### **Schéma de Base de Données Détaillé - Africage**

### **1. Introduction**

Ce document décrit la structure de la base de données pour le projet Africage. Le schéma est conçu pour être robuste, évolutif et garantir l'intégrité des données. Nous proposons l'utilisation d'une base de données de type PostgreSQL, qui offre des fonctionnalités avancées comme la gestion de types de données JSONB et des capacités géospatiales robustes.

Les conventions de nommage suivantes sont utilisées :

* Noms de tables : pluriel, en snake\_case (ex: users, parcels).
* Noms de colonnes : snake\_case (ex: first\_name, created\_at).

### **2. Diagramme Entité-Relation (Description Textuelle)**

* **Users ↔ Parcels** : Un utilisateur (user) peut envoyer plusieurs colis (parcels) (relation *One-to-Many*). Un utilisateur peut aussi transporter plusieurs colis (relation *One-to-Many*).
* **Users ↔ Trips** : Un utilisateur (user) peut proposer plusieurs trajets (trips) (relation *One-to-Many*).
* **Trips ↔ Parcels** : Un trajet (trip) peut inclure plusieurs colis (parcels) (relation *One-to-Many*). Un colis est associé à un seul trajet.
* **Users ↔ KYC\_Documents** : Un utilisateur (user) a un seul dossier de vérification (kyc\_document) (relation *One-to-One*).
* **Users ↔ Ratings** : Un utilisateur (user) peut donner et recevoir plusieurs évaluations (ratings). La relation est liée à un colis spécifique pour s'assurer que l'évaluation concerne une transaction réelle.
* **Users ↔ Messages** : Les utilisateurs peuvent s'échanger des messages (messages). La table messages lie un expéditeur et un destinataire.

### **3. Description Détaillée des Tables**

#### **Table: users**

Stocke les informations de base de tous les utilisateurs de la plateforme.

| **Nom de la colonne** | **Type de données** | **Contraintes** | **Description** |
| --- | --- | --- | --- |
| id | UUID | PRIMARY KEY | Identifiant unique de l'utilisateur (PK). |
| first\_name | VARCHAR(100) | NOT NULL | Prénom de l'utilisateur. |
| last\_name | VARCHAR(100) | NOT NULL | Nom de famille de l'utilisateur. |
| email | VARCHAR(255) | UNIQUE, NOT NULL | Adresse email unique, utilisée pour la connexion. |
| phone\_number | VARCHAR(30) | UNIQUE, NOT NULL | Numéro de téléphone unique. |
| password\_hash | VARCHAR(255) | NOT NULL | Hash sécurisé du mot de passe de l'utilisateur. |
| profile\_picture\_url | TEXT |  | URL de la photo de profil. |
| kyc\_status | ENUM | DEFAULT 'PENDING' | Statut de la vérification KYC ('PENDING', 'VERIFIED', 'REJECTED'). |
| created\_at | TIMESTAMPTZ | NOT NULL | Date et heure de création du compte. |
| updated\_at | TIMESTAMPTZ | NOT NULL | Date et heure de la dernière mise à jour. |

#### **Table: kyc\_documents**

Stocke les informations relatives à la vérification d'identité (Know Your Customer).

| **Nom de la colonne** | **Type de données** | **Contraintes** | **Description** |
| --- | --- | --- | --- |
| id | UUID | PRIMARY KEY | Identifiant unique du document (PK). |
| user\_id | UUID | FOREIGN KEY (users.id), UNIQUE | Lien vers l'utilisateur (relation 1-to-1). |
| document\_type | ENUM | NOT NULL | Type de document ('ID\_CARD', 'PASSPORT', 'DRIVER\_LICENSE'). |
| document\_front\_url | TEXT | NOT NULL | URL de l'image du recto du document. |
| document\_back\_url | TEXT |  | URL de l'image du verso (si applicable). |
| status | ENUM | DEFAULT 'SUBMITTED' | Statut de la vérification ('SUBMITTED', 'APPROVED', 'REJECTED'). |
| submitted\_at | TIMESTAMPTZ | NOT NULL | Date de soumission. |
| reviewed\_at | TIMESTAMPTZ |  | Date de la revue par un administrateur. |

#### **Table: parcels**

Contient toutes les informations sur les colis proposés à la livraison.

| **Nom de la colonne** | **Type de données** | **Contraintes** | **Description** |
| --- | --- | --- | --- |
| id | UUID | PRIMARY KEY | Identifiant unique du colis (PK). |
| sender\_id | UUID | FOREIGN KEY (users.id) | ID de l'utilisateur qui envoie le colis. |
| traveler\_id | UUID | FOREIGN KEY (users.id) | ID de l'utilisateur qui transporte le colis (NULL si non assigné). |
| trip\_id | UUID | FOREIGN KEY (trips.id) | ID du trajet associé (NULL si non assigné). |
| origin | JSONB | NOT NULL | Objet JSON décrivant l'origine ({"city": "...", "country": "..."}). |
| destination | JSONB | NOT NULL | Objet JSON décrivant la destination. |
| description | TEXT | NOT NULL | Description du contenu du colis. |
| size | ENUM | NOT NULL | Taille du colis ('SMALL', 'MEDIUM', 'LARGE'). |
| weight\_kg | DECIMAL(5, 2) | NOT NULL | Poids du colis en kilogrammes. |
| photo\_url | TEXT |  | URL d'une photo du colis. |
| proposed\_price | DECIMAL(10, 2) | NOT NULL | Prix proposé par l'expéditeur pour le transport. |
| currency | VARCHAR(3) | NOT NULL | Devise du prix (ex: 'XOF', 'EUR'). |
| status | ENUM | DEFAULT 'PENDING\_ACCEPTANCE' | Statut du colis ('PENDING\_ACCEPTANCE', 'ACCEPTED', 'IN\_TRANSIT', 'DELIVERED', 'CANCELLED'). |
| delivery\_code | VARCHAR(10) |  | Code secret que le destinataire donne au voyageur pour confirmer la livraison. |
| tracking\_code | VARCHAR(20) | UNIQUE, NOT NULL | Code de suivi unique généré pour le colis. |
| created\_at | TIMESTAMPTZ | NOT NULL | Date de création de l'annonce. |
| updated\_at | TIMESTAMPTZ | NOT NULL | Date de la dernière mise à jour du statut. |

#### **Table: trips**

Contient les annonces de trajets des voyageurs.

| **Nom de la colonne** | **Type de données** | **Contraintes** | **Description** |
| --- | --- | --- | --- |
| id | UUID | PRIMARY KEY | Identifiant unique du trajet (PK). |
| traveler\_id | UUID | FOREIGN KEY (users.id) | ID de l'utilisateur qui voyage. |
| origin | JSONB | NOT NULL | Objet JSON décrivant l'origine du trajet. |
| destination | JSONB | NOT NULL | Objet JSON décrivant la destination du trajet. |
| departure\_datetime | TIMESTAMPTZ | NOT NULL | Date et heure de départ. |
| arrival\_datetime | TIMESTAMPTZ | NOT NULL | Date et heure d'arrivée estimée. |
| available\_space | ENUM | NOT NULL | Espace disponible ('SMALL', 'MEDIUM', 'LARGE'). |
| status | ENUM | DEFAULT 'ACTIVE' | Statut du trajet ('ACTIVE', 'COMPLETED', 'CANCELLED'). |
| created\_at | TIMESTAMPTZ | NOT NULL | Date de création de l'annonce. |

#### **Table: messages**

Stocke les messages échangés entre les utilisateurs.

| **Nom de la colonne** | **Type de données** | **Contraintes** | **Description** |
| --- | --- | --- | --- |
| id | UUID | PRIMARY KEY | Identifiant unique du message (PK). |
| sender\_id | UUID | FOREIGN KEY (users.id) | ID de l'expéditeur. |
| receiver\_id | UUID | FOREIGN KEY (users.id) | ID du destinataire. |
| parcel\_id | UUID | FOREIGN KEY (parcels.id) | Contexte du message (le colis concerné). |
| content | TEXT | NOT NULL | Contenu du message. |
| sent\_at | TIMESTAMPTZ | NOT NULL | Date d'envoi. |
| read\_at | TIMESTAMPTZ |  | Date de lecture (NULL si non lu). |

### **4. Stratégie d'Indexation**

Pour garantir des performances optimales, les index suivants seront créés :

* **Index sur les clés étrangères (Foreign Keys)** : Toutes les colonnes \*\_id (ex: parcels.sender\_id, messages.receiver\_id) seront automatiquement indexées pour accélérer les jointures.
* **Index sur les champs de recherche courants** :
  + users(email) et users(phone\_number) pour des connexions rapides.
  + parcels(status) pour filtrer rapidement les colis par leur statut.
  + parcels(tracking\_code) pour la recherche de suivi.
* **Index composites** :
  + messages(sender\_id, receiver\_id, sent\_at) pour charger efficacement les conversations.
* **Index sur JSONB** : Des index GIN peuvent être créés sur les champs origin et destination pour accélérer les recherches basées sur la ville ou le pays.