

# Avenir AI Solutions - Documentation de Base de Données

Dernière mise à jour: 22 octobre 2025  
Nombre total de tables: 24  
Base de données: PostgreSQL (Supabase + Neon Failover)

## Table des Matières

- 1. [Gestion des Clients](#) (1 table)
- 2. [Traitement des Leads](#) (3 tables)
- 3. [Intelligence IA & Apprentissage](#) (5 tables)
- 4. [Découverte de Prospects](#) (4 tables)
- 5. [Automatisation des Approches](#) (5 tables)
- 6. [Système de Traduction](#) (2 tables)
- 7. [Traitement en Arrière-Plan](#) (1 table)
- 8. [Tables de Support](#) (3 tables)

## 1. Gestion des Clients

`clients`

**Objectif:** Stocke tous les comptes clients, clés API et paramètres de configuration.

**Ce qu'elle fait:**

- Gère l'inscription et l'authentification des clients
- Stocke les clés API uniques pour l'intégration de formulaires
- Garde les préférences de branding des clients (ton d'email, liens de réservation, industrie)
- Surveille la santé de la connexion (dernière connexion, dernier lead reçu)
- Supporte les clients bilingues (EN/FR)

**Champs clés:**

- `id` - UUID interne
- `client_id` - Identifiant public du client
- `api_key` - Clé unique pour l'authentification API
- `business_name` - Nom de l'entreprise du client
- `email` - Email de connexion du client
- `password_hash` - Mot de passe chiffré

- `language` - Langue préférée (en/fr)
- `email_tone` - Style d'email (Professional/Friendly/Formal/Energetic)
- `ai_personalized_reply` - Basculer pour les réponses email automatisées
- `booking_link` - Lien de calendrier de réservation personnalisé
- `last_connection` - Quand le dernier lead a été reçu
- `is_test` - Marque les comptes de test/démo
- `is_internal` - Marque les comptes internes Avenir

Nombre de lignes: ~5-10 clients

## 2. Traitement des Leads

### `lead_memory`

**Objectif:** Stockage central pour tous les leads entrants avec données d'enrichissement IA.

**Ce qu'elle fait:**

- Capture chaque lead des formulaires de contact des clients
- Stocke l'analyse IA (intention, ton, urgence, confiance)
- Suit les changements de statut (archivé, supprimé, tagué)
- Maintient l'historique des prédictions IA au fil du temps
- Lie les leads à des clients spécifiques

**Champs clés:**

- `id` - Identifiant unique du lead
- `name` - Nom du lead
- `email` - Email du lead
- `message` - Message original du lead
- `ai_summary` - Résumé généré par l'IA
- `intent` - Intention détectée par l'IA (ex: "Partenariat B2B")
- `tone` - Ton détecté par l'IA (ex: "Professionnel")
- `urgency` - Urgence détectée par l'IA (Élevée/Moyenne/Faible)
- `confidence_score` - Confiance de l'IA (0.00-1.00)
- `tone_history` - Tableau JSONB des changements de ton
- `confidence_history` - Tableau JSONB des changements de confiance
- `urgency_history` - Tableau JSONB des changements d'urgence
- `archived` - Statut archivé du lead
- `deleted` - Statut supprimé du lead
- `current_tag` - Tag actuel (ex: "Lead Chaud", "Converti")
- `client_id` - À quel client ce lead appartient
- `language` - Langue du lead (en/fr)
- `is_test` - Marque les leads de test

**Nombre de lignes:** Croissant (100-1000+ leads par client)

---

### `lead_actions`

**Objectif:** Piste d'audit de toutes les actions effectuées sur les leads.

**Ce qu'elle fait:**

- Enregistre chaque action sur les leads (tag, archivage, suppression, réactivation)
- Suit les événements de conversion
- Enregistre les raisons de réversion (si un lead converti est réverti)
- Alimente le Journal d'Activité dans le tableau de bord client
- Permet des analyses spécifiques au client

**Champs clés:**

- `id` - UUID de l'action
- `lead_id` - À quel lead cette action s'applique
- `client_id` - Quel client a effectué l'action
- `action_type` - Type d'action ("tagged", "archived", "deleted", "reactivated", "created")
- `tag` - Tag appliqué (si l'action est un tagging)
- `timestamp` - Quand l'action s'est produite
- `conversion_outcome` - TRUE si le lead a converti
- `reversion_reason` - Pourquoi le lead converti a été réverti
- `is_test` - Marque les actions de test

**Nombre de lignes:** Croissant (plusieurs actions par lead)

**Note:** La colonne est `action_type` dans la base de données, mappée à `action` dans le frontend.

---

### `lead_notes`

**Objectif:** Stocke les notes ajoutées par les utilisateurs pour les leads.

**Ce qu'elle fait:**

- Permet aux clients d'ajouter des notes aux leads
- Suit qui a ajouté la note
- Maintient l'historique des notes
- Supporte l'UI de notes extensibles dans le tableau de bord

**Champs clés:**

- `id` - UUID de la note
- `lead_id` - À quel lead cette note appartient
- `client_id` - Quel client a créé la note
- `note` - Le contenu de la note

- `performed_by` - Qui a créé la note (généralement "admin")
- `created_at` - Quand la note a été créée
- `updated_at` - Quand la note a été modifiée pour la dernière fois
- `is_test` - Marque les notes de test

**Nombre de lignes:** Variable (0-10+ notes par lead)

---

### 3. Intelligence IA & Apprentissage

`growth_brain`

**Objectif:** Stocke les insights de croissance générés par l'IA et les analyses pour chaque client.

**Ce qu'elle fait:**

- Analyse TOUS les leads actifs pour chaque client
- Identifie les principales intentions, les modèles d'urgence, les distributions de ton
- Calcule les scores d'engagement
- Génère des insights prédictifs (bilingues)
- Mises à jour hebdomadaires ou à la demande via "Copilote de Croissance"

**Champs clés:**

- `id` - UUID de l'insight
- `client_id` - Pour quel client cette analyse est faite
- `analysis_period_start` - Début de la plage d'analyse
- `analysis_period_end` - Fin de la plage d'analyse
- `total_leads` - Combien de leads analysés
- `top_intents` - Tableau JSONB des 5 principales intentions
- `urgency_distribution` - JSONB: {high, medium, low} comptes
- `urgency_trend_percentage` - Changement d'urgence semaine après semaine
- `tone_distribution` - Tableau JSONB des fréquences de ton
- `tone_sentiment_score` - Score de signal professionnel 0-100
- `avg_confidence` - Confiance IA moyenne sur tous les leads
- `confidence_trajectory` - Tableau JSONB suivant la confiance au fil du temps
- `language_ratio` - JSONB: {en, fr} pourcentages
- `engagement_score` - Métrique composite (0-100)
- `predictive_insights` - JSONB: {en: {}, fr: {}} avec recommandations
- `analyzed_at` - Quand l'analyse a été effectuée

**Nombre de lignes:** 1 ligne par client (UPSERT sur `client_id` )

**Contrainte unique:** `growth_brain_client_unique` empêche les doublons

---

### feedback\_tracking

**Objectif:** Suit les résultats de l'IA et les retours utilisateurs pour l'apprentissage continu.

**Ce qu'elle fait:**

- Enregistre chaque prédiction de l'IA et son résultat
- Compare les résultats prédits vs. réels
- Suit les résultats positifs/négatifs/neutres
- Alimente le système d'apprentissage adaptatif
- Expire après 1 an (nettoyage automatique)

**Champs clés:**

- `id` - UUID du feedback
- `lead_id` - Lead associé (optionnel)
- `client_id` - Client associé (optionnel)
- `action_type` - Catégorie de feedback: `lead_conversion`, `email_response`, `user_action`, `system_performance`
- `outcome` - Résultat: `positive`, `negative`, `neutral`
- `confidence_score` - Confiance dans ce feedback (0.0-1.0)
- `impact_score` - Impact sur le système (-100 à +100)
- `context_data` - JSONB avec détails spécifiques à l'action
- `notes` - Description lisible par l'humain
- `notes_en` - Notes en anglais
- `notes_fr` - Notes en français
- `learning_applied` - A été traité par les algorithmes d'apprentissage?
- `learning_impact` - Impact sur les modèles IA (0.0-1.0)
- `processed_at` - Quand le feedback a été traité
- `expires_at` - Date de nettoyage automatique (NOW + 1 an)

**Nombre de lignes:** Croissant (chaque prédiction IA enregistrée)

### performance\_metrics

**Objectif:** Suit les performances du système, la précision de l'IA et les temps de réponse.

**Ce qu'elle fait:**

- Surveille les temps de réponse API
- Suit la précision de l'analyse IA
- Mesure les taux de succès
- Détecte les modèles d'erreurs
- Permet l'optimisation des performances

**Champs clés:**

- `id` - UUID de la métrique

- `event_type` - Catégorie: `api_response`, `ai_analysis`, `translation`, `lead_processing`, `email_response`
- `metric_name` - Métrique spécifique: `response_time`, `accuracy`, `success_rate`, `error_rate`
- `metric_value` - La mesure réelle
- `metric_unit` - Unité: `ms`, `percent`, `count`, `score`
- `response_time_ms` - Temps de réponse en millisecondes
- `success_rate` - Pourcentage de succès (0.0-1.0)
- `ai_accuracy` - Score de précision IA (0.0-1.0)
- `error_count` - Nombre d'erreurs
- `source_component` - Quel composant a enregistré ceci: `lead_api`, `translation_service`, `intelligence_engine`
- `client_id` - Métriques spécifiques au client (optionnel)
- `recorded_at` - Quand la métrique a été enregistrée
- `metadata` - JSONB avec contexte supplémentaire
- `error_message_en` - Message d'erreur en anglais
- `error_message_fr` - Message d'erreur en français

**Nombre de lignes:** Croissant (journalisation continue)

---

`prompt_registry`

**Objectif:** Stocke toutes les variantes de prompts IA avec scores de performance.

**Ce qu'elle fait:**

- Suit différentes versions de prompts IA
- Note les performances des prompts
- Sélectionne les prompts les plus performants
- Permet les tests A/B
- Alimente le système d'évolution des prompts

**Champs clés:**

- `id` - UUID du prompt
- `prompt_name` - Identifiant du prompt
- `variant_version` - Numéro de version
- `variant_id` - Identifiant unique de la variante
- `prompt_text` - Le contenu réel du prompt
- `language` - Langue du prompt (en/fr)
- `score` - Score de performance (0.0-1.0)
- `usage_count` - Combien de fois utilisé
- `created_at` - Quand créé
- `updated_at` - Dernière mise à jour

**Nombre de lignes:** Croissant (plusieurs variantes par prompt)

---

### `prompt_performance`

**Objectif:** Suit les résultats d'exécution individuels des prompts IA.

**Ce qu'elle fait:**

- Enregistre chaque exécution de prompt
- Mesure la qualité, la précision, la cohérence
- Suit l'utilisation des tokens et les coûts
- Lie au feedback pour l'apprentissage
- Permet l'optimisation des prompts

**Champs clés:**

- `id` - UUID d'exécution
- `prompt_registry_id` - Quelle variante de prompt a été utilisée
- `execution_id` - Identifiant unique d'exécution
- `client_id` - Suivi spécifique au client (optionnel)
- `input_data` - JSONB entrée au prompt
- `output_data` - JSONB sortie du prompt
- `output_quality_score` - Note de qualité (0.000-1.000)
- `response_time_ms` - Vitesse d'exécution du prompt
- `token_count` - Tokens utilisés
- `cost_usd` - Coût en USD
- `accuracy_score` - Note de précision
- `consistency_score` - Score de conformité au format
- `completeness_score` - Note de complétude
- `error_occurred` - Une erreur s'est-elle produite?
- `error_message` - Détails de l'erreur
- `feedback_id` - Lien vers l'enregistrement de feedback
- `user_rating` - Note utilisateur 1-5 (optionnel)
- `executed_at` - Quand exécuté
- `environment` - `production`, `staging`, `test`

**Nombre de lignes:** Croissant (chaque exécution de prompt enregistrée)

### `prompt_ab_tests`

**Objectif:** Gère les tests A/B des variantes de prompts.

**Ce qu'elle fait:**

- Configure les tests A/B pour les prompts
- Alloue le trafic entre contrôle/traitement
- Suit les résultats des tests et la signification statistique
- Sélectionne les variantes de prompts gagnantes

- Automatise l'optimisation des prompts

### Champs clés:

- `id` - UUID du test
- `test_name` - Identifiant du test
- `prompt_name` - Quel prompt est testé
- `control_variant_id` - Variante de prompt de contrôle
- `treatment_variant_id` - Variante de prompt de traitement
- `control_traffic_percentage` - Allocation de trafic de contrôle
- `treatment_traffic_percentage` - Allocation de trafic de traitement
- `min_sample_size` - Échantillons minimums avant de conclure (défaut: 100)
- `max_duration_days` - Durée maximale du test (défaut: 7 jours)
- `significance_level` - Signification statistique (défaut: 0.05)
- `status` - `draft`, `running`, `completed`, `cancelled`
- `control_metrics` - JSONB résultats du groupe de contrôle
- `treatment_metrics` - JSONB résultats du groupe de traitement
- `statistical_significance` - Valeur P
- `winner_variant_id` - Prompt gagnant

Nombre de lignes: ~10-50 tests

`prompt_evolution`

**Objectif:** Suit comment les prompts IA évoluent et s'améliorent au fil du temps.

### Ce qu'elle fait:

- Enregistre les relations parent-enfant des prompts
- Enregistre les stratégies d'évolution (mutation, crossover, optimisation)
- Compare les améliorations de performance
- Lie le feedback aux changements de prompts
- Documente l'historique d'évolution

### Champs clés:

- `id` - UUID d'évolution
- `parent_prompt_id` - Prompt original
- `child_prompt_id` - Prompt évolué
- `evolution_type` - `mutation`, `crossover`, `optimization`, `manual_edit`
- `evolution_strategy` - Stratégie spécifique utilisée
- `parent_performance` - JSONB métriques du parent
- `child_performance` - JSONB métriques de l'enfant
- `improvement_score` - Amélioration mesurée (0.000-1.000)
- `feedback_data` - JSONB feedback qui a conduit à l'évolution
- `optimization_goals` - Tableau des objectifs
- `evolved_at` - Quand l'évolution s'est produite



- `evolution_algorithm` - Algorithme utilisé

**Nombre de lignes:** Croissant (suit toutes les améliorations de prompts)

---

## 4. Découverte de Prospects

### `prospect_candidates`

**Objectif:** Stocke les prospects découverts de People Data Labs, Google, Apollo.

**Ce qu'elle fait:**

- Contient toutes les entreprises prospects découvertes
- Suit les scores de besoin d'automatisation (page 45-95)
- Surveille le statut de contact
- Lie aux journaux d'approche et aux tests de formulaires
- Alimente la file d'attente quotidienne de prospects

**Champs clés:**

- `id` - UUID du prospect
- `business_name` - Nom de l'entreprise
- `website` - Site web de l'entreprise (unique)
- `contact_email` - Email de contact
- `industry` - Catégorie d'industrie
- `region` - Région géographique
- `language` - Langue cible (en/fr)
- `form_url` - URL du formulaire de contact
- `last_tested` - Date du dernier test de formulaire
- `response_score` - Score de qualité de réponse
- `automation_need_score` - Besoin d'automatisation calculé par l'IA (45-95)
- `contacted` - A été contacté?
- `metadata` - JSONB avec source de découverte, données LinkedIn, etc.
- `created_at` - Quand découvert
- `updated_at` - Dernière mise à jour

**Nombre de lignes:** 34,823+ prospects du mapping taxonomique PDL

---

### `prospect_outreach_log`

**Objectif:** Suit tous les emails d'approche envoyés aux prospects.

**Ce qu'elle fait:**

- Enregistre chaque email d'approche

- Suit l'engagement (ouvert, répondu)
- Surveille le statut de l'email
- Stocke le contenu de la réponse
- Permet l'analyse des performances d'approche

#### Champs clés:

- `id` - UUID du journal
- `prospect_id` - Quel prospect a été contacté
- `subject` - Ligne d'objet de l'email
- `email_body` - Contenu de l'email
- `sent_at` - Quand l'email a été envoyé
- `opened_at` - Quand l'email a été ouvert
- `replied_at` - Quand le prospect a répondu
- `status` - `sent`, `opened`, `replied`, `bounced`, `ignored`
- `reply_content` - Réponse du prospect
- `metadata` - JSONB avec infos de campagne

**Nombre de lignes:** Croissant (une ligne par email d'approche)

---

`prospect_industry_performance`

**Objectif:** Agrège les performances des prospects par industrie pour l'apprentissage.

#### Ce qu'elle fait:

- Suit quelles industries répondent le mieux
- Calcule les taux d'ouverture et de réponse par industrie
- Mesure les temps de réponse moyens
- Assigne des scores de priorité aux industries
- Optimise le ciblage basé sur les performances

#### Champs clés:

- `id` - UUID de performance
- `industry` - Nom de l'industrie (unique)
- `total_contacted` - Nombre total d'approches
- `total_opened` - Total d'emails ouverts
- `total_replied` - Total de réponses reçues
- `open_rate` - Pourcentage d'ouverture
- `reply_rate` - Pourcentage de réponse
- `avg_response_time_hours` - Temps moyen de réponse
- `priority_score` - Priorité de l'industrie (0-100)
- `last_updated` - Date du dernier calcul

**Nombre de lignes:** ~10-20 industries

---

### prospect\_form\_tests

**Objectif:** Enregistre les résultats des tests automatisés de formulaires.

**Ce qu'elle fait:**

- Teste les formulaires de contact des prospects
- Mesure la qualité de la réponse
- Détecte les répondeurs automatiques
- Note la personnalisation des répondeurs automatiques
- Valide la fonctionnalité des formulaires

**Champs clés:**

- `id` - UUID du test
- `prospect_id` - Quel prospect a été testé
- `test_submitted_at` - Quand le test de formulaire a été soumis
- `response_received_at` - Quand la réponse est arrivée
- `response_time_minutes` - Délai de réponse
- `has_autoresponder` - Avaient-ils un répondeur automatique?
- `autoresponder_tone` - `robotic`, `human`, `personalized`, `none`
- `autoresponder_content` - Le message du répondeur automatique
- `score` - Score de qualité (0-100)
- `test_status` - `pending`, `completed`, `failed`, `timeout`
- `metadata` - JSONB avec détails du test

**Nombre de lignes:** Croissant (une ligne par test de formulaire)

## 5. Automatisation des Approches

### outreach\_campaigns

**Objectif:** Gère les campagnes d'approche pour l'engagement des prospects.

**Ce qu'elle fait:**

- Organise les approches en campagnes
- Lie les campagnes aux clients
- Suit le statut de la campagne
- Stocke les critères de ciblage
- Gère les calendriers de suivi

**Champs clés:**

- `id` - UUID de campagne
- `name` - Nom de la campagne
- `client_id` - Quel client possède cette campagne

- `status` - `draft`, `active`, `paused`, `completed`
- `target_criteria` - JSONB règles de filtrage
- `email_template_id` - Quel modèle utiliser
- `follow_up_schedule` - JSONB timing de suivi
- `created_at` - Date de création de la campagne
- `updated_at` - Dernière mise à jour
- `metadata` - JSONB paramètres de campagne

**Nombre de lignes:** Croissant (~5-20 campagnes par client)

#### `outreach_emails`

**Objectif:** Stocke les emails d'approche individuels avec suivi.

**Ce qu'elle fait:**

- Génère des emails d'approche personnalisés
- Suit le statut de l'email (pending → sent → opened → replied)
- Lie aux prospects et campagnes
- Surveille la livraison Gmail
- Gère les séquences de suivi

**Champs clés:**

- `id` - UUID de l'email
- `campaign_id` - À quelle campagne cet email appartient
- `prospect_id` - Prospect cible
- `prospect_email` - Adresse email du prospect
- `prospect_name` - Nom du prospect
- `company_name` - Nom de l'entreprise
- `website` - Site web de l'entreprise
- `template_id` - Modèle d'email utilisé
- `subject` - Objet de l'email
- `content` - Corps de l'email (HTML)
- `status` - `pending`, `approved`, `rejected`, `sent`, `delivered`, `opened`, `replied`, `bounced`
- `sent_at` - Quand envoyé
- `opened_at` - Quand ouvert
- `replied_at` - Quand répondu
- `thread_id` - ID de fil Gmail
- `gmail_message_id` - ID de message Gmail
- `follow_up_sequence` - Numéro de suivi (1, 2, 3...)
- `sender_email` - Adresse email d'envoi
- `missing_email` - Indicateur pour prospects sans email
- `metadata` - JSONB détails de suivi

**Nombre de lignes:** Croissant (une ligne par email)

---

### outreach\_tracking

**Objectif:** Suit les événements granulaires d'engagement par email.

**Ce qu'elle fait:**

- Enregistre chaque action email (envoyé, livré, ouvert, cliqué, répondu)
- Horodate chaque événement
- Lie les événements aux emails, prospects, campagnes
- Permet des analyses d'engagement détaillées

**Champs clés:**

- `id` - UUID de l'événement
- `email_id` - Pour quel email cet événement est-il
- `prospect_id` - Prospect associé
- `campaign_id` - Campagne associée
- `action` - Type d'événement: `sent`, `delivered`, `opened`, `clicked`, `replied`, `bounced`, `unsubscribed`
- `timestamp` - Quand l'événement s'est produit
- `metadata` - JSONB détails de l'événement

**Nombre de lignes:** Croissant (plusieurs événements par email)

---

### outreach\_metrics

**Objectif:** Agrège les métriques de performance des campagnes d'approche.

**Ce qu'elle fait:**

- Calcule les statistiques au niveau de la campagne
- Suit les emails totaux envoyés, ouverts, répondus
- Calcule les taux d'ouverture et de réponse
- Surveille les revenus générés par les campagnes
- Permet l'optimisation des campagnes

**Champs clés:**

- `id` - UUID des métriques
- `campaign_id` - Pour quelle campagne ces métriques sont-elles
- `total_emails_sent` - Nombre total envoyé
- `total_emails_delivered` - Livrés avec succès
- `total_emails_opened` - Emails ouverts
- `total_emails_clicked` - Liens cliqués
- `total_emails_replied` - Réponses reçues
- `total_emails_bounced` - Emails rebondis
- `total_emails_unsubscribed` - Désabonnements

- `open_rate` - Pourcentage d'ouverture
- `click_rate` - Pourcentage de clic
- `reply_rate` - Pourcentage de réponse
- `bounce_rate` - Pourcentage de rebond
- `revenue_generated` - Revenus USD de la campagne
- `created_at` - Quand le suivi des métriques a commencé
- `updated_at` - Dernière mise à jour

**Nombre de lignes:** 1 ligne par campagne (mise à jour continue)

---

### `email_templates`

**Objectif:** Stocke les modèles d'email réutilisables pour les approches.

**Ce qu'elle fait:**

- Gère les modèles d'email
- Supporte les variables de personnalisation
- Suit les performances des modèles
- Permet les tests A/B
- Alimente les approches automatisées

**Champs clés:**

- `id` - UUID du modèle
- `name` - Nom du modèle
- `description` - Description du modèle
- `subject` - Objet de l'email (avec variables)
- `html_content` - Corps de l'email HTML
- `plain_content` - Corps de l'email en texte brut
- `language` - Langue du modèle (en/fr)
- `category` - Type de modèle
- `variables` - JSONB liste des variables de personnalisation
- `is_active` - Modèle activé?
- `usage_count` - Foix utilisé
- `performance_score` - Note de performance (0.0-1.0)
- `created_at` - Quand créé
- `updated_at` - Dernière mise à jour

**Nombre de lignes:** ~10-30 modèles

---

## 6. Système de Traduction

### translation\_cache

**Objectif:** Met en cache les traductions IA pour réduire les coûts API.

**Ce qu'elle fait:**

- Stocke le texte traduit pour réutilisation
- Empêche les appels API de traduction en double
- Accélère les réponses bilingues
- Suit la qualité de la traduction
- Expire les vieilles traductions (90 jours)

**Champs clés:**

- `id` - UUID du cache
- `source_text` - Texte original
- `source_language` - Langue source (en/fr)
- `target_language` - Langue cible (fr/en)
- `translated_text` - Résultat traduit
- `context_type` - Contexte: `lead_summary`, `intent`, `tone`, `ui_text`
- `quality_score` - Qualité de la traduction (0.0-1.0)
- `verified` - Vérifié par un humain?
- `created_at` - Quand traduit
- `last_used_at` - Dernière utilisation
- `expires_at` - Date d'expiration (NOW + 90 jours)

**Nombre de lignes:** Croissant (traductions mises en cache)

### translation\_dictionary

**Objectif:** Stocke les traductions vérifiées pour les termes courants.

**Ce qu'elle fait:**

- Mappe des termes spécifiques aux traductions vérifiées
- Assure une terminologie cohérente
- Fournit des recherches instantanées (pas d'IA nécessaire)
- Supporte la correspondance floue
- Glossaire bilingue

**Champs clés:**

- `id` - UUID du dictionnaire
- `source_text` - Terme original
- `source_language` - Langue source
- `target_language` - Langue cible
- `translated_text` - Traduction vérifiée
- `context` - Contexte d'utilisation

- `category` - Catégorie de terme
- `verified` - Vérifié par un humain?
- `usage_count` - Fois utilisé
- `confidence` - Confiance de traduction (0.0-1.0)
- `created_at` - Quand ajouté
- `updated_at` - Dernière mise à jour

**Nombre de lignes:** ~50-200 termes vérifiés

---

## 7. Traitement en Arrière-Plan

`queue_jobs`

**Objectif:** File d'attente de travaux en arrière-plan pour les opérations longues.

**Ce qu'elle fait:**

- File d'attente des travaux asynchrones (découverte quotidienne de prospects, opérations en masse)
- Traite les travaux avec un délai d'expiration de 300 secondes (vs. limite API de 60s)
- Suit le statut des travaux (pending → processing → completed/failed)
- Stocke les résultats et erreurs des travaux
- Permet un traitement en arrière-plan fiable

**Champs clés:**

- `id` - UUID du travail
- `job_type` - Catégorie de travail: `daily_prospect_queue`, `bulk_email`, `intelligence_analysis`
- `status` - `pending`, `processing`, `completed`, `failed`
- `payload` - JSONB données d'entrée pour le travail
- `result` - JSONB données de sortie après achèvement
- `error` - Message d'erreur si échoué
- `created_at` - Quand le travail a été mis en file d'attente
- `started_at` - Quand le traitement a commencé
- `completed_at` - Quand le travail s'est terminé

**Nombre de lignes:** Croissant (nettoyé périodiquement après achèvement)

---

## 8. Tables de Support

`integration_logs`

**Objectif:** Enregistre toutes les intégrations API et soumissions de formulaires.

**Ce qu'elle fait:**



- Suit chaque requête API des formulaires clients
- Enregistre les données de requête/réponse
- Surveille la santé de l'intégration
- Débogue les problèmes d'intégration
- Valide l'utilisation de la clé API

**Champs clés:**

- `id` - UUID du journal
- `client_id` - Intégration de quel client
- `api_key` - Clé API utilisée (partielle)
- `endpoint` - Point de terminaison API appelé
- `method` - Méthode HTTP
- `request_body` - JSONB données de requête
- `response_status` - Code de statut HTTP
- `response_body` - JSONB données de réponse
- `ip_address` - IP de la requête
- `user_agent` - User agent de la requête
- `created_at` - Quand enregistré

**Nombre de lignes:** Croissant (chaque appel API enregistré)

---

`avenir_profile_embeddings`

**Objectif:** Stocke les embeddings IA du profil de l'entreprise Avenir.

**Ce qu'elle fait:**

- Contient les embeddings vectoriels de la description de l'entreprise
- Permet la correspondance sémantique avec les prospects
- Alimente le scoring de similarité
- Optimise le ciblage des prospects

**Champs clés:**

- `id` - UUID de l'embedding
- `chunk_text` - Morceau de texte du profil de l'entreprise
- `embedding` - Embedding vectoriel (1536 dimensions)
- `embedding_model` - Modèle utilisé (ex: text-embedding-ada-002)
- `metadata` - JSONB métadonnées du morceau
- `created_at` - Quand créé

**Nombre de lignes:** ~10-50 morceaux (profil de l'entreprise divisé en morceaux)

---

`intent_translations`

**Objectif:** Mappe les intentions de leads entre l'anglais et le français.

**Ce qu'elle fait:**

- Assure une classification d'intention cohérente entre les langues
- Fournit des traductions d'intention Anglais ↔ Français
- Alimente les analyses bilingues
- Maintient la taxonomie des intentions

**Champs clés:**

- `id` - UUID de traduction
- `intent_en` - Intention en anglais
- `intent_fr` - Intention en français
- `category` - Catégorie d'intention
- `created_at` - Quand ajouté

**Nombre de lignes:** ~20-50 mappings d'intentions



## Statistiques de Base de Données

**Total de Tables: 24**

**Par Catégorie:**

- Gestion des Clients: 1 table
- Traitement des Leads: 3 tables
- Intelligence IA & Apprentissage: 5 tables
- Découverte de Prospects: 4 tables
- Automatisation des Approches: 5 tables
- Système de Traduction: 2 tables
- Traitement en Arrière-Plan: 1 table
- Tables de Support: 3 tables

**Répartition du Stockage:**

- **Données de Leads:** 70-80% (lead\_memory, lead\_actions, lead\_notes)
- **Données de Prospects:** 10-15% (prospect\_candidates, outreach\_emails)
- **Données d'Apprentissage:** 5-10% (feedback\_tracking, performance\_metrics, prompt\_performance)
- **Données de Support:** 5% (caches, journaux, traductions)

**Taux de Croissance:**

- **Croissance Élevée:** lead\_memory, lead\_actions, feedback\_tracking, performance\_metrics
- **Croissance Moyenne:** prospect\_candidates, outreach\_emails, prompt\_performance
- **Croissance Faible:** prompt\_registry, translation\_dictionary, intent\_translations

- **Statique/Lent:** email\_templates, growth\_brain (1 par client)



## Sécurité & Contrôle d'Accès

**Row Level Security (RLS):** Activé sur TOUTES les tables

### Types de Politiques:

1. **Isolation Client:** Les clients ne peuvent voir que leurs propres données
2. **Rôle de Service:** Accès complet pour les opérations système
3. **Lecture Publique:** Certaines tables permettent la lecture publique pour les démos
4. **Écriture Authentifiée:** La plupart des écritures nécessitent une authentification

### Fonctionnalités de Sécurité Clés:

- Authentification par clé API
- Hashes de mots de passe chiffrés
- Isolation des données spécifiques au client
- Pistes d'audit (lead\_actions, integration\_logs)
- Marquage automatique des données de test (colonne `is_test` )



## Fonctionnalités Clés de la Base de Données

### 1. Architecture Multi-Locataire

- Chaque table lie à `client_id`
- RLS applique une isolation stricte des données
- Les clients ne voient que leurs propres leads/actions/notes

### 2. Infrastructure d'Apprentissage Adaptatif

- `feedback_tracking` + `performance_metrics` = Boucles d'apprentissage
- `prompt_registry` + `prompt_performance` = Optimisation des prompts
- `growth_brain` = Mises à jour d'intelligence hebdomadaires

### 3. Flexibilité JSONB

- Champs `metadata` dans la plupart des tables pour l'extensibilité
- `context_data` pour le système d'apprentissage
- `predictive_insights` pour les recommandations bilingues
- Tableaux de suivi d'historique (tone\_history, confidence\_history)

## 4. Support Bilingue

- Colonnes `notes_en` + `notes_fr`
- Colonnes `language` partout
- Table de mapping `intent_translations`
- Insights bilingues dans `growth_brain`

## 5. Optimisations de Performance

- Indexation complète sur toutes les colonnes de requête
- Index de timestamp pour le tri
- Index composites pour les requêtes courantes
- Index de clé étrangère pour les jointures



## Flux de Données

### Flux de Capture de Leads:

```
Formulaire Client → /api/lead → lead_memory (insertion)
                                → Enrichissement IA
                                → lead_actions (log "created")
                                → feedback_tracking (log prédiction)
                                → Réponse Email
```

### Flux de Boucle d'Apprentissage:

```
Prédiction IA → feedback_tracking (log)
                → performance_metrics (log vitesse, précision)
                → Résultat Réel
                → Comparer & Noter
                → prompt_performance (log résultat)
                → Analyse Hebdomadaire
                → growth_brain (mettre à jour les insights)
                → Meilleures Prédictions Futures
```

### Flux de Découverte de Prospects:

```
Définition ICP → Recherche PDL/Google/Apollo
                → prospect_candidates (insertion 16+ quotidien)
                → calcul_automation_need_score
                → prospect_outreach_log (email envoyé)
                → outreach_tracking (événements d'engagement)
                → prospect_industry_performance (mettre à jour stats)
```

**Documentation Préparée Par:** Équipe d'Infrastructure IA de Croissance

**Type de Base de Données:** PostgreSQL (Supabase Principal, Neon Failover)

**Total de Tables:** 24

**Dernière Mise à Jour:** 22 octobre 2025