1. Supabase Schemas

DROP TABLE IF EXISTS arattai\_message\_logs CASCADE;

DROP TABLE IF EXISTS arattai\_scheduled\_messages CASCADE;

DROP TABLE IF EXISTS arattai\_broadcast\_campaigns CASCADE;

DROP TABLE IF EXISTS arattai\_automation\_rules CASCADE;

DROP TABLE IF EXISTS arattai\_templates CASCADE;

DROP TABLE IF EXISTS arattai\_contacts CASCADE;

DROP TABLE IF EXISTS arattai\_settings CASCADE;

DROP TABLE IF EXISTS vipudev\_ai CASCADE;

DROP TABLE IF EXISTS cbse\_registration CASCADE;

DROP TABLE IF EXISTS id\_card\_generator CASCADE;

DROP TABLE IF EXISTS fee\_tally\_integration CASCADE;

DROP TABLE IF EXISTS timesubbehave\_ai\_premium CASCADE;

DROP TABLE IF EXISTS vipu\_ai CASCADE;

DROP TABLE IF EXISTS whatsapp\_alerts CASCADE;

DROP TABLE IF EXISTS system\_settings CASCADE;

DROP TABLE IF EXISTS school\_events CASCADE;

DROP TABLE IF EXISTS substitutions CASCADE;

DROP TABLE IF EXISTS behavior\_tracking CASCADE;

DROP TABLE IF EXISTS timetable CASCADE;

DROP TABLE IF EXISTS attendance CASCADE;

DROP TABLE IF EXISTS fee\_payments CASCADE;

DROP TABLE IF EXISTS fee\_structure CASCADE;

DROP TABLE IF EXISTS subjects CASCADE;

DROP TABLE IF EXISTS students CASCADE;

DROP TABLE IF EXISTS classes CASCADE;

DROP TABLE IF EXISTS teachers CASCADE;

DROP TABLE IF EXISTS user\_profiles CASCADE;

DROP TABLE IF EXISTS schools CASCADE;

DROP TABLE IF EXISTS school\_modules CASCADE;

DROP TABLE IF EXISTS student\_distribution CASCADE;

DROP TABLE IF EXISTS report\_tracker CASCADE;

DROP TABLE IF EXISTS question\_paper\_generation CASCADE;

DROP TABLE IF EXISTS question\_extractor CASCADE;

DROP TABLE IF EXISTS voice\_to\_text CASCADE;

DROP TABLE IF EXISTS admission\_management CASCADE;

DROP TABLE IF EXISTS pdf\_tools CASCADE;

DROP TABLE IF EXISTS audit\_logs CASCADE;

DROP TYPE IF EXISTS user\_role CASCADE;

DROP TYPE IF EXISTS attendance\_status CASCADE;

DROP TYPE IF EXISTS message\_status CASCADE;

DROP TYPE IF EXISTS arattai\_message\_category CASCADE;

DROP TYPE IF EXISTS arattai\_send\_type CASCADE;

DROP TYPE IF EXISTS arattai\_priority CASCADE;

-- Step 1: Drop existing ENUM types (if any) to avoid conflicts

DROP TYPE IF EXISTS user\_role CASCADE;

DROP TYPE IF EXISTS attendance\_status CASCADE;

DROP TYPE IF EXISTS message\_status CASCADE;

DROP TYPE IF EXISTS arattai\_message\_category CASCADE;

DROP TYPE IF EXISTS arattai\_send\_type CASCADE;

DROP TYPE IF EXISTS arattai\_priority CASCADE;

-- Step 1: Drop existing ENUM types (if any) to avoid conflicts

DROP TYPE IF EXISTS user\_role CASCADE;

DROP TYPE IF EXISTS attendance\_status CASCADE;

DROP TYPE IF EXISTS message\_status CASCADE;

DROP TYPE IF EXISTS arattai\_message\_category CASCADE;

DROP TYPE IF EXISTS arattai\_send\_type CASCADE;

DROP TYPE IF EXISTS arattai\_priority CASCADE;

-- Step 2: Create ENUM types (clean ASCII only)

CREATE TYPE user\_role AS ENUM ('super\_admin', 'school\_admin', 'teacher', 'parent', 'student');

CREATE TYPE attendance\_status AS ENUM ('present', 'absent', 'late', 'leave');

CREATE TYPE message\_status AS ENUM ('pending', 'sent', 'delivered', 'read', 'failed', 'cancelled', 'processing');

CREATE TYPE arattai\_message\_category AS ENUM (

    'attendance', 'academic', 'events', 'behavior', 'fee\_due',

    'exam', 'holiday', 'admission', 'emergency', 'other'

);

CREATE TYPE arattai\_send\_type AS ENUM ('automated', 'manual', 'scheduled', 'broadcast');

CREATE TYPE arattai\_priority AS ENUM ('low', 'medium', 'high', 'urgent');

-- Step 3: Enable UUID extension

CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

-- Step 4: Create core tables

-- Schools table

CREATE TABLE schools (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    name VARCHAR(255) NOT NULL,

    created\_at TIMESTAMPTZ DEFAULT NOW(),

    updated\_at TIMESTAMPTZ DEFAULT NOW()

);

CREATE TABLE user\_profiles (

    id UUID PRIMARY KEY REFERENCES auth.users(id) ON DELETE CASCADE,

    school\_id UUID REFERENCES schools(id) ON DELETE SET NULL,

    role user\_role NOT NULL,

    first\_name VARCHAR(100),

    last\_name VARCHAR(100),

    email VARCHAR(255) UNIQUE,

    is\_active BOOLEAN DEFAULT true,

    created\_at TIMESTAMPTZ DEFAULT NOW(),

    updated\_at TIMESTAMPTZ DEFAULT NOW()

);

-- Teachers

CREATE TABLE teachers (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE CASCADE UNIQUE NOT NULL,

    teacher\_id VARCHAR(50) NOT NULL,

    UNIQUE(school\_id, teacher\_id)

);

-- Classes

CREATE TABLE classes (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    name VARCHAR(50) NOT NULL,

    section VARCHAR(10) NOT NULL,

    class\_teacher\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL,

    academic\_year VARCHAR(10) NOT NULL,

    UNIQUE(school\_id, name, section, academic\_year)

);

-- Subjects

CREATE TABLE public.subjects (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES public.schools(id) ON DELETE CASCADE,

    name VARCHAR(255) NOT NULL,

    code VARCHAR(50),

    description TEXT,

    created\_at TIMESTAMPTZ DEFAULT NOW(),

    updated\_at TIMESTAMPTZ DEFAULT NOW()

);

-- Students

CREATE TABLE students (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    admission\_number VARCHAR(50),

    class\_id UUID REFERENCES classes(id) ON DELETE SET NULL,

    student\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL UNIQUE,

    parent\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL,

    UNIQUE(school\_id, admission\_number)

);

-- 7. Fee Structure

CREATE TABLE fee\_structure (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    class\_id UUID REFERENCES classes(id) ON DELETE CASCADE NOT NULL,

    fee\_type VARCHAR(50) NOT NULL,

    amount DECIMAL(10,2) NOT NULL,

    academic\_year VARCHAR(10) NOT NULL,

    UNIQUE(school\_id, class\_id, fee\_type, academic\_year)

);

-- 8. Fee Payments

CREATE TABLE fee\_payments (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    student\_id UUID REFERENCES students(id) ON DELETE CASCADE NOT NULL,

    amount\_paid DECIMAL(10,2) NOT NULL

);

-- 9. Attendance

CREATE TABLE attendance (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    student\_id UUID REFERENCES students(id) ON DELETE CASCADE NOT NULL,

    status attendance\_status NOT NULL

);

-- 10. Timetable

CREATE TABLE timetable (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    class\_id UUID REFERENCES classes(id) ON DELETE CASCADE NOT NULL,

    teacher\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL,

    day\_of\_week INTEGER NOT NULL,

    period\_number INTEGER NOT NULL,

    UNIQUE(school\_id, class\_id, day\_of\_week, period\_number)

);

-- 11. Behavior Tracking

CREATE TABLE behavior\_tracking (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    student\_id UUID REFERENCES students(id) ON DELETE CASCADE NOT NULL,

    reported\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 12. Substitution Log

CREATE TABLE substitutions (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    absent\_teacher\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL,

    substitute\_teacher\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 13. Student Distribution

CREATE TABLE student\_distribution (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    class\_id UUID REFERENCES classes(id) ON DELETE CASCADE

);

-- 14. Report Tracker

CREATE TABLE report\_tracker (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    student\_id UUID REFERENCES students(id) ON DELETE CASCADE,

    report\_data JSONB

);

-- 15. Question Paper Generation

CREATE TABLE question\_paper\_generation (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    generated\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 16. Question Extractor

CREATE TABLE question\_extractor (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    extracted\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 17. Voice to Text

CREATE TABLE voice\_to\_text (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL

);

ALTER TABLE public.voice\_to\_text

ADD COLUMN transcribed\_by\_user\_profile\_id UUID REFERENCES public.user\_profiles(id) ON DELETE SET NULL;

-- 18. Admission Management

CREATE TABLE admission\_management (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL

);

-- 19. PDF Tools

CREATE TABLE pdf\_tools (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES schools(id) ON DELETE CASCADE,

    used\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 20. School Event Log

CREATE TABLE school\_events (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES schools(id) ON DELETE CASCADE

);

-- 21. WhatsApp Alerts

CREATE TABLE whatsapp\_alerts (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES schools(id) ON DELETE CASCADE

);

-- 22. Vipu AI

CREATE TABLE vipu\_ai (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES schools(id) ON DELETE CASCADE

);

-- 23. Timesubbehave AI Premium

CREATE TABLE timesubbehave\_ai\_premium (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES schools(id) ON DELETE CASCADE,

    generated\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 24. Fee Tally Integration

CREATE TABLE fee\_tally\_integration (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES schools(id) ON DELETE CASCADE

);

-- 25. ID Card Generator

CREATE TABLE id\_card\_generator (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID NOT NULL REFERENCES schools(id) ON DELETE CASCADE,

    requested\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 26. CBSE Registration

CREATE TABLE cbse\_registration (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    submitted\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 27. Vipudev AI

CREATE TABLE vipudev\_ai (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL

);

-- 28. School Modules

CREATE TABLE school\_modules (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    module\_name VARCHAR(100) NOT NULL,

    is\_enabled BOOLEAN DEFAULT TRUE,

    UNIQUE(school\_id, module\_name)

);

-- 29. System Settings

CREATE TABLE system\_settings (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE,

    setting\_key VARCHAR(255) NOT NULL,

    setting\_value TEXT,

    UNIQUE(school\_id, setting\_key)

);

-- 30. Arattai Templates

CREATE TABLE arattai\_templates (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE,

    template\_category arattai\_message\_category NOT NULL,

    template\_type arattai\_send\_type NOT NULL,

    message\_content TEXT NOT NULL

);

-- 31. Arattai Contacts

CREATE TABLE arattai\_contacts (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE CASCADE,

    phone\_number VARCHAR(20) NOT NULL,

    consent\_given BOOLEAN DEFAULT FALSE,

    is\_opted\_out BOOLEAN DEFAULT FALSE,

    UNIQUE(school\_id, phone\_number)

);

-- 32. Arattai Scheduled Messages

CREATE TABLE arattai\_scheduled\_messages (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    template\_id UUID REFERENCES arattai\_templates(id) ON DELETE SET NULL,

    arattai\_contact\_id UUID REFERENCES arattai\_contacts(id) ON DELETE SET NULL,

    created\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 33. Arattai Broadcast Campaigns

CREATE TABLE arattai\_broadcast\_campaigns (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    template\_id UUID REFERENCES arattai\_templates(id) ON DELETE SET NULL

);

-- 34. Arattai Automation Rules

CREATE TABLE arattai\_automation\_rules (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    template\_id UUID REFERENCES arattai\_templates(id) ON DELETE SET NULL,

    created\_by\_user\_profile\_id UUID REFERENCES user\_profiles(id) ON DELETE SET NULL

);

-- 35. Arattai Message Logs

CREATE TABLE arattai\_message\_logs (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL,

    arattai\_contact\_id UUID REFERENCES arattai\_contacts(id) ON DELETE SET NULL,

    arattai\_external\_message\_id VARCHAR(100) UNIQUE

);

-- 36. Arattai Settings

CREATE TABLE arattai\_settings (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE NOT NULL UNIQUE

);

-- 37. Audit Logs (General table for all modules)

CREATE TABLE audit\_logs (

    id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

    school\_id UUID REFERENCES schools(id) ON DELETE CASCADE,

    user\_id UUID,

    action VARCHAR(100) NOT NULL,

    entity\_id UUID,

    details JSONB,

    created\_at TIMESTAMPTZ DEFAULT NOW()

);

2. RLS SQL

-- 🔥 FINAL RLS SCRIPT: USING EXPLICIT PARENTHESES FOR 100% PARSING SUCCESS 🔥

------------------------------------------------------------------------------------

-- 1. HELPER FUNCTIONS

------------------------------------------------------------------------------------

CREATE OR REPLACE FUNCTION get\_user\_id()

RETURNS UUID

LANGUAGE sql STABLE

AS $$

  SELECT auth.uid();

$$;

CREATE OR REPLACE FUNCTION get\_user\_school\_id()

RETURNS UUID

LANGUAGE sql STABLE SECURITY DEFINER

AS $$

  SELECT school\_id FROM public.user\_profiles WHERE id = get\_user\_id();

$$;

CREATE OR REPLACE FUNCTION get\_user\_role()

RETURNS user\_role

LANGUAGE sql STABLE SECURITY DEFINER

AS $$

  SELECT role FROM public.user\_profiles WHERE id = get\_user\_id();

$$;

------------------------------------------------------------------------------------

-- 2. ENABLE RLS ON ALL RELEVANT TABLES

------------------------------------------------------------------------------------

ALTER TABLE public.schools ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.user\_profiles ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.teachers ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.classes ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.students ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.subjects ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.fee\_structure ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.fee\_payments ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.attendance ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.timetable ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.behavior\_tracking ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.substitutions ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.student\_distribution ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.report\_tracker ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.question\_paper\_generation ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.question\_extractor ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.voice\_to\_text ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.admission\_management ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.pdf\_tools ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.school\_events ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.whatsapp\_alerts ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.vipu\_ai ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.timesubbehave\_ai\_premium ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.fee\_tally\_integration ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.id\_card\_generator ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.cbse\_registration ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.vipudev\_ai ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.school\_modules ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.system\_settings ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.arattai\_templates ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.arattai\_contacts ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.arattai\_scheduled\_messages ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.arattai\_broadcast\_campaigns ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.arattai\_automation\_rules ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.arattai\_message\_logs ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.arattai\_settings ENABLE ROW LEVEL SECURITY;

ALTER TABLE public.audit\_logs ENABLE ROW LEVEL SECURITY;

------------------------------------------------------------------------------------

-- 3. CORE IDENTITY POLICIES

------------------------------------------------------------------------------------

DROP POLICY IF EXISTS "Schools: Select own school or Super Admin" ON public.schools;

CREATE POLICY "Schools: Select own school or Super Admin" ON public.schools FOR SELECT TO authenticated

USING ((id = get\_user\_school\_id()) OR (get\_user\_role() = 'super\_admin'));

DROP POLICY IF EXISTS "Schools: Manage own school (Admin/Super)" ON public.schools;

CREATE POLICY "Schools: Manage own school (Admin/Super)" ON public.schools FOR ALL TO authenticated

USING ((id = get\_user\_school\_id()) AND (get\_user\_role() IN ('school\_admin', 'super\_admin')))

WITH CHECK ((id = get\_user\_school\_id()) AND (get\_user\_role() IN ('school\_admin', 'super\_admin')));

-- User Profiles

-- Check and enable RLS if not enabled

ALTER TABLE public.user\_profiles ENABLE ROW LEVEL SECURITY;

-- Drop old policy if exists

DROP POLICY IF EXISTS "User Profiles: Access Control" ON public.user\_profiles;

-- Create policy

CREATE POLICY "User Profiles: Access Control" ON public.user\_profiles FOR ALL TO authenticated

USING (

  (id = get\_user\_id())

  OR ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('school\_admin', 'super\_admin')))

  OR (get\_user\_role() = 'super\_admin')

)

WITH CHECK (

  (id = get\_user\_id())

  OR ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('school\_admin', 'super\_admin')))

  OR (get\_user\_role() = 'super\_admin')

);

-- Teachers

DROP POLICY IF EXISTS "Teachers: Select" ON public.teachers;

CREATE POLICY "Teachers: Select" ON public.teachers FOR SELECT TO authenticated

USING (

  (school\_id = get\_user\_school\_id())

  AND (

    (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')) -- Admins/Teachers see all

    OR (user\_profile\_id = get\_user\_id()) -- Self-access

  )

);

DROP POLICY IF EXISTS "Teachers: Manage" ON public.teachers;

CREATE POLICY "Teachers: Manage" ON public.teachers FOR ALL TO authenticated

USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin')))

WITH CHECK (school\_id = get\_user\_school\_id());

-- Classes

DROP POLICY IF EXISTS "Classes: Select" ON public.classes;

CREATE POLICY "Classes: Select" ON public.classes FOR SELECT TO authenticated

USING (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "Classes: Manage" ON public.classes;

CREATE POLICY "Classes: Manage" ON public.classes FOR ALL TO authenticated

USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin')))

WITH CHECK (school\_id = get\_user\_school\_id());

-- Students

DROP POLICY IF EXISTS "Students: Select" ON public.students;

CREATE POLICY "Students: Select" ON public.students FOR SELECT TO authenticated

USING (

  (school\_id = get\_user\_school\_id())

  AND (

    (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')) -- Admins/Teachers see all

    OR (parent\_user\_profile\_id = get\_user\_id())

    OR (student\_user\_profile\_id = get\_user\_id())

  )

);

DROP POLICY IF EXISTS "Students: Manage" ON public.students;

CREATE POLICY "Students: Manage" ON public.students FOR ALL TO authenticated

USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin')))

WITH CHECK (school\_id = get\_user\_school\_id());

-- Fee Structure

DROP POLICY IF EXISTS "Fee Structure: Select" ON public.fee\_structure;

CREATE POLICY "Fee Structure: Select" ON public.fee\_structure FOR SELECT TO authenticated

USING (

  (school\_id = get\_user\_school\_id())

  AND (

    (get\_user\_role() IN ('super\_admin', 'school\_admin'))

    OR EXISTS (

      SELECT 1 FROM public.students s

      WHERE (s.parent\_user\_profile\_id = get\_user\_id())

      AND (s.class\_id = fee\_structure.class\_id)

    )

  )

);

DROP POLICY IF EXISTS "Fee Structure: Manage" ON public.fee\_structure;

CREATE POLICY "Fee Structure: Manage" ON public.fee\_structure FOR ALL TO authenticated

USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin')))

WITH CHECK (school\_id = get\_user\_school\_id());

-- Fee Payments

DROP POLICY IF EXISTS "Fee Payments: Select" ON public.fee\_payments;

CREATE POLICY "Fee Payments: Select" ON public.fee\_payments FOR SELECT TO authenticated

USING (

  (school\_id = get\_user\_school\_id())

  AND (

    (get\_user\_role() IN ('super\_admin', 'school\_admin'))

    OR EXISTS (

      SELECT 1 FROM public.students s

      WHERE (s.parent\_user\_profile\_id = get\_user\_id())

      AND (s.id = fee\_payments.student\_id)

    )

  )

);

DROP POLICY IF EXISTS "Fee Payments: Manage" ON public.fee\_payments;

CREATE POLICY "Fee Payments: Manage" ON public.fee\_payments FOR ALL TO authenticated

USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin')))

WITH CHECK (school\_id = get\_user\_school\_id());

-- Attendance

DROP POLICY IF EXISTS "Attendance: Select" ON public.attendance;

CREATE POLICY "Attendance: Select" ON public.attendance FOR SELECT TO authenticated

USING (

  (school\_id = get\_user\_school\_id())

  AND (

    (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher'))

    OR EXISTS (

      SELECT 1 FROM public.students s

      WHERE (s.parent\_user\_profile\_id = get\_user\_id())

      AND (s.id = attendance.student\_id)

    )

  )

);

DROP POLICY IF EXISTS "Attendance: Manage" ON public.attendance;

CREATE POLICY "Attendance: Manage" ON public.attendance FOR ALL TO authenticated

USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')))

WITH CHECK (school\_id = get\_user\_school\_id());

-- Behavior Tracking

DROP POLICY IF EXISTS "Behavior: Select" ON public.behavior\_tracking;

CREATE POLICY "Behavior: Select" ON public.behavior\_tracking FOR SELECT TO authenticated

USING (

  (school\_id = get\_user\_school\_id())

  AND (

    (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher'))

    OR EXISTS (

      SELECT 1 FROM public.students s

      WHERE (s.parent\_user\_profile\_id = get\_user\_id())

      AND (s.id = behavior\_tracking.student\_id)

    )

  )

);

DROP POLICY IF EXISTS "Behavior: Manage" ON public.behavior\_tracking;

CREATE POLICY "Behavior: Manage" ON public.behavior\_tracking FOR ALL TO authenticated

USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')))

WITH CHECK (school\_id = get\_user\_school\_id());

-- Timetable

DROP POLICY IF EXISTS "Timetable: Select" ON public.timetable;

CREATE POLICY "Timetable: Select" ON public.timetable FOR SELECT TO authenticated USING (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "Timetable: Manage" ON public.timetable;

CREATE POLICY "Timetable: Manage" ON public.timetable FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin'))) WITH CHECK (school\_id = get\_user\_school\_id());

-- AI & TOOLING POLICIES (Individual Policies for varying column names)

DROP POLICY IF EXISTS "RLS: question\_paper\_generation Select (Admin/Teacher)" ON public.question\_paper\_generation;

CREATE POLICY "RLS: question\_paper\_generation Select (Admin/Teacher)" ON public.question\_paper\_generation FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: question\_paper\_generation Manage (Admin/Creator)" ON public.question\_paper\_generation;

CREATE POLICY "RLS: question\_paper\_generation Manage (Admin/Creator)" ON public.question\_paper\_generation FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin') OR (generated\_by\_user\_profile\_id = get\_user\_id()))) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "RLS: question\_extractor Select (Admin/Teacher)" ON public.question\_extractor;

CREATE POLICY "RLS: question\_extractor Select (Admin/Teacher)" ON public.question\_extractor FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: question\_extractor Manage (Admin/Creator)" ON public.question\_extractor;

CREATE POLICY "RLS: question\_extractor Manage (Admin/Creator)" ON public.question\_extractor FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin') OR (extracted\_by\_user\_profile\_id = get\_user\_id()))) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "RLS: voice\_to\_text Select (Admin/Teacher)" ON public.voice\_to\_text;

CREATE POLICY "RLS: voice\_to\_text Select (Admin/Teacher)" ON public.voice\_to\_text FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: voice\_to\_text Manage (Admin/Creator)" ON public.voice\_to\_text;

CREATE POLICY "RLS: voice\_to\_text Manage (Admin/Creator)" ON public.voice\_to\_text FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin') OR (transcribed\_by\_user\_profile\_id = get\_user\_id()))) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "RLS: pdf\_tools Select (Admin/Teacher)" ON public.pdf\_tools;

CREATE POLICY "RLS: pdf\_tools Select (Admin/Teacher)" ON public.pdf\_tools FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: pdf\_tools Manage (Admin/Creator)" ON public.pdf\_tools;

CREATE POLICY "RLS: pdf\_tools Manage (Admin/Creator)" ON public.pdf\_tools FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin') OR (used\_by\_user\_profile\_id = get\_user\_id()))) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "RLS: vipu\_ai Select (Admin/Teacher)" ON public.vipu\_ai;

CREATE POLICY "RLS: vipu\_ai Select (Admin/Teacher)" ON public.vipu\_ai FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: vipu\_ai Manage (Admins/Super)" ON public.vipu\_ai;

CREATE POLICY "RLS: vipu\_ai Manage (Admins/Super)" ON public.vipu\_ai FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin'))) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "RLS: timesubbehave\_ai\_premium Select (Admin/Teacher)" ON public.timesubbehave\_ai\_premium;

CREATE POLICY "RLS: timesubbehave\_ai\_premium Select (Admin/Teacher)" ON public.timesubbehave\_ai\_premium FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: timesubbehave\_ai\_premium Manage (Admin/Creator)" ON public.timesubbehave\_ai\_premium;

CREATE POLICY "RLS: timesubbehave\_ai\_premium Manage (Admin/Creator)" ON public.timesubbehave\_ai\_premium FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin') OR (generated\_by\_user\_profile\_id = get\_user\_id()))) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "RLS: id\_card\_generator Select (Admin/Teacher)" ON public.id\_card\_generator;

CREATE POLICY "RLS: id\_card\_generator Select (Admin/Teacher)" ON public.id\_card\_generator FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: id\_card\_generator Manage (Admin/Creator)" ON public.id\_card\_generator;

CREATE POLICY "RLS: id\_card\_generator Manage (Admin/Creator)" ON public.id\_card\_generator FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin') OR (requested\_by\_user\_profile\_id = get\_user\_id()))) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "RLS: vipudev\_ai Select (Admin/Teacher)" ON public.vipudev\_ai;

CREATE POLICY "RLS: vipudev\_ai Select (Admin/Teacher)" ON public.vipudev\_ai FOR SELECT TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "RLS: vipudev\_ai Manage (Admins/Super)" ON public.vipudev\_ai;

CREATE POLICY "RLS: vipudev\_ai Manage (Admins/Super)" ON public.vipudev\_ai FOR ALL TO authenticated USING ((school\_id = get\_user\_school\_id()) AND (get\_user\_role() IN ('super\_admin', 'school\_admin'))) WITH CHECK (school\_id = get\_user\_school\_id());

-- GENERAL ADMIN/SYSTEM POLICIES (Batch Creation)

DO $$

DECLARE

    t TEXT;

    admin\_tables TEXT[] := ARRAY[

        'subjects', 'substitutions', 'student\_distribution',

        'admission\_management', 'school\_events',

        'whatsapp\_alerts', 'fee\_tally\_integration', 'cbse\_registration',

        'timetable', 'report\_tracker', 'audit\_logs'

    ];

BEGIN

    FOREACH t IN ARRAY admin\_tables

LOOP

    EXECUTE format('DROP POLICY IF EXISTS "RLS: %I Select (All Staff/Admins)" ON public.%I', t, t);

    EXECUTE format('

        CREATE POLICY "RLS: %I Select (All Staff/Admins)" ON public.%I FOR SELECT TO authenticated

        USING (school\_id = get\_user\_school\_id());

    ', t, t);

    EXECUTE format('DROP POLICY IF EXISTS "RLS: %I Manage (Admins/Super)" ON public.%I', t, t);

    EXECUTE format('

        CREATE POLICY "RLS: %I Manage (Admins/Super)" ON public.%I FOR ALL TO authenticated

        USING (school\_id = get\_user\_school\_id() AND get\_user\_role() IN (''super\_admin'', ''school\_admin''))

        WITH CHECK (school\_id = get\_user\_school\_id());

    ', t, t);

END LOOP;

END

$$ LANGUAGE plpgsql;

-- System Settings

DROP POLICY IF EXISTS "System Settings: Select (All Staff)" ON public.system\_settings;

CREATE POLICY "System Settings: Select (All Staff)" ON public.system\_settings FOR SELECT TO authenticated USING (school\_id = get\_user\_school\_id() OR school\_id IS NULL);

DROP POLICY IF EXISTS "System Settings: Manage (Super Admin Only)" ON public.system\_settings;

CREATE POLICY "System Settings: Manage (Super Admin Only)" ON public.system\_settings FOR ALL TO authenticated USING (get\_user\_role() = 'super\_admin') WITH CHECK (get\_user\_role() = 'super\_admin');

-- ARATTAI MESSAGING POLICIES

DROP POLICY IF EXISTS "Arattai Templates: Select" ON public.arattai\_templates;

CREATE POLICY "Arattai Templates: Select" ON public.arattai\_templates FOR SELECT TO authenticated USING (school\_id = get\_user\_school\_id() OR school\_id IS NULL);

DROP POLICY IF EXISTS "Arattai Templates: Manage" ON public.arattai\_templates;

CREATE POLICY "Arattai Templates: Manage" ON public.arattai\_templates FOR ALL TO authenticated USING (school\_id = get\_user\_school\_id() AND get\_user\_role() IN ('school\_admin', 'super\_admin')) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "Arattai Contacts: Select" ON public.arattai\_contacts;

CREATE POLICY "Arattai Contacts: Select" ON public.arattai\_contacts FOR SELECT TO authenticated

USING (school\_id = get\_user\_school\_id() AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher') OR user\_profile\_id = get\_user\_id()));

DROP POLICY IF EXISTS "Arattai Contacts: Manage" ON public.arattai\_contacts;

CREATE POLICY "Arattai Contacts: Manage" ON public.arattai\_contacts FOR ALL TO authenticated USING (school\_id = get\_user\_school\_id() AND get\_user\_role() IN ('school\_admin', 'super\_admin')) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "Arattai Scheduled: Select" ON public.arattai\_scheduled\_messages;

CREATE POLICY "Arattai Scheduled: Select" ON public.arattai\_scheduled\_messages FOR SELECT TO authenticated

USING (school\_id = get\_user\_school\_id() AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher') OR created\_by\_user\_profile\_id = get\_user\_id()));

DROP POLICY IF EXISTS "Arattai Scheduled: Manage" ON public.arattai\_scheduled\_messages;

CREATE POLICY "Arattai Scheduled: Manage" ON public.arattai\_scheduled\_messages FOR ALL TO authenticated USING (school\_id = get\_user\_school\_id() AND get\_user\_role() IN ('school\_admin', 'super\_admin')) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "Arattai Broadcast: Select" ON public.arattai\_broadcast\_campaigns;

CREATE POLICY "Arattai Broadcast: Select" ON public.arattai\_broadcast\_campaigns FOR SELECT TO authenticated

USING (school\_id = get\_user\_school\_id() AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "Arattai Broadcast: Manage" ON public.arattai\_broadcast\_campaigns;

CREATE POLICY "Arattai Broadcast: Manage" ON public.arattai\_broadcast\_campaigns FOR ALL TO authenticated USING (school\_id = get\_user\_school\_id() AND get\_user\_role() IN ('school\_admin', 'super\_admin')) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "Arattai Automation: Select" ON public.arattai\_automation\_rules;

CREATE POLICY "Arattai Automation: Select" ON public.arattai\_automation\_rules FOR SELECT TO authenticated

USING (school\_id = get\_user\_school\_id() AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher')));

DROP POLICY IF EXISTS "Arattai Automation: Manage" ON public.arattai\_automation\_rules;

CREATE POLICY "Arattai Automation: Manage" ON public.arattai\_automation\_rules FOR ALL TO authenticated USING (school\_id = get\_user\_school\_id() AND get\_user\_role() IN ('school\_admin', 'super\_admin')) WITH CHECK (school\_id = get\_user\_school\_id());

DROP POLICY IF EXISTS "Arattai Logs: Select" ON public.arattai\_message\_logs;

CREATE POLICY "Arattai Logs: Select" ON public.arattai\_message\_logs FOR SELECT TO authenticated

USING (school\_id = get\_user\_school\_id() AND (get\_user\_role() IN ('super\_admin', 'school\_admin', 'teacher') OR arattai\_contact\_id IN (SELECT id FROM arattai\_contacts WHERE user\_profile\_id = get\_user\_id())));

DROP POLICY IF EXISTS "Arattai Settings: Select/Manage" ON public.arattai\_settings;

CREATE POLICY "Arattai Settings: Select/Manage" ON public.arattai\_settings FOR ALL TO authenticated USING (school\_id = get\_user\_school\_id() AND get\_user\_role() IN ('school\_admin', 'super\_admin')) WITH CHECK (school\_id = get\_user\_school\_id());

3. Provision\_First\_Super\_Admin Auth

BEGIN;

-- Assume 'SmartGenEduX Global HQ' school exists

WITH global\_tenant AS (

  SELECT id FROM schools WHERE name = 'SmartGenEduX Global HQ'

),

new\_user\_auth AS (

  SELECT id, email FROM auth.users WHERE email = 'dev@smartgen.ai'

)

INSERT INTO user\_profiles (id, school\_id, role, first\_name, email, is\_active, created\_at, updated\_at)

SELECT

  new\_user\_auth.id,

  global\_tenant.id,

  'super\_admin'::user\_role, -- direct cast for enum

  'Vipul',

  new\_user\_auth.email,

  TRUE,

  NOW(),

  NOW()

FROM new\_user\_auth, global\_tenant

ON CONFLICT (id) DO UPDATE

SET role = EXCLUDED.role,

    school\_id = EXCLUDED.school\_id,

    is\_active = TRUE,

    updated\_at = NOW();

COMMIT;

4. Provision\_Super\_Admin

WITH UserAuth AS (

  SELECT id FROM auth.users WHERE email = 'dev@smartgen.ai'

),

Tenant AS (

  SELECT id AS school\_id FROM schools WHERE name = 'SmartGenEduX Global HQ'

)

INSERT INTO user\_profiles (id, school\_id, role, first\_name, email, is\_active)

SELECT

  ua.id,

  t.school\_id,

  'super\_admin'::user\_role,

  'Vipul',

  'dev@smartgen.ai',

  TRUE

FROM UserAuth ua, Tenant t

ON CONFLICT (id) DO UPDATE

SET

  role = EXCLUDED.role,

  school\_id = EXCLUDED.school\_id,

  is\_active = TRUE;

5. user\_profile\_Constrain

SELECT conname, contype, conkey

FROM pg\_constraint

WHERE conrelid = 'public.user\_profiles'::regclass;