

SCHÉMA DE LA BASE DE DONNÉES

Plateforme de Certification Numérique des Documents Académiques

Version : 1.0

Date : Novembre 2025

SGBD : PostgreSQL (recommandé)

INTRODUCTION

Cette base de données supporte une plateforme de certification numérique qui permet : - Aux **administrations** (universités, écoles, organismes) d'émettre des documents certifiés - Aux **étudiants** de gérer leur profil et recevoir des recommandations d'emploi - Aux **recruteurs** de publier des offres et identifier des candidats qualifiés - Au **public** de vérifier l'authenticité des documents

Le schéma est conçu pour garantir : - **Sécurité** : hachage SHA-256 + signatures Ed25519 - **Traçabilité** : logs de vérification et audit - **Performance** : indexation sur les champs de recherche fréquents - **Conformité RGPD** : journalisation, possibilité de suppression

TABLES ET CHAMPS DÉTAILLÉS

1. Utilisateur

Rôle : Table centrale gérant tous les comptes de la plateforme (étudiants, recruteurs, administrateurs, admins système).

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique auto-incrémenté	Oui
prenom	VARCHAR(100)	Nom de l'utilisateur	Oui
nom	VARCHAR(100)	Nom de famille de l'utilisateur	Oui
email	VARCHAR(255) UNIQUE	Adresse email (login unique)	Oui
mot_de_passe	VARCHAR(255)	Hash bcrypt ou Argon2 du mot de passe	Oui

Champ	Type	Description	Obligatoire
role	ENUM('etudiant', 'recruteur', 'administration', 'admin')	Type de compte	Oui
telephone	VARCHAR(20)	Numéro de téléphone avec indicatif pays	Non
photo_url	VARCHAR(500)	URL de la photo de profil (stockée sur S3/CDN)	Non
date_creation	TIMESTAMP	Date et heure de création du compte	Oui (auto)
derniere_connexion	TIMESTAMP	Dernière connexion réussie	Non
is_active	BOOLEAN	Compte actif (true) ou désactivé (false)	Oui (default: true)
langue	VARCHAR(5)	Code langue ISO (fr, en, es...) pour l'interface	Non (default: 'fr')
token_2fa_secret	VARCHAR(255)	Clé TOTP pour authentification 2 facteurs	Non

Explications supplémentaires : - **mot_de_passe_hash** : Ne JAMAIS stocker le mot de passe en clair. Utiliser bcrypt avec cost factor 12. - **role** : Détermine les permissions et l'interface affichée à l'utilisateur. - **is_active** : Permet de suspendre un compte sans le supprimer (RGPD).

Indexes recommandés : - **email** (UNIQUE) pour login rapide - **role** pour filtrer par type d'utilisateur

2. Administration

Rôle : Représente les établissements émetteurs de documents certifiés (universités, écoles, organismes de formation, administrations publiques).

Champ	Type	Description	Obligatoire
id	SERIAL PRIMARY KEY	Identifiant unique de l'administration	Oui

Champ	Type	Description	Obligatoire
nom	VARCHAR(255)	nom officiel de l'établissement	Oui
type_administration	ENUM('universite','type_d'établissement')	'ecole', 'organ- isme_formation', 'adminis- tra- tion_publique', 'autre')	Oui
pays	VARCHAR(100)	pays où se situe l'établissement	Oui
ville	VARCHAR(100)	ville principale	Non
adresse	TEXT	Adresse complète	Non
numero_accreditation	VARCHAR(100)	numero officiel d'accréditation ou SIRET/équivalent	Oui
email_contact	VARCHAR(255)	email officiel de contact	Oui
telephone_contact	VARCHAR(20)	téléphone de contact	Non
cle_publique_ed25519	TEXT	Clé publique Ed25519 en format PEM ou hex pour vérifier les signatures	Oui
logo_url	VARCHAR(500)	URL du logo officiel	Non
site_web	VARCHAR(255)	site web officiel	Non
date_inscription	TIMESTAMP	Date d'inscription sur la plateforme	Oui (auto)
statut	ENUM('en_attente','verifie','sus-pendu')	de vérification par la plateforme	Oui (default: 'en_attente')

Explications supplémentaires : - **numero_accreditation** : Vérifié manuellement lors de l'inscription pour garantir l'authenticité. - **cle_publique_ed25519** : Permet à quiconque de vérifier les signatures des documents émis. La clé privée correspondante ne doit JAMAIS être stockée en base (uniquement dans un HSM/KMS sécurisé). - **type_administration** : Permet de généraliser au-delà des universités (écoles, organismes, etc.). - **statut** : Seules les administrations "verifie" peuvent émettre des documents.

Indexes recommandés : - **numero_accreditation** (UNIQUE) - **statut** pour filtrer les administrations actives

3. Document

Rôle : Table centrale stockant tous les documents académiques certifiés (diplômes, relevés de notes, attestations).

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique du document	Oui
uuid_document	UUID UNIQUE	Identifiant public unique (pour URL de vérification)	Oui
etudiant_id	INTEGER FK → Utilisateur	Titulaire du document	Oui
administration	INTEGER FK → Adminis- tration	Établissement émetteur	Oui
type_document	ENUM('diplome', 'releve_notes', 'attesta- tion', 'certificat', 'autre')	Type de document	Oui
titre	VARCHAR(255)	Titré du document (ex: "Master Informatique")	Oui
file_url	VARCHAR(500)	URL du fichier original (PDF/image) stocké sur S3/CDN	Oui
file_size_kb	INTEGER	Taille du fichier en Ko	Non
hash_sha256	CHAR(64)	Empreinte SHA-256 du fichier (64 caractères hexadécimaux)	Oui
signature_ed25519		Signature numérique Ed25519 du hash (par l'administration)	Oui
qr_code_url	VARCHAR(500)	URL du QR code généré pour vérification rapide	Oui

Champ	Type	Description	Obligatoire
blockchain_tx_hash	VARCHAR(255)	hash de la transaction blockchain (si ancré)	Non
blockchain_network	VARCHAR(50)	nom du réseau blockchain (Ethereum, Polygon...)	Non
statut	ENUM('ACTIF', 'REVOQUE', 'EXPIRE')	État du document	Oui (default: 'ACTIF')
date_emission	DATE	Date d'obtention du document par l'étudiant	Oui
date_certification	TIMESTAMP	Date et heure de certification numérique sur la plateforme	Oui (auto)
date_expiration	DATE	Date d'expiration (si applicable, ex: certificats temporaires)	Non
metadata	JSONB	Métadonnées additionnelles : {année, mention, spécialité, note, crédits ECTS, etc.}	Non

Explications supplémentaires : - `uuid_document` : Utilisé dans les URLs publiques (ex: <https://plateforme.com/verify/{uuid}>) pour éviter d'exposer les IDs séquentiels. - `hash_sha256` : Empreinte unique du fichier. Le moindre changement dans le fichier produit un hash complètement différent. - `signature_ed25519` : Signature cryptographique du hash par l'administration. Prouve que l'administration a bien émis ce document. Vérifiable avec la clé publique de l'administration. - `qr_code_url` : QR code contenant le lien de vérification, incrusté sur le document PDF. - `blockchain_tx_hash` : Optionnel mais renforce la preuve d'antériorité. Une fois ancré sur blockchain, impossible de modifier la date. - `statut` : - ACTIF = valide - REVOQUE = annulé par l'administration (fraude, erreur) - EXPIRE = validité temporelle dépassée - `metadata` : Stocke des infos spécifiques en JSON flexible (pas besoin de créer 50 colonnes).

Indexes recommandés : - `uuid_document` (UNIQUE) pour vérification rapide - `hash_sha256` (UNIQUE) pour détecter les doublons et vérifier par upload - `etudiant_id` pour lister les documents d'un étudiant - `administration_id` pour stats par établissement - `statut` pour filtrer les documents actifs

Contrainte de sécurité : - Un même `hash_sha256` ne peut être certifié qu'une

seule fois (empêche la duplication).

4. Competence

Rôle : Référentiel des compétences professionnelles et académiques.

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique de la compétence	Oui
nom	VARCHAR(255)	Nom de la compétence (ex: “Python”, “Gestion de projet”, “Anglais”)	Oui
nom_normalise	VARCHAR(255)	Version normalisée en minuscules sans accents pour recherche	Oui
categorie	ENUM('technique', 'transversale', 'langue', 'logiciel', 'framework', 'domaine', 'autre')	Catégorie de compétence	Oui
description	TEXT	Description détaillée de la compétence	Non
referentiel_externe	VARCHAR(100)	Identifiant dans un référentiel externe (ex: ESCO, O*NET)	Non
synonymes	TEXT[]	Array de synonymes (ex: [“JS”, “JavaScript”])	Non
popularite	INTEGER	Score de popularité (nombre de fois utilisée)	Non (default: 0)
date_creation	TIMESTAMP	Date d'ajout de la compétence	Oui (auto)

Explications supplémentaires : - **nom_normalise** : Facilite la recherche insensible à la casse et aux accents (ex: “développement” → “developpe-

ment"). - **categorie** : Permet de filtrer et d'organiser les compétences. - **referentiel_externe_id** : Permet de mapper avec des standards internationaux (ESCO pour UE, O*NET pour US). - **synonymes** : Aide le moteur de matching à reconnaître "JavaScript" et "JS" comme identiques. - **popularite** : Permet de trier les compétences les plus demandées dans les analytics.

Indexes recommandés : - **nom_normalise** pour recherche rapide - **categorie** pour filtrage

5. ProfilEtudiant

Rôle : Informations complémentaires du profil étudiant (au-delà de la table Utilisateur).

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique du profil	Oui
utilisateur_id	INTEGER FK → Utilisateur UNIQUE	Lien vers le compte utilisateur	Oui
bio	TEXT	Présentation personnelle de l'étudiant	Non
cv_url	VARCHAR(500)	URL du CV complet (PDF)	Non
linkedin_url	VARCHAR(255)	profil LinkedIn	Non
github_url	VARCHAR(255)	profil GitHub (pour développeurs)	Non
portfolio_url	VARCHAR(255)	site web personnel/portfolio	Non
disponibilite	ENUM('immediat', '1_mois', '3_mois', 'non_disponible')	disponibilité pour un poste	Non
localisation_actuelle	VARCHAR(255)	pays actuel	Non
localisation_souhaitee	VARCHAR(255)	Ville(s) ou pays où l'étudiant cherche (peut être multiple)	Non

Champ	Type	Description	Obligatoire
mobilite	ENUM('localéType de mobilité 'nationale', acceptée 'internationale', 'teletravail')		Non
salaire_minimale	INTEGER	salaire minimum annuel en €/\$/monnaie locale	Non
types_contrat_souhaitez	EXTRA	Array : ['CDI', 'CDD', 'stage', 'alternance']	Non
profil_public	BOOLEAN	Le profil est visible par les recruteurs	Oui (default: true)
date_mise_aTTE	TIMESTAMP	Dernière modification du profil	Oui (auto update)

Explications supplémentaires : - **utilisateur_id** : Un utilisateur ne peut avoir qu'un seul profil étudiant (relation 1-1). - **bio** : Courte présentation permettant à l'étudiant de se démarquer. - **disponibilite** : Aide les recruteurs à filtrer selon l'urgence. - **profil_public** : Permet à l'étudiant de contrôler sa visibilité (RGPD). - **types_contrat_souhaitez** : Array PostgreSQL pour stocker plusieurs valeurs facilement.

Indexes recommandés : - **utilisateur_id** (UNIQUE) - **profil_public** pour filtrer les profils visibles

6. ProfilCompetence

Rôle : Table de liaison entre les profils étudiants et leurs compétences (extraction automatique par IA + ajout manuel).

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique de l'association	Oui
profil_etudiant	INTEGER	Profil concerné FK → ProfilEtu- diant	Oui

Champ	Type	Description	Obligatoire
competence _ID	INTEGER	Compétence détectée/ajoutée	Oui
niveau	ENUM('debut', 'intermediaire', 'avance', 'expert')	Niveau de maîtrise	Non
source	ENUM('ia_extraction', 'manuel', 'import_cv', 'validation_admin')	cette compétence a été ajoutée	Oui
source_document	INTEGER	Document d'où provient la compétence Document (si extraction IA)	Non
score_confiance	DECIMAL(5,2)	Score de confiance de l'extraction IA (0-100)	Non
annees_experience	DECIMAL(3,2)	Nombre d'années de pratique déclarées	Non
validee_par_etudiant	BOLLEAN	L'étudiant a confirmé cette compétence	Oui (default: false)
date_extraction	TIMESTAMP	Date d'ajout de la compétence	Oui (auto)
date_validation	TIMESTAMP	Date de validation par l'étudiant	Non

Explications supplémentaires : - **source** : Permet de tracer l'origine de la compétence : - **ia_extraction** : Détectée automatiquement par NLP dans un document - **manuel** : Ajoutée manuellement par l'étudiant - **import_cv** : Importée depuis un CV - **validation_admin** : Ajoutée/validée par un administrateur - **score_confiance** : Si extraction IA, indique la fiabilité (ex: 95% = très sûr, 60% = à vérifier). - **validee_par_etudiant** : L'étudiant peut accepter/refuser les compétences suggérées par l'IA. - **annees_experience** : Permet de valoriser l'expérience (ex: Python débutant vs Python 5 ans).

Contrainte : - Un couple (profil_etudiant_id, competence_id) doit être UNIQUE pour éviter les doublons.

Indexes recommandés : - (profil_etudiant_id, competence_id) UNIQUE - competence_id pour rechercher les profils ayant une compétence donnée

7. Offre

Rôle : Offres d'emploi publiées par les recruteurs.

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique de l'offre	Oui
recruteur_id	INTEGER FK → Utilisateur	Recruteur ayant publié l'offre	Oui
titre	VARCHAR(255)	Titre du poste (ex: “Développeur Full Stack Junior”)	Oui
description	TEXT	Description complète du poste, missions, environnement	Oui
entreprise	VARCHAR(255)	Nom de l'entreprise	Oui
secteur_activite	VARCHAR(100)	Acteur (IT, Finance, Santé, etc.)	Non
lieu	VARCHAR(255)	Localisation du poste (ville, pays)	Oui
type_contrat	ENUM('CDI', ‘CDD’, ‘stage’, ‘al- ternance’, ‘freelance’, ‘interim’)	Type de contrat proposé	Oui
duree_contrat	INTEGER	Durée en mois (pour CDD, stage, alternance)	Non
teletravail	ENUM('non', ‘partiel’, ‘total’)	Possibilité de télétravail	Non
salaire_min	INTEGER	Salaire minimum annuel brut	Non
salaire_max	INTEGER	Salaire maximum annuel brut	Non
devise	VARCHAR(3)	Code devise ISO (EUR, USD, etc.)	Non (default: ‘EUR’)

Champ	Type	Description	Obligatoire
niveau_etudes	ENUM	‘Niveau’ (‘bac’, ‘Niveau minimum’, ‘bac+2’, ‘bac+3’, ‘bac+5’, ‘bac+8’, ‘sans_diplome’)	Non
annees_experience	INTEGER	Années d’expérience minimales requises	Non
date_publication	TIMESTAMP	Date de publication de l’offre	Oui (auto)
date_expiration	DATE	Date limite de candidature	Non
statut	ENUM	‘BROUILLON’ (l’offre ‘PUBLIEE’, ‘EXPIREE’, ‘POURVUE’, ‘ARCHIVEE’)	Oui (default: ‘BROUILLON’)
nombre_vues	INTEGER	Nombre de fois que l’offre a été consultée	Non (default: 0)
nombre_candidatures	INTEGER	Nombre de candidatures reçues	Non (default: 0)

Explications supplémentaires : - **description** : Doit contenir toutes les infos (missions, profil recherché, avantages, process de recrutement). - **teletravail** : Information importante post-COVID. - **salaire_min/max** : Optionnels mais recommandés pour attirer plus de candidatures (transparence). - **statut** : - BROUILLON : Offre en cours de rédaction, non visible - PUBLIEE : Active et visible par les étudiants - EXPIREE : Date limite dépassée - POURVUE : Poste pourvu, offre retirée - ARCHIVEE : Conservée pour historique - **nombre_vues / nombre_candidatures** : Analytics pour le recruteur.

Indexes recommandés : - recruteur_id pour lister les offres d’un recruteur - statut pour filtrer les offres actives - date_publication pour tri chronologique

8. OffreCompetence

Rôle : Table de liaison entre les offres et les compétences requises/souhaitées.

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique	Oui
offre_id	INTEGER FK → Offre	Offre concernée	Oui
competence_id	INTEGER FK → Competence	Compétence requise/souhaitée	Oui
niveau_requis	ENUM('debut', 'intermediaire', 'avance', 'expert')	Niveau minimum attendu	Non
importance	ENUM('indispensable', 'importante', 'souhaitee', 'bonus')	Possibilité de la compétence	Oui
poids	INTEGER	Poids pour le calcul du score de matching (1-10)	Non (default: 5)

Explications supplémentaires : - **importance** : - indispensable : Deal breaker, candidat doit avoir - importante : Fortement recommandée - souhaitée : Nice to have - bonus : Avantage supplémentaire - **poids** : Utilisé par l'algorithme de matching. Compétences indispensables = poids 10, bonus = poids 1. - Le score de matching sera calculé en fonction du nombre de compétences communes pondérées par leur importance.

Contrainte : - (offre_id, competence_id) UNIQUE pour éviter les doublons

Indexes recommandés : - (offre_id, competence_id) UNIQUE - competence_id pour trouver toutes les offres demandant une compétence

9. Matching

Rôle : Table stockant les résultats du matching automatique entre offres et profils étudiants.

Champ	Type	Description	Obligatoire
id	SERIAL PRI-MARY KEY	Identifiant unique du matching	Oui
offre_id	INTEGER FK → Offre	Offre concernée	Oui
etudiant_id	INTEGER FK → Utilisateur	Étudiant concerné	Oui
score_global	DECIMAL(5,2)	Score de compatibilité global (0-100)	Oui
score_competences	DECIMAL(5,2)	Score basé uniquement sur les compétences (0-100)	Oui
score_localisation	DECIMAL(5,2)	Score de compatibilité géographique (0-100)	Non
score_experience	DECIMAL(5,2)	Score basé sur l'expérience (0-100)	Non
competences_JSON_fondnees	JSON	Liste des compétences en commun avec détails	Non
competences_JSON_quantifiées	JSON	Quantifiée Liste des compétences requises que l'étudiant n'a pas	Non
points_forts	TEXT[]	Array de points forts du profil pour cette offre	Non
points_amelioration	TEXT[]	Array de suggestions d'amélioration	Non
algorithme_version	VARCHAR(20)	Version de l'algorithme de matching utilisé	Oui
seuil_notification	DECIMAL(5,2)	Seuil utilisé pour déclencher la notification	Oui
notifie	BOOLEAN	Notification envoyée à l'étudiant	Oui (default: false)
date_notification	TIMESTAMP	Date d'envoi de la notification	Non
date_matching	TIMESTAMP	Date du calcul du matching	Oui (auto)
vu_par_etudiant	BOOLEAN	L'étudiant a consulté l'offre	Oui (default: false)
date_vue_etudiant	TIMESTAMP	Date de consultation par l'étudiant	Non

Champ	Type	Description	Obligatoire
interesse	BOOLEAN	L'étudiant a marqué son intérêt	Non
vu_par_recruit	BOOLEAN	Le recruteur a consulté le profil	Oui (default: false)

Explications supplémentaires : - score_global : Score final combinant plusieurs critères (compétences, localisation, expérience, etc.). - Formule exemple : `score_global = (score_competences * 0.7) + (score_localisation * 0.2) + (score_experience * 0.1)` - competences_matchees : JSON contenant les détails, exemple : `json [{"competence": "Python", "niveau_etudiant": "avance", "niveau_requis": "intermediaire", "match": true}, {"competence": "React", "niveau_etudiant": "intermediaire", "niveau_requis": "avance", "match": "partiel"}]` - competences_manquantes : JSON listant ce qui manque au candidat, exemple : `json [{"competence": "Docker", "niveau_requis": "intermediaire", "importance": "importante"}, {"competence": "AWS", "niveau_requis": "debutant", "importance": "souhaitee"}]` - algorithme_version : Permet de tracer quelle version de l'algorithme a produit ce score (utile si on améliore l'algorithme). - seuil_notification : Par défaut 65%, configurable. Si score_global > seuil → notification envoyée. - notifie : Une fois à true, on ne notifie plus (évite les doublons). - interesse : L'étudiant peut marquer son intérêt, ce qui remonte au recruteur.

Contrainte : - (offre_id, etudiant_id) UNIQUE pour éviter les doublons de matching

Indexes recommandés : - (offre_id, score_global DESC) pour lister les meilleurs matchs par offre - (etudiant_id, score_global DESC) pour lister les meilleures offres pour un étudiant - notifie pour traiter les matchings en attente de notification

10. Notification

Rôle : Système de notifications pour informer les utilisateurs (nouvelles offres, matchings, vérifications, alertes système).

Champ	Type	Description	Obligatoire
id	SERIAL PRI-MARY KEY	Identifiant unique de la notification	Oui

Champ	Type	Description	Obligatoire
destinataire	INTEGER	Utilisateur destinataire FK → Utilisateur	Oui
type	ENUM (‘nouvelle offre’, ‘notification matching’, ‘verification_document’, ‘message’, ‘alerte_systeme’, ‘rappel’)	Type de notification	Oui
priorite	ENUM (‘basse’, ‘normale’, ‘haute’, ‘urgente’)	Niveau de priorité	Oui (default: ‘normale’)
titre	VARCHAR(255)	Titre court de la notification	Oui
message	TEXT	Corps du message	Oui
lien_action	VARCHAR(500)	URL de redirection au clic	Non
icone	VARCHAR(50)	Nom de l’icône à afficher (ex: ‘bell’, ‘check’, ‘alert’)	Non
lue	BOOLEAN	Notification lue par le destinataire	Oui (default: false)
date_lecture	TIMESTAMP	Date et heure de lecture	Non
archivee	BOOLEAN	Notification archivée par l’utilisateur	Oui (default: false)
date_envoi	TIMESTAMP	Date de création de la notification	Oui (auto)
date_expiration	TIMESTAMP	Date après laquelle la notification peut être supprimée	Non
metadata	JSONB	Données additionnelles (ex: {offre_id: 123, score: 85})	Non

Explications supplémentaires : - **type** : Permet de filtrer et d’organiser les notifications. - **priorite** : Les notifications “urgente” peuvent être affichées différemment (popup, email immédiat). - **lien_action** : Ex: notification de matching → lien vers l’offre. - **lue** : Permet d’afficher le badge “nouvelles notifications”. - **archivee** : L’utilisateur peut archiver sans supprimer (RGPD : on garde l’historique). - **date_expiration** : Les notifications anciennes peuvent

être auto-supprimées après X jours (configurable).

Indexes recommandés : - (destinataire_id, lue) pour compter les notifications non lues - date_envoi DESC pour tri chronologique

11. VerificationLog

Rôle : Journalisation de toutes les vérifications de documents (traçabilité et analytics).

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique du log	Oui
document_id	INTEGER	Document vérifié FK → Document	Oui
verificateur_type	ENUM('publi' 'recruteur', 'adminis- tration', 'systeme')	Type de vérificateur	Oui
verificateur_id	INTEGER	ID utilisateur si authentifié FK → Utilisateur	Non
ip_hash	VARCHAR(64)	Hash SHA-256 de l'IP (RGPD : ne pas stocker IP en clair)	Oui
user_agent	TEXT	Navigateur/appareil utilisé	Non
pays	VARCHAR(100)	Code pays de l'IP (géolocalisation)	Non
ville	VARCHAR(100)	Ville de l'IP (si précision disponible)	Non
methode_verification	ENUM('mr_somme', 'up- load_fichier', 'url_direkte', 'api')	Comment la vérification a été effectuée	Oui

Champ	Type	Description	Obligatoire
resultat	ENUM('VALIDE', 'INVALIDE', 'VALIDE', 'REVOQUE', 'EXPIRE', 'ERREUR')	Résultat de la vérification	Oui
details_erreur	TEXT	Si erreur, message explicatif	Non
duree_ms	INTEGER	Temps de traitement de la vérification en millisecondes	Non
date_verification	TIMESTAMP	Date et heure de la vérification	Oui (auto)

Explications supplémentaires : - **ip_hash** : On ne stocke PAS l'IP en clair pour respecter le RGPD. On stocke son hash pour détecter les abus (ex: 1000 vérifications depuis la même IP). - **verificateur_type** : - **public** : Vérification anonyme (scan QR, upload sans login) - **recruteur** : Recruteur authentifié vérifiant un diplôme - **administration** : Établissement vérifiant un document d'un autre établissement - **système** : Vérification automatique interne - **methode_verification** : Analytics pour savoir quelle méthode est la plus utilisée. - **resultat** : - **VALIDE** : Document authentique et actif - **INVALIDE** : Hash ne correspond pas (document modifié ou faux) - **REVOQUE** : Document révoqué par l'émetteur - **EXPIRE** : Date d'expiration dépassée - **ERREUR** : Problème technique - **duree_ms** : Performance monitoring (doit être < 500ms pour une bonne UX).

Usage analytics : - Nombre de vérifications par jour/mois - Pays d'où proviennent les vérifications (mobilité internationale) - Taux de documents invalides détectés - Performance du système

Indexes recommandés : - **document_id** pour historique des vérifications d'un document - **date_verification DESC** pour stats temporelles - **resultat** pour analytics

12. Revocation

Rôle : Historique des révocations de documents (traçabilité obligatoire).

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique de la révocation	Oui
document_id	INTEGER FK → Document	Document révoqué	Oui
administration	INTEGER FK → Adminis- tration	Établissement ayant révoqué	Oui
operateur_id	INTEGER FK → Utilisateur	Personne ayant effectué la révocation	Oui
motif_categorie	ENUM('fraude', 'er- reur_administrative', 'annula- tion_diplome', 'de- mande_titulaire', 'autre')	Catégorie du motif	Oui
motif_detail	TEXT	Explication détaillée du motif	Oui
document_justificatif	MEDIUMBLOB	URL d'un document justifiant la révocation (si applicable)	Non
notification_titre	TEXT	Le titulaire a été notifié de la révocation	Oui
date_notification	TIMESTAMP	Date de notification au titulaire	Non
date_revocation	TIMESTAMP	Date effective de la révocation	Oui (auto)
irreversible	BOOLEAN	La révocation est définitive (ne peut être annulée)	Oui (default: true)

Explications supplémentaires : - **motif_categorie** : - **fraude** : Diplôme obtenu frauduleusement (triche, achat) - **erreur_administrative** : Erreur de l'administration (mauvais nom, mauvaise date) - **annulation_diplome** : Diplôme retiré après obtention (sanction disciplinaire) - **demande_titulaire** : L'étudiant demande le retrait (RGPD : droit à l'oubli partiel) - **autre** : Cas spécifiques - **motif_detail** : Obligatoire pour traçabilité légale. -

notification_titulaire : RGPD : le titulaire doit être informé sauf en cas de fraude avérée. - **irreversible** : Par défaut true. Si false, la révocation peut être annulée (ex: erreur administrative corrigée).

Règle métier : - Une fois révoqué, le document passe à statut “REVOQUE” dans la table Document. - Toute vérification ultérieure affichera “DOCUMENT REVOQUE”. - On ne supprime JAMAIS un document (traçabilité).

Indexes recommandés : - document_id pour historique - administration_id pour stats par établissement

13. AuditLog

Rôle : Journal d'audit de toutes les actions sensibles sur la plateforme (conformité, sécurité, forensics).

Champ	Type	Description	Obligatoire
id	SERIAL PRI- MARY KEY	Identifiant unique du log	Oui
utilisateur_id	INTEGER FK → Utilisateur	Utilisateur ayant effectué l'action (si authentifié)	Non
ip_hash	VARCHAR(64)	Hash de l'IP source	Oui
action	VARCHAR(100)	Type d'action (ex: ‘LOGIN’, ‘DOCUMEN-T_CERTIFIE’, ‘OFFRE_PUBLIEE’)	Oui
objet_type	VARCHAR(50)	Type d'objet concerné (Document, Offre, Utilisateur, etc.)	Non
objet_id	INTEGER	ID de l'objet concerné	Non
statut	ENUM(‘succes’, ‘echec’, ‘erreur’)	Résultat de l'action	Oui
details	JSONB	Détails de l'action en JSON	Non
message_erreur	TEXT	Message d'erreur si échec	Non
user_agent	TEXT	Navigateur/appareil	Non
date_action	TIMESTAMP	Date et heure de l'action	Oui (auto)

Explications supplémentaires : - `action` : Liste des actions auditées :
- LOGIN, LOGOUT, LOGIN_FAILED - DOCUMENT_CERTIFIE, DOCUMENT_REVOCATION - OFFRE_PUBLIEE, OFFRE_MODIFIEE - PROFIL_MODIFIE - VERIFICATION_DOCUMENT - EXPORT_DONNEES (RGPD) - SUPPRESSION_COMPTE - etc. - `details` : JSON contenant les infos pertinentes, exemple : `json { "ancien_email": "ancien@example.com", "nouvel_email": "nouveau@example.com", "champs_modifies": ["email", "telephone"] }` - `statut` : Permet de filtrer les échecs (tentatives de connexion ratées, etc.).

Usage : - Conformité RGPD : traçabilité des accès aux données personnelles
- Sécurité : détection d'activités suspectes (tentatives de connexion multiples, accès anormaux) - Forensics : investigation en cas d'incident - Analytics : comprendre l'usage de la plateforme

Indexes recommandés : - `utilisateur_id` pour historique par utilisateur - `action` pour filtrer par type d'action - `date_action DESC` pour chronologie - `statut` pour filtrer les échecs

Rétention des logs : - RGPD : conserver 1 à 3 ans selon la réglementation locale - Archivage automatique des logs > 1 an dans un stockage froid (S3 Glacier)

RELATIONS ET CONTRAINTES

Relations principales :

```
Utilisateur 1 < * Document (un utilisateur peut avoir plusieurs documents)
Administration 1 < * Document (une administration émet plusieurs documents)
Utilisateur 1 1 ProfilEtudiant (relation 1-1 si utilisateur est étudiant)
ProfilEtudiant 1 < * ProfilCompetence
Competence 1 < * ProfilCompetence
Utilisateur 1 < * Offre (un recruteur publie plusieurs offres)
Offre 1 < * OffreCompetence
Competence 1 < * OffreCompetence
Offre 1 < * Matching
Utilisateur 1 < * Matching (un étudiant peut matcher avec plusieurs offres)
Document 1 < * VerificationLog
Document 1 < * Revocation
Utilisateur 1 < * Notification
```

Contraintes d'intégrité :

1. **Unicité :**
 - `Utilisateur.email` : UNIQUE
 - `Document.uuid_document` : UNIQUE

- Document.hash_sha256 : UNIQUE (empêche duplication)
 - Administration.numero_accreditation : UNIQUE
 - (ProfilCompetence.profil_id, competence_id) : UNIQUE
 - (OffreCompetence.offre_id, competence_id) : UNIQUE
 - (Matching.offre_id, etudiant_id) : UNIQUE
2. Clés étrangères avec CASCADE :
 - Suppression d'un Utilisateur → CASCADE sur ProfilEtudiant, Notification
 - Suppression d'un Document → RESTRICT (impossible si des vérifications existent, traçabilité)
 - Suppression d'une Offre → CASCADE sur OffreCompetence, Matching
 3. Check constraints :
 - score_global, score_competences, etc. : BETWEEN 0 AND 100
 - date_expiration > date_publication (pour Offre)
 - salaire_max >= salaire_min (si les deux sont renseignés)
-

INDEXES POUR PERFORMANCE

```
-- Utilisateur
CREATE INDEX idx_utilisateur_email ON Utilisateur(email);
CREATE INDEX idx_utilisateur_role ON Utilisateur(role);

-- Document
CREATE UNIQUE INDEX idx_document_uuid ON Document(uuid_document);
CREATE UNIQUE INDEX idx_document_hash ON Document(hash_sha256);
CREATE INDEX idx_document_etudiant ON Document(etudiant_id);
CREATE INDEX idx_document_administration ON Document(administration_id);
CREATE INDEX idx_document_statut ON Document(statut);

-- Competence
CREATE INDEX idx_competence_nom_norm ON Competence(nom_normalise);
CREATE INDEX idx_competence_categorie ON Competence(categorie);

-- ProfilCompetence
CREATE UNIQUE INDEX idx_profil_comp_unique ON ProfilCompetence(profil_etudiant_id, competence_id);
CREATE INDEX idx_profil_comp_competence ON ProfilCompetence(competence_id);

-- Offre
CREATE INDEX idx_offre_recruteur ON Offre(recruteur_id);
CREATE INDEX idx_offre_statut ON Offre(statut);
CREATE INDEX idx_offre_date_pub ON Offre(date_publication DESC);

-- OffreCompetence
```

```

CREATE UNIQUE INDEX idx_offre_comp_unique ON OffreCompetence(offre_id, competence_id);
CREATE INDEX idx_offre_comp_competence ON OffreCompetence(competence_id);

-- Matching
CREATE UNIQUE INDEX idx_matching_unique ON Matching(offre_id, etudiant_id);
CREATE INDEX idx_matching_offre_score ON Matching(offre_id, score_global DESC);
CREATE INDEX idx_matching_etudiant_score ON Matching(etudiant_id, score_global DESC);
CREATE INDEX idx_matching_notifie ON Matching(notifie);

-- Notification
CREATE INDEX idx_notif_dest_lue ON Notification(destinataire_id, lue);
CREATE INDEX idx_notif_date ON Notification(date_envoi DESC);

-- VerificationLog
CREATE INDEX idx_verif_document ON VerificationLog(document_id);
CREATE INDEX idx_verif_date ON VerificationLog(date_verification DESC);

-- AuditLog
CREATE INDEX idx_audit_user ON AuditLog(utilisateur_id);
CREATE INDEX idx_audit_action ON AuditLog(action);
CREATE INDEX idx_audit_date ON AuditLog(date_action DESC);

```

SÉCURITÉ DES DONNÉES

1. Données sensibles chiffrées :

- mot_de_passe_hash : bcrypt ou Argon2 (jamais en clair)
- token_2fa_secret : chiffré au repos (AES-256)
- cle_publique_ed25519 : OK en clair (nature publique)
- Clés privées : JAMAIS en base, uniquement en HSM/KMS

2. Anonymisation :

- ip_hash : hash SHA-256 de l'IP (RGPD)
- Logs d'audit : anonymiser après 1 an

3. Chiffrement au repos :

- Base de données : chiffrement transparent (TDE pour PostgreSQL)
- Fichiers (S3) : chiffrement AES-256

4. Accès aux données :

- Principe du moindre privilège
- Utilisateurs DB séparés par rôle (lecture seule pour analytics, écriture limitée pour API)

CONFORMITÉ RGPD

Données personnelles identifiées :

- Utilisateur : nom, prénom, email, téléphone, photo
- ProfilEtudiant : bio, localisation, salaire
- Document : métadonnées

Mesures de conformité :

1. **Consentement :**
 - Acceptation explicite des CGU lors de l'inscription
 - Case à cocher pour profil public (opt-in)
 2. **Droit à l'oubli :**
 - Suppression du compte utilisateur → anonymisation des données
 - Conservation des logs d'audit (obligation légale) mais anonymisés
 3. **Portabilité :**
 - Export des données utilisateur en JSON (API /export-mes-donnees)
 4. **Limitation de la conservation :**
 - Comptes inactifs > 3 ans → notification puis suppression
 - Logs audit > 3 ans → archivage puis suppression
 5. **Traçabilité :**
 - AuditLog enregistre tous les accès aux données personnelles
-

VOLUMÉTRIE ESTIMÉE (1 AN)

Table	Nombre de lignes	Taille estimée
Utilisateur	50,000	10 MB
Administration	100	< 1 MB
Document	100,000	50 MB (+ fichiers S3)
Competence	5,000	2 MB
ProfilEtudiant	40,000	20 MB
ProfilCompetence	400,000	80 MB
Offre	5,000	50 MB
OffreCompetence	50,000	10 MB
Matching	500,000	200 MB
Notification	1,000,000	500 MB
VerificationLog	200,000	100 MB
AuditLog	5,000,000	2 GB
TOTAL	~7,000,000	~3 GB

Note : Volumétrie très raisonnable pour PostgreSQL. Pas besoin de sharding pour MVP.

ÉVOLUTIONS FUTURES POSSIBLES

1. **Indexation full-text :**
 - PostgreSQL `ts_vector` sur `Offre.description` et `ProfilEtudiant.bio` pour recherche textuelle avancée
 2. **Partitioning :**
 - Partitionner `AuditLog` et `VerificationLog` par date (mensuel) pour performance
 3. **Cache distribué :**
 - Redis pour cacher les documents fréquemment vérifiés
 4. **Search engine :**
 - Elasticsearch pour recherche avancée d'offres et profils
 5. **Data warehouse :**
 - Export vers un entrepôt de données pour analytics avancées (BigQuery, Snowflake)
-

CONCLUSION

Ce schéma de base de données est conçu pour : **Sécurité maximale** : signatures cryptographiques, hachage, audit **Traçabilité complète** : logs de vérification et audit **Performance** : indexes optimisés pour les requêtes fréquentes **Conformité RGPD** : anonymisation, droit à l'oubli, portabilité **Évolutivité** : structure normalisée, extensible

Le schéma est prêt pour un MVP et peut évoluer facilement vers des fonctionnalités avancées.

Prochaines étapes : 1. Validation du schéma 2. Génération des scripts SQL de création 3. Scripts de migration et seeds (données de test) 4. Documentation API REST basée sur ce schéma