

-- School Management System Schema Mapping

-- Users table - Base table for all user types
CREATE TABLE users (  id SERIAL PRIMARY KEY,  username VARCHAR(50) UNIQUE NOT NULL,  password VARCHAR(255) NOT NULL,  first_name VARCHAR(50) NOT NULL,  last_name VARCHAR(50) NOT NULL,  email VARCHAR(100) UNIQUE NOT NULL,  phone VARCHAR(20),  address TEXT,  last_login TIMESTAMP,  is_active BOOLEAN DEFAULT true,  fingerprint VARCHAR(255),  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Exams table
CREATE TABLE exams (  id SERIAL PRIMARY KEY,  name VARCHAR(100) NOT NULL,  subject VARCHAR(50) NOT NULL,  date DATE NOT NULL,  duration INTEGER NOT NULL, -- Duration in minutes  total_marks INTEGER NOT NULL,  created_by INTEGER REFERENCES teachers(teacher_id),  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Logs table
CREATE TABLE logs (  id SERIAL PRIMARY KEY,  user_id INTEGER REFERENCES users(id) ON DELETE CASCADE,  action VARCHAR(100) NOT NULL,  timestamp TIMESTAMP NOT NULL,  details TEXT );

-- Students table
CREATE TABLE students (  student_id SERIAL PRIMARY KEY,  user_id INTEGER REFERENCES users(id) ON DELETE CASCADE,  grade VARCHAR(10) NOT NULL,  section VARCHAR(5) NOT NULL,  enrollment_date DATE NOT NULL,  parent_name VARCHAR(100) NOT NULL,  parent_contact VARCHAR(20) NOT NULL,  gpa DECIMAL(3,2),  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Teachers table
CREATE TABLE teachers (  teacher_id SERIAL PRIMARY KEY,  user_id INTEGER REFERENCES users(id) ON DELETE CASCADE,  department VARCHAR(50) NOT NULL,  subjects TEXT[], -- Array of subjects  joining_date DATE NOT NULL,  qualification TEXT NOT NULL,  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Assessments table
CREATE TABLE assessments (  id SERIAL PRIMARY KEY,  title VARCHAR(100) NOT NULL,  description TEXT,  due_date TIMESTAMP NOT NULL,  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Principals table
CREATE TABLE principals (  school_id SERIAL PRIMARY KEY,  user_id INTEGER REFERENCES users(id) ON DELETE CASCADE,  appointment_date DATE NOT NULL,  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Admins table
CREATE TABLE admins (  admin_id SERIAL PRIMARY KEY,  user_id INTEGER REFERENCES users(id) ON DELETE CASCADE,  access_level VARCHAR(20) NOT NULL,  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Reports table
CREATE TABLE reports (  id SERIAL PRIMARY KEY,  type VARCHAR(50) NOT NULL,  date DATE NOT NULL,  generated_by INTEGER REFERENCES users(id),  content TEXT NOT NULL,  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

-- Timetable table
CREATE TABLE timetables (  id SERIAL PRIMARY KEY,  teacher_id INTEGER REFERENCES teachers(teacher_id) ON DELETE CASCADE,  subject VARCHAR(50) NOT NULL,  day VARCHAR(10) NOT NULL,  time_slot VARCHAR(10) NOT NULL,  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

<pre>CREATE TABLE logs (   id SERIAL PRIMARY KEY,    action VARCHAR(50) NOT NULL,    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,   user_id INTEGER REFERENCES users(id),    details JSONB NOT NULL );</pre>	<pre>due_date TIMESTAMP NOT NULL,  total_marks INTEGER NOT NULL,  subject VARCHAR(50) NOT NULL,  assigned_to INTEGER REFERENCES students(student_id), created_by INTEGER REFERENCES teachers(teacher_id), created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );</pre>	<pre>grade VARCHAR(10) NOT NULL,  section VARCHAR(5) NOT NULL,  schedule JSONB NOT NULL, -- Storing schedule as JSON for flexibility  valid_from DATE NOT NULL,  valid_to DATE NOT NULL,  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );</pre>
--	---	---

-- Student Records table
<pre>CREATE TABLE student_records (    student_id INTEGER PRIMARY KEY REFERENCES students(student_id),    current_grade VARCHAR(10) NOT NULL,    academic_year VARCHAR(9) NOT NULL,    subjects JSONB NOT NULL, -- Storing subjects and their details as JSON    attendance JSONB NOT NULL, -- Storing attendance records as JSON    achievements TEXT[],    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );</pre>

-- Sessions table
<pre>CREATE TABLE sessions (    id SERIAL PRIMARY KEY,    subject VARCHAR(50) NOT NULL,    teacher_id INTEGER REFERENCES teachers(teacher_id),    start_time TIME NOT NULL,    end_time TIME NOT NULL,    day VARCHAR(10) NOT NULL,    timetable_id INTEGER REFERENCES timetables(id) ON DELETE CASCADE,    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );</pre>

-- Exam Results table
<pre>CREATE TABLE exam_results (    id SERIAL PRIMARY KEY,    exam_id INTEGER REFERENCES exams(id) ON DELETE CASCADE,    student_id INTEGER REFERENCES students(student_id) ON DELETE CASCADE,    marks DECIMAL(5,2) NOT NULL,    grade VARCHAR(2),    remarks TEXT,    date DATE NOT NULL,    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );</pre>

-- Section Heads table
<pre>CREATE TABLE section_heads (    section_id SERIAL PRIMARY KEY,    teacher_id INTEGER REFERENCES teachers(teacher_id) ON DELETE CASCADE,    department VARCHAR(50) NOT NULL,    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );</pre>