

# GUIDE D'ADMINISTRATION

## Système de Plan de Classe

*Configuration, Gestion des Utilisateurs et Maintenance*

Application :Plan de Classe
Version :1.0.0
Dernière MAJ :08/12/2025
Base de données :Supabase (PostgreSQL)
Hébergement :Vercel

### Table des Matières

1. Configuration Initiale
2. Gestion des Établissements
3. Gestion des Utilisateurs
4. Structure de la Base de Données
5. Identifiants de Test
6. Dépannage
7. Maintenance et Sauvegarde

# 1. Configuration Initiale

## 1.1 Prérequis

- Compte Supabase avec un projet créé
- Accès à l'éditeur SQL de Supabase
- Scripts SQL de migration (fournis dans le dépôt)
- Variables d'environnement configurées (SUPABASE\_URL, SUPABASE\_ANON\_KEY)

## 1.2 Exécution des Scripts SQL

Les scripts doivent être exécutés **dans l'ordre suivant** dans l'éditeur SQL de Supabase :

#	Script	Description
1	001_create_schema.sql	Structure de base (tables, types, RLS)
2	002_profile_trigger.sql	Trigger pour les profils
3	003_create_rooms.sql	Tables pour les salles
4	004_create_subrooms.sql	Tables pour les sous-salles
5	005_rls_policies.sql	Politiques de sécurité RLS
6	006_create_test_users.sql	Utilisateurs de test

## 1.3 Vérification de la Création

Après l'exécution des scripts, vérifiez dans Supabase :

- **Table Editor** : Tables visibles : establishments, profiles, students, teachers, classes, rooms, sub\_rooms
- **Authentication > Users** : 8 utilisateurs de test créés
- **Database > Policies** : Politiques RLS actives sur chaque table

⚠️ Si une erreur survient, vérifiez que vous exécutez les scripts dans l'ordre et que chaque script s'exécute sans erreur avant de passer au suivant.

## 2. Gestion des Établissements

### 2.1 Ajouter un Nouvel Établissement

```
-- Exemple : Ajouter le lycée Jean Moulin
INSERT INTO public.establishments (name, code)
VALUES ('JEAN-MOULIN 75015', 'jm001');
```

#### Convention de nommage

Champ	Format	Exemple
name	NOM-ETABLISSEMENT CODE_POSTAL	"ST-MARIE 14000"
code	[initiales][número] (unique)	"stm001", "vh001", "jm001"

i Le code doit être unique et est utilisé lors de la connexion pour identifier l'établissement.

### 2.2 Modifier un Établissement

```
-- Modifier le nom d'un établissement
UPDATE public.establishments
SET name = 'NOUVEAU-NOM 12345'
WHERE code = 'stm001';
```

### 2.3 Supprimer un Établissement

X ATTENTION : La suppression d'un établissement supprime TOUTES les données associées (utilisateurs, salles, sous-salles, etc.) en cascade.

```
-- Supprimer un établissement et toutes ses données
DELETE FROM public.establishments WHERE code = 'stm001';
```

## 3. Gestion des Utilisateurs

### 3.1 Architecture Utilisateur

Chaque utilisateur est composé de **deux éléments** :

Composant	Description
profiles	Table principale — Informations de profil, rôle, établissement, identifiants
students / teachers	Tables d'extension — Informations spécifiques au rôle (classe, matière...)

### 3.2 Méthode 1 : Création via SQL

**Recommandé pour** : tests, migrations, création en masse

Exemple : Créer un utilisateur Vie Scolaire

```
-- Créer un utilisateur vie-scolaire
DO $$
DECLARE
    v_establishment_id UUID;
BEGIN
    -- 1. Récupérer l'ID de l'établissement
    SELECT id INTO v_establishment_id
    FROM public.establissemnts WHERE code = 'stm001';

    -- 2. Créer le profil (avec hash bcrypt)
    INSERT INTO public.profiles (
        establishment_id, role, username, password_hash,
        first_name, last_name, email, can_create_subrooms
    ) VALUES (
        v_establishment_id,
        'vie-scolaire',
        'marie.martin',
        crypt('MotDePasse123!', gen_salt('bf')),
        'Marie', 'Martin',
        'marie.martin@stmarie.fr',
        true
    );
    RAISE NOTICE 'Utilisateur créé avec succès!';
END $$;
```

**Exemple : Créer un Élève Délégué**

```
-- Créer un élève délégué (profil + extension students)
DO $$
DECLARE
    v_establishment_id UUID;
    v_class_id UUID;
    v_profile_id UUID;
BEGIN
    -- 1. Récupérer les IDs
    SELECT id INTO v_establishment_id
    FROM establishments WHERE code = 'stm001';

    SELECT id INTO v_class_id
    FROM classes WHERE name = '6ème A'
    AND establishment_id = v_establishment_id;

    -- 2. Créer le profil
    INSERT INTO profiles (
        establishment_id, role, username, password_hash,
        first_name, last_name, can_create_subrooms
    ) VALUES (
        v_establishment_id, 'delegue', 'jean.dupont',
        crypt('Delegue2024!', gen_salt('bf')),
        'Jean', 'Dupont', false
    ) RETURNING id INTO v_profile_id;

    -- 3. Créer l'extension students
    INSERT INTO students (
        profile_id, establishment_id, class_id,
        first_name, last_name, role, username, password_hash
    ) VALUES (
        v_profile_id, v_establishment_id, v_class_id,
        'Jean', 'Dupont', 'delegue', 'jean.dupont',
        crypt('Delegue2024!', gen_salt('bf'))
    );
END $$;
```

**3.3 Méthode 2 : Crédation via Interface****Recommandé pour :** production, utilisateurs individuels

1. Se connecter en tant que Vie Scolaire
2. Accéder à la section "Élèves" ou "Professeurs"
3. Cliquer sur "Ajouter"
4. Remplir le formulaire (nom, prénom, email, classe/matière)
5. Valider — Les identifiants sont générés automatiquement

✓ Le module user-management.ts génère automatiquement un username unique et un mot de passe sécurisé.

**3.4 Modifier un Utilisateur**

```
-- Changer le rôle d'un utilisateur
UPDATE profiles
SET role = 'vie-scolaire', can_create_subrooms = true
WHERE username = 'prof.stmarie';

-- Changer le mot de passe
UPDATE profiles
SET password_hash = crypt('NouveauMDP123!', gen_salt('bf'))
WHERE username = 'prof.stmarie';

-- Changer l'établissement
UPDATE profiles
SET establishment_id = (SELECT id FROM establishments WHERE code = 'vh001')
WHERE username = 'prof.stmarie';
```

**3.5 Supprimer un Utilisateur**

```
-- Supprimer un utilisateur (cascade sur students/teachers)
DELETE FROM profiles WHERE username = 'jean.dupont';
```

## 4. Structure de la Base de Données

### 4.1 Vue d'ensemble des tables

Table	Description
establishments	Établissements scolaires
profiles	TABLE PRINCIPALE — Utilisateurs et authentification
students	Extension élèves (délégués, éco-délégués)
teachers	Extension professeurs (matière, classes)
classes	Classes de l'établissement
rooms	Salles de classe
room_assignments	Affectations classe/salle/professeur
sub_rooms	Sous-salles (versions personnalisées)
audit_logs	Historique des modifications
archived_sub_rooms	Sous-salles archivées

### 4.2 Table profiles (détail)

Colonne	Type	Description
id	UUID	Clé primaire
establishment_id	UUID (FK)	→ establishments.id
role	user_role	vie-scolaire   professeur   delegue   eco-delegue
username	TEXT	Identifiant de connexion (unique)
password_hash	TEXT	Mot de passe haché (bcrypt)
first_name	TEXT	Prénom
last_name	TEXT	Nom
can_create_subrooms	BOOLEAN	Autorisation créer des sous-salles

### 4.3 Table sub\_rooms (détail)

Colonne	Type	Description
id	UUID	Clé primaire
room_assignment_id	UUID (FK)	→ room_assignments.id
name	TEXT	Nom de la sous-salle
type	sub_room_type	temporary   permanent
start_date	DATE	Date début (si temporary)
end_date	DATE	Date fin (si temporary)
seat_assignments	JSONOB	Placement des élèves
created_by	UUID (FK)	→ profiles.id (créateur)
creator_role	TEXT	Rôle du créateur (dénormalisé)

## 6. Identifiants de Test

Établissements : stm001 (ST-MARIE 14000), vh001 (VICTOR-HUGO 18760)

### 6.1 Comptes Vie Scolaire

Établissement	Identifiant	Mot de passe	Notes
stm001	vs.stmarie	VieScol2024!	Standard
stm001	admin.vs.stm	cpdc003	Admin migré
vh001	vs.vhugo	VieScol2024!	Standard

### 6.2 Comptes Professeur

Établissement	Identifiant	Mot de passe	Matière
stm001	prof.stmarie	Prof2024!	-
stm001	prof.dupont	password123	Maths
stm001	prof.martin	password123	Français
vh001	prof.vhugo	Prof2024!	-
stm001	admin.prof.stm	cpdc002	Admin migré

### 6.3 Comptes Délégué

Établissement	Identifiant	Mot de passe	Statut
stm001	del.stmarie	Delegue2024!	✓
stm001	admin.delegue.stm	cpdc001	✓
vh001	del.vhugo	Delegue2024!	✓

⚠️ Ces comptes sont destinés aux tests uniquement. En production, créez des comptes avec des mots de passe uniques.

## 6. Dépannage

### 6.1 Problème : "Impossible de se connecter"

8. Vérifier que le **code établissement** existe dans la table establishments
9. Vérifier que l'utilisateur existe dans profiles avec le bon username
10. Vérifier que establishment\_id dans profiles correspond au code saisi
11. Vérifier le **password\_hash** (tester avec une requête SQL)

```
-- Vérifier les credentials
SELECT id, username, role, establishment_id
FROM profiles
WHERE username = 'vs.stmarie'
    AND password_hash = crypt('VieScol2024!', password_hash);
```

### 6.2 Problème : "Erreur RLS"

Les politiques RLS (Row Level Security) limitent l'accès aux données. Si un utilisateur ne peut pas voir/modifier des données :

Rôle	Accès autorisé
vie-scolaire	Toutes les données de son établissement (lecture/écriture)
professeur	Ses classes, ses élèves, ses salles (lecture) + créer/modifier sous-salles
delegue	Sa classe, ses professeurs autorisés + créer sous-salles si autorisé

### 6.3 Problème : Page blanche / Erreur 500

- Vérifier les logs Vercel (Functions > View Function Logs)
- Vérifier les variables d'environnement SUPABASE\_URL et SUPABASE\_ANON\_KEY
- Vérifier la console navigateur (F12 > Console)
- Redéployer l'application sur Vercel

### 6.4 Réinitialiser la Base de Données

X ATTENTION : Cette opération supprime TOUTES les données de manière irréversible.

```
-- Réinitialisation complète (DESTRUCTIF)
DROP SCHEMA public CASCADE;
CREATE SCHEMA public;
```

```
-- Puis réexécuter tous les scripts dans l'ordre :
-- 001, 002, 003, 004, 005, 006
```

## 7. Maintenance et Sauvegarde

### 7.1 Sauvegarde de la base de données

Supabase propose des sauvegardes automatiques quotidiennes (plan Pro). Pour une sauvegarde manuelle :

12. Aller dans Settings > Database > Backups
13. Cliquer sur "Create backup"
14. Télécharger le fichier .sql

```
-- Export manuel via pg_dump (si accès direct)
pg_dump -h db.xxxx.supabase.co -U postgres -d postgres > backup.sql
```

### 7.2 Nettoyage périodique

```
-- Archiver les sous-salles expirées
SELECT archive_expired_subrooms();

-- Supprimer les logs d'audit > 1 an
DELETE FROM audit_logs
WHERE created_at < NOW() - INTERVAL '1 year';

-- Vérifier l'espace disque utilisé
SELECT pg_size_pretty(pg_database_size('postgres'));
```

### 7.3 Monitoring

Élément	URL / Action
Supabase Dashboard	<a href="https://app.supabase.com/project/[id]">https://app.supabase.com/project/[id]</a>
Vercel Logs	<a href="https://vercel.com/[team]/[project]/logs">https://vercel.com/[team]/[project]/logs</a>
Status Supabase	<a href="https://status.supabase.com">https://status.supabase.com</a>
Status Vercel	<a href="https://www.vercel-status.com">https://www.vercel-status.com</a>

### 7.4 Contacts Support

- **Documentation Supabase** : <https://supabase.com/docs>
- **Row Level Security** : <https://supabase.com/docs/guides/auth/row-level-security>
- **SQL Editor** : <https://supabase.com/docs/guides/database/overview>

— Fin du guide d'administration —

Plan de Classe — v1.0.0 — 08/12/2025