateur\_id = ut.id;

-- Vue pour les statistiques par commune

CREATE VIEW vue\_stats\_par\_commune AS

SELECT

commune,

COUNT(\*) as total\_infrastructures,

COUNT(CASE WHEN etat\_fonctionnement = 'Bon' THEN 1 END) as bon\_etat,

COUNT(CASE WHEN etat\_fonctionnement = 'Mauvais' THEN 1 END) as mauvais\_etat,

COUNT(CASE WHEN etat\_fonctionnement = 'Hors service' THEN 1 END) as hors\_service,

COUNT(CASE WHEN besoin\_rehabilitation = TRUE THEN 1 END) as besoin\_rehabilitation,

COUNT(CASE WHEN statut\_traitement = 'nouveau' THEN 1 END) as non\_traites

FROM infrastructures

GROUP BY commune;