-- Enable Row Level Security

ALTER TABLE auth.users ENABLE ROW LEVEL SECURITY;

-- Organizations

CREATE TABLE organizations (

id UUID DEFAULT uuid\_generate\_v4() PRIMARY KEY,

name TEXT NOT NULL,

subscription\_type TEXT NOT NULL,

settings JSONB DEFAULT '{}',

created\_at TIMESTAMPTZ DEFAULT NOW(),

updated\_at TIMESTAMPTZ DEFAULT NOW()

);

-- Enhanced Users Table (extends auth.users)

CREATE TABLE user\_profiles (

id UUID REFERENCES auth.users PRIMARY KEY,

organization\_id UUID REFERENCES organizations,

full\_name TEXT,

linkedin\_profile JSONB,

role TEXT,

settings JSONB DEFAULT '{}',

created\_at TIMESTAMPTZ DEFAULT NOW(),

updated\_at TIMESTAMPTZ DEFAULT NOW()

);

-- LinkedIn Posts

CREATE TABLE linkedin\_posts (

id UUID DEFAULT uuid\_generate\_v4() PRIMARY KEY,

user\_id UUID REFERENCES auth.users NOT NULL,

linkedin\_post\_id TEXT,

post\_url TEXT,

content TEXT NOT NULL,

media\_assets JSONB DEFAULT '[]',

post\_type TEXT NOT NULL,

audience\_type TEXT NOT NULL,

is\_scheduled BOOLEAN DEFAULT false,

scheduled\_for TIMESTAMPTZ,

posted\_at TIMESTAMPTZ,

created\_at TIMESTAMPTZ DEFAULT NOW(),

updated\_at TIMESTAMPTZ DEFAULT NOW(),

CONSTRAINT valid\_schedule CHECK (

(is\_scheduled AND scheduled\_for IS NOT NULL) OR

(NOT is\_scheduled)

)

);

-- Post Performance Tracking

CREATE TABLE post\_performance (

id UUID DEFAULT uuid\_generate\_v4() PRIMARY KEY,

post\_id UUID REFERENCES linkedin\_posts NOT NULL,

user\_id UUID REFERENCES auth.users NOT NULL,

impressions INTEGER DEFAULT 0,

reactions INTEGER DEFAULT 0,

comments INTEGER DEFAULT 0,

shares INTEGER DEFAULT 0,

clicks INTEGER DEFAULT 0,

additional\_metrics JSONB DEFAULT '{}',

measured\_at TIMESTAMPTZ NOT NULL,

created\_at TIMESTAMPTZ DEFAULT NOW(),

notes TEXT

);

-- Exercise System

CREATE TABLE exercise\_completions (

id UUID DEFAULT uuid\_generate\_v4() PRIMARY KEY,

user\_id UUID REFERENCES auth.users NOT NULL,

exercise\_type TEXT NOT NULL,

responses JSONB NOT NULL,

completion\_status TEXT NOT NULL,

score INTEGER,

completed\_at TIMESTAMPTZ DEFAULT NOW(),

updated\_at TIMESTAMPTZ DEFAULT NOW(),

CONSTRAINT valid\_status CHECK (

completion\_status IN ('complete', 'partial', 'in\_progress')

)

);

-- Notification System

CREATE TABLE notification\_preferences (

user\_id UUID REFERENCES auth.users PRIMARY KEY,

email\_enabled BOOLEAN DEFAULT true,

in\_app\_enabled BOOLEAN DEFAULT true,

reminder\_schedule TEXT DEFAULT 'default',

custom\_schedule INTEGER[],

excluded\_types TEXT[],

channel\_preferences JSONB DEFAULT '{}',

updated\_at TIMESTAMPTZ DEFAULT NOW()

);

CREATE TABLE notification\_queue (

id UUID DEFAULT uuid\_generate\_v4() PRIMARY KEY,

user\_id UUID REFERENCES auth.users NOT NULL,

type TEXT NOT NULL,

scheduled\_for TIMESTAMPTZ NOT NULL,

data JSONB,

status TEXT DEFAULT 'pending',

retry\_count INTEGER DEFAULT 0,

last\_attempted TIMESTAMPTZ,

created\_at TIMESTAMPTZ DEFAULT NOW(),

CONSTRAINT valid\_status CHECK (

status IN ('pending', 'processing', 'completed', 'failed')

)

);

-- Post Update History

CREATE TABLE post\_updates (

id UUID DEFAULT uuid\_generate\_v4() PRIMARY KEY,

post\_id UUID REFERENCES linkedin\_posts NOT NULL,

update\_type TEXT NOT NULL,

previous\_content JSONB,

new\_content JSONB,

updated\_at TIMESTAMPTZ DEFAULT NOW(),

updated\_by UUID REFERENCES auth.users NOT NULL

);

-- Brand Content

CREATE TABLE brand\_content (

id UUID DEFAULT uuid\_generate\_v4() PRIMARY KEY,

organization\_id UUID REFERENCES organizations NOT NULL,

content\_type TEXT NOT NULL,

title TEXT NOT NULL,

content JSONB NOT NULL,

is\_active BOOLEAN DEFAULT true,

created\_at TIMESTAMPTZ DEFAULT NOW(),

updated\_at TIMESTAMPTZ DEFAULT NOW(),

created\_by UUID REFERENCES auth.users NOT NULL

);

-- Row Level Security Policies

-- Users can only see their own data

CREATE POLICY "Users can view own data" ON user\_profiles

FOR SELECT USING (auth.uid() = id);

-- Users can see posts from their organization

CREATE POLICY "Organization posts access" ON linkedin\_posts

FOR SELECT USING (

auth.uid() IN (

SELECT id FROM user\_profiles

WHERE organization\_id = (

SELECT organization\_id FROM user\_profiles

WHERE id = linkedin\_posts.user\_id

)

)

);

-- Only post owners can update performance

CREATE POLICY "Update own post performance" ON post\_performance

FOR ALL USING (auth.uid() = user\_id);

-- Organization-based brand content access

CREATE POLICY "Brand content access" ON brand\_content

FOR SELECT USING (

EXISTS (

SELECT 1 FROM user\_profiles

WHERE id = auth.uid()

AND organization\_id = brand\_content.organization\_id

)

);

-- Indexes for Performance

CREATE INDEX idx\_posts\_user ON linkedin\_posts(user\_id);

CREATE INDEX idx\_performance\_post ON post\_performance(post\_id);

CREATE INDEX idx\_performance\_date ON post\_performance(measured\_at);

CREATE INDEX idx\_exercises\_user ON exercise\_completions(user\_id, exercise\_type);

CREATE INDEX idx\_notifications\_schedule ON notification\_queue(scheduled\_for, status);

NOTE: Key Features of the Schema:

1. **Comprehensive User Profiles**:
   * LinkedIn integration
   * Organization membership
   * Role-based access
2. **Robust Post Management**:
   * Full post history
   * Media asset tracking
   * Scheduling support
3. **Detailed Performance Tracking**:
   * Multiple metric types
   * Timestamped measurements
   * Additional metrics flexibility
4. **Exercise System Integration**:
   * Progress tracking
   * Response storage
   * Completion status
5. **Advanced Notification System**:
   * Multiple channel support
   * Custom schedules
   * Retry handling
6. **Organization & Brand Content**:
   * Multi-tenant support
   * Content management
   * Access control
7. **Security & Performance**:
   * Row Level Security
   * Optimized indexes
   * Data integrity constraints