



YABARA - Analyse des Données SQL



STRUCTURE DE BASE DE DONNÉES

TABLES PRINCIPALES

1. Table users (Gestion centralisée)

sql

users:

- id (BIGINT, PK, AI)
- email (VARCHAR(255), UNIQUE, NOT NULL)
- password_hash (VARCHAR(255), NOT NULL)
- user_type (ENUM: talent, entreprise, admin)
- status (ENUM: active, inactive, pending, suspended)
- email_verified_at (TIMESTAMP)
- created_at, updated_at (TIMESTAMP)

2. Table talents

sql

talents:

- id (BIGINT, PK, AI)
- user_id (BIGINT, FK -> users.id)
- first_name, last_name (VARCHAR)
- phone (VARCHAR(20))
- pole_id (INT, FK -> poles.id)
- famille_metier_id (INT, FK -> familles_metiers.id)
- niveau_etude (ENUM: BAC à BAC+8)
- cv_reference (VARCHAR(20), UNIQUE) -- CVYB0675BC
- profile_completion_percentage (DECIMAL(5,2))
- parrain_cv_reference (VARCHAR(20), FK -> talents.cv_reference)
- avatar_type (VARCHAR(50))
- total_applications, total_interviews, total_offers_viewed (INT)

3. Table entreprises

sql

entreprises:

- id (**BIGINT**, PK, AI)
- user_id (**BIGINT**, FK -> users.id)
- nom_entreprise (**VARCHAR**(255))
- logo_url (**VARCHAR**(500))
- pole_activite_id (**INT**, FK -> poles.id)
- numero_legal (**VARCHAR**(100)) -- *SIREN/SIRET/RCCM*
- effectif (**ENUM**: <50, 50-100, 100-500, >500)
- responsable_rh_nom, responsable_rh_prenom (**VARCHAR**)
- is_verified (**BOOLEAN**)
- total_offres_publiees, total_candidatures_recues (**INT**)
- notif_nouvelle_candidature, notif_deplacement_kanban (**BOOLEAN**)

4. Table admins

sql

admins:



- id (**BIGINT**, PK, AI)
- user_id (**BIGINT**, FK -> users.id)
- nom, prenom (**VARCHAR**(100))
- role (**ENUM**: super_admin, admin, moderator)
- permissions (**JSON**)
- mode_affichage (**ENUM**: etendue, condensee)
- mode_nuit (**BOOLEAN**)

TABLES DE RÉFÉRENCE







5. Table poles (5 pôles d'activité)

sql

poles:

- id (**INT**, PK, AI)
- nom (**VARCHAR**(100)) -- *TERTIAIRE, SECONDAIRE, etc.*
- icone (**VARCHAR**(50)) --  , , etc.
- ordre_affichage (**INT**)

Data:

1. TERTIAIRE (Services & Fonctions support)  
2. SECONDAIRE (Industrie, Construction & Production) 
3. NUMÉRIQUE & INNOVATION 
4. COMMERCIAL & RELATION CLIENT 
5. MÉTIERS PRATIQUES & ÉCONOMIE INFORMELLE 

6. Table familles_metiers

sql

familles_metiers:

- id (**INT**, PK, AI)
- pole_id (**INT**, FK -> poles.id)
- nom (**VARCHAR**(150))
- ordre_affichage (**INT**)

Exemples par pôle:

- Pôle 1: Administratif, Finance, RH, Conseil, Banque, Enseignement, Santé...
- Pôle 2: Industrie, BTP, Logistique, Environnement, Agriculture...
- Pôle 3: Digital, IT, Audiovisuel, Jeux Vidéo...
- Pôle 4: Marketing, Commerce, Immobilier...
- Pôle 5: Artisanat, Sécurité, Bien-être, Tourisme, Arts...

7. Table types_contrat

sql

types_contrat:

- id (**INT**, PK, AI)
- nom (**VARCHAR**(100))
- code (**VARCHAR**(20))

Data: CDI, CDD, TEMPS_PARTIEL, TEMPORAIRE, APPRENTISSAGE,
JOURNALIER, ESSAI, STAGE, ALTERNANCE

GESTION DES OFFRES

8. Table offres_emploi

sql

offres_emploi:

- id (**BIGINT**, PK, AI)
 - entreprise_id (**BIGINT**, FK -> entreprises.id)
 - reference_offre (**VARCHAR(20)**, **UNIQUE**) -- YB0953AC
 - titre (**VARCHAR(255)**)
 - descriptif (**TEXT**)
 - type_contrat_id (**INT**, FK -> types_contrat.id)
 - famille_metier_id (**INT**, FK -> familles_metiers.id)
 - niveau_diplome_requis (**ENUM**: BAC à BAC+8)
 - experience_minimum (**ENUM**: 0-2, 3-5, 6-10, +10)
 - remuneration_min, remuneration_max (**DECIMAL(10,2)**)
 - lieu_poste (**VARCHAR(255)**)
 - teletravail_possible, mobilite_nationale_exigee (**BOOLEAN**)
 - statut (**ENUM**: brouillon, publiee, expiree, pourvue, archivee)
 - nb_vues, nb_candidatures, nb_entretiens (**INT**)
 - date_publication, date_expiration (**TIMESTAMP**)
-



GESTION DES CV

9. Table cvs_anonymes

sql

cvs_anonymes:

- id (**BIGINT**, PK, AI)
 - talent_id (**BIGINT**, FK -> talents.id)
 - cv_original_path (**VARCHAR(500)**)
 - experiences (JSON) -- *Expériences structurées*
 - formations (JSON) -- *Formations structurées*
 - competences (JSON) -- *Compétences et niveaux*
 - langues (JSON) -- *Langues et niveaux*
 - informations_complementaires (**TEXT**)
 - version (**INT**)
 - is_current_version (**BOOLEAN**)
-



WORKFLOW CANDIDATURES

10. Table candidatures (Cœur du système)

sql

candidatures:

- id (**BIGINT**, PK, AI)
 - talent_id (**BIGINT**, FK -> talents.id)
 - offre_id (**BIGINT**, FK -> offres_emploi.id)
 - reference_candidature (**VARCHAR(20)**, **UNIQUE**)
- Statuts parallèles pour les 3 vues
- statut_talent (**ENUM**: en_validation_yabara, validee_yabara, validee_entreprise, entretien_cours, retenue)
 - statut_admin (**ENUM**: candidature_recue, envoyee_entreprise, preselctionnee, entretien, retenue)
 - statut_entreprise (**ENUM**: candidature_recue, preselctionnee, entretien, retenue)
- type_candidature (**ENUM**: spontanee, liee_admin, liee_entreprise)
 - coordonnees_partagees (**BOOLEAN**)
 - commentaires_internes (**TEXT**)
- Timestamps des étapes
- date_validation_yabara (**TIMESTAMP**)
 - date_envoi_entreprise (**TIMESTAMP**)
 - date_preselection (**TIMESTAMP**)
 - date_entretien (**TIMESTAMP**)
 - date_decision_finale (**TIMESTAMP**)

SYSTÈME DE GAMIFICATION

11. Table badges

sql

badges:

- id (**INT**, PK, AI)
- code (**VARCHAR(50)**, **UNIQUE**)
- nom (**VARCHAR(100)**)
- description (**TEXT**)
- icone (**VARCHAR(50)**)
- categorie (**ENUM**: talent, entreprise, admin)
- conditions (JSON) -- Conditions d'obtention structurées
- recompense_points (**INT**)
- rarete (**ENUM**: common, rare, epic, legendary)

Exemples Talents: Profil Complet 🗨️, Candidat Actif 🏠, Premier Pas 

Exemples Entreprises: Recruteur Express 🚀, Chasseur Talents 🗨️, Star du Mois 🌟

12. Table user_badges

sql

user_badges:

- id (**BIGINT**, PK, AI)
- user_id (**BIGINT**, FK -> users.id)
- badge_id (**INT**, FK -> badges.id)
- date_obtention (**TIMESTAMP**)
- progression_actuelle, progression_requise (**INT**)
- is_unlocked (**BOOLEAN**)

13. Table parrainages

sql

parrainages:

- id (**BIGINT**, PK, AI)
- parrain_id (**BIGINT**, FK -> users.id)
- parrain_type (**ENUM**: talent, entreprise)
- filleul_email (**VARCHAR**(255))
- filleul_id (**BIGINT**, FK -> users.id) -- *NULL jusqu'à inscription*
- statut (**ENUM**: invite, inscrit, actif, recrute)
- code_parrainage (**VARCHAR**(50), **UNIQUE**)
- recompense_parrain_accordee (**BOOLEAN**)

SYSTÈME DE NOTIFICATIONS

14. Table notifications

sql

notifications:

- id (**BIGINT**, PK, AI)
- user_id (**BIGINT**, FK -> users.id)
- type (**VARCHAR**(100))
- titre (**VARCHAR**(255))
- message (**TEXT**)
- data (**JSON**) -- *Données contextuelles*
- canal_email, canal_push, canal_in_app (**BOOLEAN**)
- is_read (**BOOLEAN**)
- priorite (**ENUM**: low, normal, high, urgent)

15. Table email_queue

sql

email_queue:

- id (**BIGINT**, PK, AI)
 - to_email (**VARCHAR(255)**)
 - sujet (**VARCHAR(255)**)
 - contenu_html (**TEXT**)
 - template_id (**INT**)
 - variables (JSON)
 - **status** (**ENUM**: pending, processing, sent, failed)
 - priorite (**ENUM**: low, normal, high, urgent)
 - attempts (**INT**)
 - send_at (**TIMESTAMP**)
-

★ FONCTIONNALITÉS UTILISATEUR

16. Table favoris

sql

favoris:

- id (**BIGINT**, PK, AI)
- user_id (**BIGINT**, FK -> users.id)
- favori_type (**ENUM**: offre, talent, entreprise)
- favori_id (**BIGINT**) -- *ID polymorphe*
- note_personnelle (**TEXT**)
- tags (JSON)

17. Table recherches_sauvegardees

sql

recherches_sauvegardees:

- id (**BIGINT**, PK, AI)
 - user_id (**BIGINT**, FK -> users.id)
 - nom_recherche (**VARCHAR(255)**)
 - type_recherche (**ENUM**: offres, talents)
 - criteres_recherche (JSON)
 - alerte_nouveaux_resultats (**BOOLEAN**)
-

ANALYTICS ET STATISTIQUES

18. Table analytics_events

sql

analytics_events:

- id (BIGINT, PK, AI)
- user_id (BIGINT, FK -> users.id)
- user_type (ENUM: talent, entreprise, admin, anonymous)
- session_id (VARCHAR(255))
- event_type (VARCHAR(100)) -- *page_view, search, apply, etc.*
- event_category, event_action (VARCHAR)
- metadata (JSON)
- ip_address (VARCHAR(45))

19. Table kpi_snapshots

sql

kpi_snapshots:

- id (BIGINT, PK, AI)
- date_snapshot (DATE, UNIQUE)
- total_talents_inscrits (INT)
- total_entreprises_actives (INT)
- total_offres_publiees (INT)
- total_candidatures_deposees (INT)
- nb_offres_pourvues (INT)
- taux_transformation_offre_embauche (DECIMAL(5,2))
- delai_moyen_recrutement_jours (DECIMAL(5,1))

SÉCURITÉ ET LOGS

20. Table login_logs

sql

login_logs:

- id (BIGINT, PK, AI)
- user_id (BIGINT, FK -> users.id)
- email (VARCHAR(255))
- status (ENUM: success, failed, blocked)
- ip_address (VARCHAR(45))
- user_agent (TEXT)
- is_suspicious (BOOLEAN)

21. Table audit_logs

sql

audit_logs:

- id (**BIGINT**, PK, AI)
- user_id (**BIGINT**, FK -> users.id)
- action (**VARCHAR(100)**) -- create, update, delete
- resource_type (**VARCHAR(100)**) -- candidature, offre, user
- resource_id (**BIGINT**)
- old_values, new_values (JSON)
- description (**TEXT**)

CONFIGURATION

22. Table system_settings

sql

system_settings:

- id (**INT**, PK, AI)
- setting_key (**VARCHAR(100)**, **UNIQUE**)
- setting_value (**TEXT**)
- setting_type (**ENUM**: string, integer, boolean, json)
- category (**VARCHAR(50)**)

Exemples:

- cv_reference_prefix: 'CVYB'
- offre_reference_prefix: 'YB'
- max_cv_file_size_mb: 10
- parainage_enabled: true
- badges_enabled: true

RELATIONS CLÉS

sql

```
-- Relations principales
users (1) -> (1) talents|entreprises|admins
poles (1) -> (*) familles_metiers
poles (1) -> (*) talents (via pole_id)
familles_metiers (1) -> (*) talents
familles_metiers (1) -> (*) offres_emploi
entreprises (1) -> (*) offres_emploi
talents (*) -> (*) offres_emploi (via candidatures)
users (1) -> (*) notifications
users (1) -> (*) user_badges
talents (1) -> (1) cvs_anonymes

-- Relations de parrainage
talents.cv_reference -> talents.parrain_cv_reference (auto-référence)
users (parrain) (1) -> (*) parrainages
```

Cette structure SQL pure supporte tous les workflows YABARA avec une approche relationnelle classique optimisée pour les performances et la cohérence des données.