-- =============================================================================

-- KURZORA TRADING PLATFORM - MASTER DATABASE SCHEMA

-- PostgreSQL with Supabase

-- Version: 1.0 (Final)

-- =============================================================================

-- Enable required extensions

CREATE EXTENSION IF NOT EXISTS "uuid-ossp";

CREATE EXTENSION IF NOT EXISTS "pg\_stat\_statements";

-- =============================================================================

-- 1. ENUMS & CUSTOM TYPES

-- =============================================================================

-- Signal types

CREATE TYPE signal\_type\_enum AS ENUM ('bullish', 'bearish', 'neutral');

-- Signal status

CREATE TYPE signal\_status\_enum AS ENUM ('active', 'triggered', 'expired', 'cancelled');

-- Signal actions for history tracking

CREATE TYPE signal\_action\_enum AS ENUM ('generated', 'triggered', 'stop\_loss\_hit', 'take\_profit\_hit', 'expired', 'manual\_close');

-- Alert types

CREATE TYPE alert\_type\_enum AS ENUM ('signal\_alert', 'trade\_update', 'portfolio\_summary', 'account\_notification');

-- Delivery channels

CREATE TYPE delivery\_channel\_enum AS ENUM ('email', 'telegram', 'push', 'sms');

-- Delivery status

CREATE TYPE delivery\_status\_enum AS ENUM ('pending', 'sent', 'delivered', 'failed', 'bounced');

-- Subscription tiers

CREATE TYPE subscription\_tier\_enum AS ENUM ('starter', 'professional', 'elite');

-- Trade types

CREATE TYPE trade\_type\_enum AS ENUM ('buy', 'sell');

-- =============================================================================

-- 2. USERS & AUTHENTICATION

-- =============================================================================

-- Users table (synced with Supabase Auth)

CREATE TABLE users (

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

-- Basic info

email VARCHAR(255) UNIQUE NOT NULL,

name VARCHAR(255) NOT NULL,

avatar\_url TEXT,

-- Subscription info

subscription\_tier subscription\_tier\_enum DEFAULT 'starter',

subscription\_status VARCHAR(50) DEFAULT 'trial',

subscription\_ends\_at TIMESTAMP WITH TIME ZONE,

stripe\_customer\_id VARCHAR(255),

-- Preferences

timezone VARCHAR(50) DEFAULT 'UTC',

language VARCHAR(10) DEFAULT 'en',

notification\_settings JSONB DEFAULT '{}',

-- Portfolio settings

starting\_balance DECIMAL(12,2) DEFAULT 10000.00,

current\_balance DECIMAL(12,2) DEFAULT 10000.00,

risk\_percentage DECIMAL(3,2) DEFAULT 2.00, -- 2% default risk per trade

-- Metadata

is\_active BOOLEAN DEFAULT true,

last\_login\_at TIMESTAMP WITH TIME ZONE,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- User sessions (for tracking active sessions)

CREATE TABLE user\_sessions (

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

user\_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

session\_token VARCHAR(255) UNIQUE NOT NULL,

device\_info JSONB,

ip\_address INET,

is\_active BOOLEAN DEFAULT true,

expires\_at TIMESTAMP WITH TIME ZONE NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- =============================================================================

-- 3. STOCK UNIVERSE & MARKET DATA

-- =============================================================================

-- Stock universe (all stocks we track)

CREATE TABLE stock\_universe (

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

ticker VARCHAR(10) UNIQUE NOT NULL,

company\_name VARCHAR(255) NOT NULL,

sector VARCHAR(100),

industry VARCHAR(100),

market\_cap BIGINT,

is\_active BOOLEAN DEFAULT true,

is\_islamic\_compliant BOOLEAN DEFAULT false,

last\_updated TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- Market data snapshots

CREATE TABLE market\_data\_snapshots (

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

ticker VARCHAR(10) NOT NULL,

price DECIMAL(10,4) NOT NULL,

volume BIGINT NOT NULL,

change\_percent DECIMAL(8,4),

high\_52w DECIMAL(10,4),

low\_52w DECIMAL(10,4),

market\_cap BIGINT,

snapshot\_time TIMESTAMP WITH TIME ZONE NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

-- Index for efficient queries

UNIQUE(ticker, snapshot\_time)

);

-- =============================================================================

-- 4. TRADING SIGNALS

-- =============================================================================

-- Main trading signals table

CREATE TABLE trading\_signals (

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

-- Signal identification

ticker VARCHAR(10) NOT NULL,

signal\_type signal\_type\_enum NOT NULL,

confidence\_score INTEGER NOT NULL CHECK (confidence\_score >= 0 AND confidence\_score <= 100),

-- Technical analysis

rsi\_value DECIMAL(5,2),

macd\_signal DECIMAL(8,4),

volume\_ratio DECIMAL(8,4),

support\_level DECIMAL(10,4),

resistance\_level DECIMAL(10,4),

timeframe VARCHAR(10) DEFAULT '15m',

-- Signal details

entry\_price DECIMAL(10,4),

stop\_loss DECIMAL(10,4),

take\_profit DECIMAL(10,4),

risk\_reward\_ratio DECIMAL(5,2),

-- AI explanation

explanation TEXT,

-- Status and timing

status signal\_status\_enum DEFAULT 'active',

expires\_at TIMESTAMP WITH TIME ZONE,

triggered\_at TIMESTAMP WITH TIME ZONE,

-- Metadata

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- Signal history for tracking performance

CREATE TABLE signal\_history (

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

signal\_id UUID NOT NULL REFERENCES trading\_signals(id) ON DELETE CASCADE,

action signal\_action\_enum NOT NULL,

price DECIMAL(10,4),

notes TEXT,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- =============================================================================

-- 5. PAPER TRADING

-- =============================================================================

-- Paper trades table

CREATE TABLE paper\_trades (

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

user\_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

signal\_id UUID REFERENCES trading\_signals(id),

-- Trade details

ticker VARCHAR(10) NOT NULL,

trade\_type trade\_type\_enum NOT NULL,

quantity INTEGER NOT NULL,

entry\_price DECIMAL(10,4) NOT NULL,

exit\_price DECIMAL(10,4),

-- Trade management

stop\_loss DECIMAL(10,4),

take\_profit DECIMAL(10,4),

-- Performance

profit\_loss DECIMAL(12,2),

profit\_loss\_percentage DECIMAL(8,4),

-- Status and timing

is\_open BOOLEAN DEFAULT true,

opened\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

closed\_at TIMESTAMP WITH TIME ZONE,

-- Metadata

notes TEXT,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- Portfolio snapshots for performance tracking

CREATE TABLE portfolio\_snapshots (

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

user\_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

-- Portfolio metrics

total\_value DECIMAL(12,2) NOT NULL,

cash\_balance DECIMAL(12,2) NOT NULL,

invested\_amount DECIMAL(12,2) NOT NULL,

unrealized\_pnl DECIMAL(12,2) DEFAULT 0,

realized\_pnl DECIMAL(12,2) DEFAULT 0,

-- Performance metrics

total\_return\_percentage DECIMAL(8,4),

daily\_change DECIMAL(12,2),

daily\_change\_percentage DECIMAL(8,4),

-- Metadata

snapshot\_date DATE NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

UNIQUE(user\_id, snapshot\_date)

);

-- =============================================================================

-- 6. ALERTS & NOTIFICATIONS

-- =============================================================================

-- User alert settings

CREATE TABLE user\_alert\_settings (

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

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

-- Alert preferences

min\_signal\_score INTEGER DEFAULT 80,

max\_alerts\_per\_day INTEGER DEFAULT 10,

trading\_hours\_only BOOLEAN DEFAULT true,

-- Delivery preferences

email\_enabled BOOLEAN DEFAULT true,

telegram\_enabled BOOLEAN DEFAULT false,

telegram\_chat\_id VARCHAR(255),

push\_enabled BOOLEAN DEFAULT true,

-- Timing preferences

quiet\_hours\_start TIME,

quiet\_hours\_end TIME,

timezone VARCHAR(50) DEFAULT 'UTC',

-- Metadata

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- Alert delivery log

CREATE TABLE alert\_delivery\_log (

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

user\_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

signal\_id UUID REFERENCES trading\_signals(id),

-- Alert details

alert\_type alert\_type\_enum NOT NULL,

delivery\_channel delivery\_channel\_enum NOT NULL,

delivery\_status delivery\_status\_enum DEFAULT 'pending',

-- Content

subject VARCHAR(255),

message TEXT,

-- Delivery tracking

sent\_at TIMESTAMP WITH TIME ZONE,

delivered\_at TIMESTAMP WITH TIME ZONE,

error\_message TEXT,

-- Metadata

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- =============================================================================

-- 7. DASHBOARD & USER INTERFACE

-- =============================================================================

-- Dashboard metrics (cached/aggregated data)

CREATE TABLE dashboard\_metrics (

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

user\_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

-- Signal metrics

todays\_signals INTEGER DEFAULT 0,

active\_signals INTEGER DEFAULT 0,

avg\_signal\_score DECIMAL(5,2) DEFAULT 0,

success\_rate DECIMAL(5,2) DEFAULT 0,

new\_signals\_last\_hour INTEGER DEFAULT 0,

-- Performance highlights

best\_performer\_symbol VARCHAR(10),

best\_performer\_profit DECIMAL(10,2),

best\_performer\_percentage DECIMAL(5,2),

-- Latest activity

latest\_signal\_symbol VARCHAR(10),

latest\_signal\_score INTEGER,

latest\_signal\_time TIMESTAMP WITH TIME ZONE,

alerts\_count INTEGER DEFAULT 0,

alerts\_description TEXT,

-- Cache timing

calculated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- User watchlists

CREATE TABLE watchlists (

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

user\_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

name VARCHAR(100) NOT NULL,

description TEXT,

is\_default BOOLEAN DEFAULT false,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

UNIQUE(user\_id, name)

);

-- Watchlist items

CREATE TABLE watchlist\_items (

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

watchlist\_id UUID NOT NULL REFERENCES watchlists(id) ON DELETE CASCADE,

ticker VARCHAR(10) NOT NULL,

notes TEXT,

added\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

UNIQUE(watchlist\_id, ticker)

);

-- =============================================================================

-- 8. SYSTEM ADMINISTRATION

-- =============================================================================

-- System health metrics

CREATE TABLE system\_health (

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

metric\_name VARCHAR(100) NOT NULL,

metric\_value DECIMAL(12,4) NOT NULL,

metric\_unit VARCHAR(20),

status VARCHAR(20) DEFAULT 'healthy',

recorded\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- API usage tracking

CREATE TABLE api\_usage\_log (

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

user\_id UUID REFERENCES users(id),

endpoint VARCHAR(255) NOT NULL,

method VARCHAR(10) NOT NULL,

status\_code INTEGER NOT NULL,

response\_time\_ms INTEGER,

ip\_address INET,

user\_agent TEXT,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- =============================================================================

-- 9. FUNCTIONS AND TRIGGERS

-- =============================================================================

-- Function to update updated\_at timestamp

CREATE OR REPLACE FUNCTION update\_updated\_at\_column()

RETURNS TRIGGER AS $$

BEGIN

NEW.updated\_at = NOW();

RETURN NEW;

END;

$$ language 'plpgsql';

-- Apply updated\_at trigger to relevant tables

CREATE TRIGGER update\_users\_updated\_at BEFORE UPDATE ON users

FOR EACH ROW EXECUTE FUNCTION update\_updated\_at\_column();

CREATE TRIGGER update\_trading\_signals\_updated\_at BEFORE UPDATE ON trading\_signals

FOR EACH ROW EXECUTE FUNCTION update\_updated\_at\_column();

CREATE TRIGGER update\_paper\_trades\_updated\_at BEFORE UPDATE ON paper\_trades

FOR EACH ROW EXECUTE FUNCTION update\_updated\_at\_column();

CREATE TRIGGER update\_stock\_universe\_updated\_at BEFORE UPDATE ON stock\_universe

FOR EACH ROW EXECUTE FUNCTION update\_updated\_at\_column();

CREATE TRIGGER update\_user\_alert\_settings\_updated\_at BEFORE UPDATE ON user\_alert\_settings

FOR EACH ROW EXECUTE FUNCTION update\_updated\_at\_column();

CREATE TRIGGER update\_watchlists\_updated\_at BEFORE UPDATE ON watchlists

FOR EACH ROW EXECUTE FUNCTION update\_updated\_at\_column();

-- =============================================================================

-- 10. ROW LEVEL SECURITY (RLS) POLICIES

-- =============================================================================

-- Enable RLS on user-specific tables

ALTER TABLE users ENABLE ROW LEVEL SECURITY;

ALTER TABLE paper\_trades ENABLE ROW LEVEL SECURITY;

ALTER TABLE portfolio\_snapshots ENABLE ROW LEVEL SECURITY;

ALTER TABLE user\_alert\_settings ENABLE ROW LEVEL SECURITY;

ALTER TABLE alert\_delivery\_log ENABLE ROW LEVEL SECURITY;

ALTER TABLE dashboard\_metrics ENABLE ROW LEVEL SECURITY;

ALTER TABLE watchlists ENABLE ROW LEVEL SECURITY;

ALTER TABLE watchlist\_items ENABLE ROW LEVEL SECURITY;

-- Users can only see their own data

CREATE POLICY users\_own\_data ON users

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

CREATE POLICY paper\_trades\_own\_data ON paper\_trades

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

CREATE POLICY portfolio\_snapshots\_own\_data ON portfolio\_snapshots

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

CREATE POLICY user\_alert\_settings\_own\_data ON user\_alert\_settings

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

CREATE POLICY alert\_delivery\_log\_own\_data ON alert\_delivery\_log

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

CREATE POLICY dashboard\_metrics\_own\_data ON dashboard\_metrics

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

CREATE POLICY watchlists\_own\_data ON watchlists

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

-- Watchlist items inherit from watchlist permissions

CREATE POLICY watchlist\_items\_own\_data ON watchlist\_items

FOR ALL USING (

EXISTS (

SELECT 1 FROM watchlists

WHERE watchlists.id = watchlist\_items.watchlist\_id

AND watchlists.user\_id = auth.uid()

)

);

-- Public read access for signals and stock data

CREATE POLICY trading\_signals\_public\_read ON trading\_signals

FOR SELECT USING (true);

CREATE POLICY stock\_universe\_public\_read ON stock\_universe

FOR SELECT USING (true);

-- =============================================================================

-- 11. INITIAL DATA SETUP

-- =============================================================================

-- Insert initial system health metrics

INSERT INTO system\_health (metric\_name, metric\_value, metric\_unit, status) VALUES

('signal\_processing\_speed', 0, 'ms', 'healthy'),

('database\_connections', 0, 'count', 'healthy'),

('api\_response\_time', 0, 'ms', 'healthy'),

('active\_users', 0, 'count', 'healthy');

-- =============================================================================

-- SCHEMA COMPLETE

-- =============================================================================

-- Schema version tracking

CREATE TABLE schema\_migrations (

version VARCHAR(255) PRIMARY KEY,

applied\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

INSERT INTO schema\_migrations (version) VALUES ('v1.0\_complete\_schema');

-- =============================================================================

-- 12. CREATE INDEXES FOR PERFORMANCE

-- =============================================================================

-- Indexes for trading\_signals table (for faster queries)

CREATE INDEX idx\_signals\_ticker ON trading\_signals (ticker);

CREATE INDEX idx\_signals\_score ON trading\_signals (confidence\_score);

CREATE INDEX idx\_signals\_status ON trading\_signals (status);

CREATE INDEX idx\_signals\_created ON trading\_signals (created\_at);

-- Indexes for market\_data\_snapshots

CREATE INDEX idx\_market\_data\_ticker\_time ON market\_data\_snapshots (ticker, snapshot\_time);

-- Indexes for paper\_trades

CREATE INDEX idx\_paper\_trades\_user ON paper\_trades (user\_id);

CREATE INDEX idx\_paper\_trades\_ticker ON paper\_trades (ticker);

CREATE INDEX idx\_paper\_trades\_open ON paper\_trades (is\_open);

-- Indexes for API usage tracking

CREATE INDEX idx\_api\_usage\_user ON api\_usage\_log (user\_id);

CREATE INDEX idx\_api\_usage\_endpoint ON api\_usage\_log (endpoint);

CREATE INDEX idx\_api\_usage\_created ON api\_usage\_log (created\_at);