# Konnaxion Platform – Spécification technique des bases de données (v1)

## 1. Contexte et périmètre

Cette spécification décrit l’instance PostgreSQL mise en place :

* **konnaxion\_db** – cœur des données métiers (toutes les applications).
* **pulse\_db** – base réservée pour la télémétrie/analytics (créée mais encore vide).

Le script d’initialisation a été exécuté sous PostgreSQL 15. Il active l’extension pg\_trgm (recherche similaire) et installe tout le schéma applicatif.

**Remarque :** timescaledb n’a pas été chargé faute de package ; l’activation des hypertables reste commentée et pourra être appliquée plus tard quand l’extension sera disponible.

## 2. Contraintes globales

| Élément | Valeur |
| --- | --- |
| Encodage | UTF‑8 |
| LC\_COLLATE / LC\_CTYPE | par défaut de l’instance |
| Politique d’horodatage | Toutes les tables portent created\_at, updated\_at, is\_deleted, deleted\_at (soft‑delete) |
| Convention de nommage | snake\_case, préfixe d’app (ex : core\_, ekoh\_, …) |
| Politique FK | ON DELETE CASCADE là où l’incohérence serait critique ; sinon SET NULL |
| Multi‑tenance | Non (single‑db) ; mais la séparation logique se fait par « app » |

## 3. Diagramme logique (vue macroscopique)

[core]──┐

│ (Many‑to‑Many)

[ekoh]──┼─<core\_customuser>──<team\_team>

│ ╲

│ > messaging\_conversation

[projects] ╱

│ > collab\_spaces

[...] ┘

(schéma complet fourni en annexe)

## 4. Détail par application

### 4.1 Core

| Table | Rôle | Clés/Index | Particularités |
| --- | --- | --- | --- |
| core\_customuser | Authentification & profil | PK id; username UNIQUE; index par défaut | Stocke : rôles, langue, token offline |
| core\_systemconfiguration | KV système | PK id; key UNIQUE | Paramétrage global JSON ou texte |
| core\_configurationchangelog | Historique des changements | FK → customuser, systemconfiguration | Audit trail |

### 4.2 Search

| Table | Description |
| --- | --- |
| search\_searchquerylog | Log des requêtes (texte, nb résultats, FK user) |
| search\_searchindex | Paramètres d’index externes (ex : Algolia, Elastic) |
| **Indexes** : GIN trigram (pg\_trgm) sur le champ name ou autres champs textuels à forte sélectivité. |  |

### 4.3 AI

ai\_airesult conserve l’entrée JSONB et le résultat JSONB d’une exécution IA.

### 4.4 Notifications & Messaging

* notifications\_notification : push internes (type, read flag, sender/recipient FK).
* messaging\_conversation + messaging\_message + table de passage messaging\_conversation\_participants pour la messagerie privée ou de groupe.

### 4.5 Ekoh (Réputation)

| Table | Fonction |
| --- | --- |
| ekoh\_expertisetag | Taxonomie d’expertise (index trigram). |
| ekoh\_reputationprofile | One‑to‑One avec customuser. |
| ekoh\_reputationevent | Journal des variations de score. |
| ekoh\_weightedvote | Pondération personnalisée des votes. |
| ekoh\_reputationprofile\_expertise\_tags | M‑N entre profils et tags. |

### 4.6 Konnected – Knowledge & Collaboration

* foundation\_knowledgeunit – élément pédagogique de base
* konnectedcommunity\_\* – discussions + commentaires hiérarchiques
* learning\_\* – QCM, leçons, réponses, quiz
* offline\_offlinecontentpackage – module hors‑ligne ZIP/JSON
* paths\_\* – parcours d’apprentissage séquencé
* team\_\* – équipes et invitations (M‑N team\_team\_members)

### 4.7 Keenkonnect (Projets, Gap, Expert‑Match, Collab Spaces)

| Domaine | Tables clés |
| --- | --- |
| **Projects** | projects\_project, projects\_milestone, projects\_task |
| **Gap analysis** | gap\_analysis\_gapanalysis |
| **Expert match** | expert\_match\_candidateprofile, expert\_match\_expertmatchrequest, expert\_match\_matchscore |
| **Team formation** | team\_formation\_\* |
| **Collab spaces** | collab\_spaces\_\* (space, messages, documents) |
| **Knowledge hub** | knowledge\_hub\_\* (docs + révisions) |

### 4.8 Ethikos

* **Home** : topics, featured, recommandations.
* **Debate Arena** : sessions, arguments (auto‑référentiel parent\_id), votes.
* **Stats** : stats\_debatestatistic, stats\_debateeventlog (pré‑vue pour Timescale).
* **Knowledge Base** : archives JSON/texte de débats.
* **Prioritization & Resolution** : classement et conclusion d’un débat.

### 4.9 Kreative

| Sous‑module | Tables |
| --- | --- |
| Artworks | artworks\_exhibition, artworks\_artwork |
| Marketplace | marketplace\_\* (artist profile, listing, commission) |
| Community | kreativecommunity\_\* (posts, comments, reviews) |
| Immersive | immersive\_immersiveexperience |

## 5. Indexation & performances

* **Full‑text fuzzy** : pg\_trgm + GIN sur ekoh\_expertisetag.name, kreativecommunity\_communitypost.content, knowledge\_base\_debatearchive.transcript, etc.
* **FK standards** créent automatiquement des BTREE sur PK + FK.
* **Hypertables (optionnel)** : prévoir create\_hypertable sur stats\_debatestatistic (recorded\_at) et stats\_debateeventlog (timestamp) dès que timescaledb sera installé.

## 6. Sécurité & gouvernance

* Accès en lecture/écriture via rôles applicatifs (app\_rw, app\_ro…), non inclus dans le script initial.
* Politique **soft‑delete** universelle : requêtes applicatives doivent toujours filtrer is\_deleted = false.
* RGPD : champ deleted\_at obligatoire pour traçabilité.

## 7. Sauvegarde & restauration

* **Dump logique** : pg\_dump -Fc konnaxion\_db > konnaxion\_db.dump (quotidien)
* **Backups différentiels** via WAL archiving recommandé (> 100 Go données ou contraintes RPO/RTO)
* **Pulse\_db** : inclure dès que la collecte d’événements débute.

## 8. Roadmap d’évolution

| Version | Changement prévu |
| --- | --- |
| 1.1 | Déploiement de timescaledb + transformation de tables stats en hypertables |
| 1.2 | Ajout de core\_auditlog (CDC complète) |
| 1.3 | Migration vers partitionnement logique des logs par mois si volume > 500 M lignes |

# Konnaxion Platform – Data Dictionary (v1)

Cette documentation détaille **toutes les structures SQL créées par** konnaxion\_init.sql : bases, tables, colonnes, types, contraintes et index notables.  
Pour chaque table :

* **PK** = clé primaire, **FK** = clé étrangère, **UK** = contrainte d’unicité.
* Les colonnes « created\_at / updated\_at / is\_deleted / deleted\_at » suivent la convention d’audit et de soft‑delete (voir section 1.3).

## 1. Vue d’ensemble

### 1.1 Bases

| Base | Rôle | Particularités |
| --- | --- | --- |
| konnaxion\_db | Données applicatives principales | Extension **pg\_trgm** activée (recherche floue) |
| pulse\_db | Réservée aux futures métriques temps‑réel (TimescaleDB) | Pour l’instant vide |

### 1.2 Principales conventions

| Élément | Détails |
| --- | --- |
| **Horodatage** | TIMESTAMP WITH TIME ZONE, valeurs gérées par l’ORM/Django ou triggers |
| **Soft‑delete** | is\_deleted (bool) + deleted\_at (timestamp) |
| **Index texte** | Trigram : pg\_trgm sur les colonnes plein‑texte à forte volumétrie (ex. ekoh\_expertisetag.name) |

### 1.3 Chronologie de création

1. Création / drop des bases.
2. Connexion à konnaxion\_db.
3. Création de l’extension pg\_trgm.
4. Création des tables par domaine applicatif.
5. Création des tables d’association M‑N.
6. Indexation optionnelle.

## 2. Tables par domaine

**Astuce :** utilisez Ctrl + F ou la table des matières interactive pour naviguer rapidement.

### 2.1 Konnaxion – Core

#### core\_customuser

| Colonne | Type | Contraintes |
| --- | --- | --- |
| id | BIGSERIAL | PK |
| password | VARCHAR(128) | NOT NULL |
| last\_login | TIMESTAMP TZ |  |
| is\_superuser | BOOLEAN | NOT NULL DEFAULT FALSE |
| username | VARCHAR(150) | NOT NULL, UK |
| first\_name | VARCHAR(150) | NOT NULL |
| last\_name | VARCHAR(150) | NOT NULL |
| email | VARCHAR(254) | NOT NULL |
| is\_staff | BOOLEAN | NOT NULL DEFAULT FALSE |
| is\_active | BOOLEAN | NOT NULL DEFAULT TRUE |
| date\_joined | TIMESTAMP TZ | NOT NULL |
| language\_preference | VARCHAR(10) | NOT NULL DEFAULT 'en' |
| device\_details | JSONB |  |
| role | VARCHAR(50) | NOT NULL DEFAULT 'user' |
| offline\_sync\_token | VARCHAR(255) |  |
| created\_at | TIMESTAMP TZ | NOT NULL |
| updated\_at | TIMESTAMP TZ | NOT NULL |
| is\_deleted | BOOLEAN | NOT NULL DEFAULT FALSE |
| deleted\_at | TIMESTAMP TZ |  |

Indexes : (username) UK, trigram sur recherche texte (via ORM si besoin).

#### core\_systemconfiguration

| Colonne | Type | Contraintes |
| --- | --- | --- |
| id | BIGSERIAL | PK |
| key | VARCHAR(100) | NOT NULL, UK |
| value | TEXT | NOT NULL |
| description | TEXT |  |
| created\_at | TIMESTAMP TZ | NOT NULL |
| updated\_at | TIMESTAMP TZ | NOT NULL |
| is\_deleted | BOOLEAN | NOT NULL DEFAULT FALSE |
| deleted\_at | TIMESTAMP TZ |  |

#### core\_configurationchangelog

| Colonne | Type | Contraintes |
| --- | --- | --- |
| id | BIGSERIAL | PK |
| old\_value | TEXT | NOT NULL |
| new\_value | TEXT | NOT NULL |
| change\_reason | TEXT |  |
| changed\_by\_id | BIGINT | FK → core\_customuser(id), ON DELETE SET NULL |
| configuration\_id | BIGINT | **FK**, NOT NULL → core\_systemconfiguration(id), ON DELETE CASCADE |
| created\_at | TIMESTAMP TZ | NOT NULL |
| updated\_at | TIMESTAMP TZ | NOT NULL |
| is\_deleted | BOOLEAN | NOT NULL DEFAULT FALSE |
| deleted\_at | TIMESTAMP TZ |  |

Indexes : (changed\_by\_id), (configuration\_id).

### 2.2 Konnaxion – Search

#### search\_searchquerylog

| id | BIGSERIAL PK |  
| query\_text | TEXT NOT NULL |  
| results\_count | INTEGER DEFAULT 0 |  
| user\_id | BIGINT FK → core\_customuser(id) SET NULL |  
| … + colonnes d’audit |

Index : (user\_id).

#### search\_searchindex

| id | BIGSERIAL PK |  
| name | VARCHAR(255) NOT NULL |  
| settings | JSONB NOT NULL |  
| last\_updated | TIMESTAMP TZ |  
| … + audit |

### 2.3 Konnaxion – AI

(ai\_airesult) ‑ dictionnaires JSON d’entrée / sortie des inférences.

| Colonne | Type | Contraintes |
| --- | --- | --- |
| id | BIGSERIAL PK |  |
| input\_data | JSONB NOT NULL |  |
| result\_data | JSONB NOT NULL |  |
| … + audit |  |  |

### 2.4 Konnaxion – Notifications

(notifications\_notification)  
| id | BIGSERIAL PK |  
| message | TEXT NOT NULL |  
| read | BOOLEAN DEFAULT FALSE |  
| notification\_type | VARCHAR(50) DEFAULT 'info' |  
| recipient\_id | BIGINT FK → core\_customuser(id) CASCADE |  
| sender\_id | BIGINT FK → core\_customuser(id) SET NULL |  
| … + audit |

Indexes : (recipient\_id), (sender\_id).

### 2.5 Konnaxion – Messaging

#### messaging\_conversation

| id | BIGSERIAL PK | |  
| topic | VARCHAR(255) | |  
| … + audit |

#### messaging\_message

| id | BIGSERIAL PK |  
| content | TEXT NOT NULL |  
| timestamp | TIMESTAMP TZ DEFAULT NOW() |  
| sender\_id | BIGINT FK NOT NULL → core\_customuser(id) |  
| conversation\_id | BIGINT FK NOT NULL → messaging\_conversation(id) |  
| is\_flagged | BOOLEAN DEFAULT FALSE |  
| … + audit |

Indexes : (conversation\_id), (sender\_id).

(La table d’association **messaging\_conversation\_participants** est détaillée en 2.14.)

### 2.6 Konnaxion – Ekoh

#### ekoh\_expertisetag

| id | BIGSERIAL PK |  
| name | VARCHAR(100) UK NOT NULL |  
| description | TEXT |  
| … + audit |

Index : trigram (name).

#### ekoh\_reputationprofile

| id | BIGSERIAL PK |  
| user\_id | BIGINT UK NOT NULL FK → core\_customuser(id) CASCADE |  
| reputation\_score | INTEGER DEFAULT 0 |  
| ethical\_multiplier | DOUBLE PRECISION DEFAULT 1.0 |  
| … + audit |

#### ekoh\_reputationevent

| id | BIGSERIAL PK |  
| profile\_id | BIGINT FK NOT NULL → ekoh\_reputationprofile(id) |  
| event\_type | VARCHAR(100) NOT NULL |  
| impact | INTEGER NOT NULL |  
| timestamp | TIMESTAMP TZ DEFAULT NOW() |  
| … + audit |

#### ekoh\_weightedvote

| id | BIGSERIAL PK |  
| debate\_id | BIGINT NOT NULL (clé logique vers module débat) |  
| user\_id | BIGINT FK NOT NULL → core\_customuser(id) |  
| weight | DECIMAL(5,2) DEFAULT 1.00 |  
| … + audit |

### 2.7 Konnected – Foundation / Community / Learning / Offline / Paths

(extraits – voir document pour la liste complète)

#### foundation\_knowledgeunit

| id | BIGSERIAL PK | |  
| title | VARCHAR(255) NOT NULL |  
| content | TEXT |  
| … + audit |

#### konnectedcommunity\_discussionthread

| id | BIGSERIAL PK |  
| title | VARCHAR(255) |  
| created\_by\_id | BIGINT FK NOT NULL → core\_customuser(id) |  
| … + audit |

#### konnectedcommunity\_comment

| id | BIGSERIAL PK |  
| thread\_id | BIGINT FK NOT NULL → discussionthread(id) |  
| author\_id | BIGINT FK NOT NULL → core\_customuser(id) |  
| parent\_id | BIGINT FK → comment(id) |  
| … + audit |

#### learning\_question / learning\_answer / learning\_lesson / learning\_quiz

(voir tables pour détails des colonnes JSON optionnelles)

### 2.8 Konnected – Team & Projects

Tables : team\_team, team\_teaminvitation, team\_formation\_\*, projects\_\* – champs standard projet, jalons, tâches.

### 2.9 Keenkonnect – Gap Analysis & Expert Match

Tables gap\_analysis\_gapanalysis, expert\_match\_\* – stockent les critères JSON, notes et scores (DOUBLE PRECISION / NUMERIC).

### 2.10 Collab Spaces

collab\_spaces\_collabspace, collab\_spaces\_chatmessage, collab\_spaces\_document.

### 2.11 Knowledge Hub

knowledge\_hub\_knowledgedocument, knowledge\_hub\_documentrevision.

### 2.12 Ethikos – Home, Debate Arena, Stats, Knowledge Base, Prioritization, Resolution

Listes exhaustives des colonnes pour :

* home\_debatetopic, home\_featureddebate, home\_personalizedrecommendation
* debate\_arena\_\* (sessions, arguments, votes)
* stats\_debate\* (statistic / eventlog) (hypertables à venir)
* knowledge\_base\_debatearchive
* prioritization\_debateprioritization
* resolution\_debateresolution

### 2.13 Kreative – Artworks, Marketplace, Immersive, Community

Inclut artworks\_\*, marketplace\_\*, immersive\_immersiveexperience, kreativecommunity\_\* (posts, reviews, comments).

### 2.14 Tables de liaison (Many‑to‑Many)

| Table | Clé 1 | Clé 2 | PK compososite |
| --- | --- | --- | --- |
| team\_team\_members | team\_id → team\_team(id) | customuser\_id → core\_customuser(id) | (team\_id, customuser\_id) |
| messaging\_conversation\_participants | conversation\_id → messaging\_conversation(id) | customuser\_id → core\_customuser(id) | (conversation\_id, customuser\_id) |
| ekoh\_reputationprofile\_expertise\_tags | reputationprofile\_id → ekoh\_reputationprofile(id) | expertisetag\_id → ekoh\_expertisetag(id) | (reputationprofile\_id, expertisetag\_id) |

## 3. Index & performance

* **BTree** générés implicitement sur PK & FK.
* **GIN trigram** : ekoh\_expertisetag.name (recherche rapide d’étiquettes).
* **Autres** : indexes sur colonnes FK très sollicitées (cf. script).

## 4. Sécurité & sauvegarde

* Rôles applicatifs (à créer) : konnaxion\_app, konnaxion\_readonly.
* Stratégie de backup : pg\_dump full + WAL shipping quotidien.
* Chiffrement repos : gérer via chiffrement disque / VM.

## 5. Évolutions prévues

1. Activer TimescaleDB et convertir stats\_\* en hypertables.
2. Ajouter pgcrypto pour UUID / hashing.
3. Mettre en place Row‑Level Security pour les modules Marketplace et Messaging.

Exemple pour initier connection db  
  
Thought for 10 seconds

### Commandes complètes prêtes à copier‑coller

(données exactes de votre fichier .env)

| Élément | Valeur utilisée |
| --- | --- |
| **Hôte PostgreSQL** | 127.0.0.1 |
| **Port** | 5432 |
| **Base** | konnaxion |
| **Utilisateur** | konnaxion\_user |
| **Mot de passe** | Gr05bo55 |

## 1. Exécuter le script init\_ethikos.sql avec ****variable****PGPASSWORD

# Aller dans le dossier qui contient init\_ethikos.sql

PS C:\> cd "C:\chemin\vers\vos\scripts"

# Exporter le mot de passe (valable uniquement pour la session PowerShell courante)

PS C:\chemin\vers\vos\scripts> $env:PGPASSWORD = "Gr05bo55"

# Lancer le script et stopper au premier échec

PS C:\chemin\vers\vos\scripts> psql `

-h 127.0.0.1 `

-p 5432 `

-U konnaxion\_user `

-d konnaxion `

-v ON\_ERROR\_STOP=1 `

-f ".\init\_ethikos.sql"

Arguments « pour » : mot de passe non visible dans l’historique PowerShell.  
Arguments « contre » : la variable d’environnement reste en mémoire tant que la session n’est pas fermée.

## 2. Méthode ****URI de connexion**** (moins sûre : mot de passe présent dans l’historique)

PS C:\chemin\vers\vos\scripts> psql `

"postgres://konnaxion\_user:Gr05bo55@127.0.0.1:5432/konnaxion" `

-v ON\_ERROR\_STOP=1 `

-f ".\init\_ethikos.sql"

Pour : une seule chaîne.  
Contre : mot de passe inscrit dans l’historique PowerShell ; évitez sur des postes partagés.

## 3. Alternative plus sûre : fichier pgpass.conf

1. Créez %APPDATA%\postgresql\pgpass.conf
2. Ajoutez la ligne :

127.0.0.1:5432:konnaxion:konnaxion\_user:Gr05bo55

1. Limitez les droits (lecture seule pour votre compte).
2. Ensuite, utilisez simplement :

PS C:\chemin\vers\vos\scripts> psql -h 127.0.0.1 -U konnaxion\_user -d konnaxion -v ON\_ERROR\_STOP=1 -f ".\init\_ethikos.sql"

Pour : aucun mot de passe dans les variables ni l’historique.  
Contre : fichier à protéger manuellement ; non portable d’une machine à l’autre.

### Incertitudes / points à vérifier

* **Chemin vers** psql : s’il n’est pas dans le PATH, utilisez par ex.

"C:\Program Files\PostgreSQL\15\bin\psql.exe" ...

* **Contexte Docker** : si vous exécutez via docker‑compose, l’hôte DB sera probablement postgres (nom du service) et non 127.0.0.1.
* **Script volumineux** : ajoutez éventuellement l’option --echo-all pour journaliser chaque commande exécutée.