RASAS

-- Users and Authentication

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

first\_name VARCHAR(100) NOT NULL,

last\_name VARCHAR(100) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

last\_login TIMESTAMP NULL,

is\_active BOOLEAN DEFAULT true

);

CREATE TABLE roles (

role\_id INT AUTO\_INCREMENT PRIMARY KEY,

role\_name VARCHAR(50) UNIQUE NOT NULL -- 'participant', 'administrator', 'reviewer', 'visitor'

);

CREATE TABLE user\_roles (

user\_id INT,

role\_id INT,

assigned\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (user\_id, role\_id),

FOREIGN KEY (user\_id) REFERENCES users(user\_id),

FOREIGN KEY (role\_id) REFERENCES roles(role\_id)

);

-- Abstract Submission and Review

CREATE TABLE abstracts (

abstract\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

title VARCHAR(255) NOT NULL,

content TEXT NOT NULL,

keywords VARCHAR(255),

submission\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status VARCHAR(50) NOT NULL, -- 'pending', 'under\_review', 'approved', 'rejected'

last\_updated TIMESTAMP,

track VARCHAR(100),

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

CREATE TABLE abstract\_reviews (

review\_id INT AUTO\_INCREMENT PRIMARY KEY,

abstract\_id INT,

reviewer\_id INT,

evaluation\_score INT,

feedback TEXT,

recommendation VARCHAR(50), -- 'approve', 'reject', 'revise'

review\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

last\_updated TIMESTAMP,

FOREIGN KEY (abstract\_id) REFERENCES abstracts(abstract\_id),

FOREIGN KEY (reviewer\_id) REFERENCES users(user\_id)

);

-- Event Registration

CREATE TABLE events (

event\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

description TEXT,

start\_date TIMESTAMP NOT NULL,

end\_date TIMESTAMP NOT NULL,

venue VARCHAR(255),

capacity INT,

registration\_deadline TIMESTAMP,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE registrations (

registration\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

event\_id INT,

registration\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status VARCHAR(50) NOT NULL, -- 'pending', 'confirmed', 'cancelled'

payment\_status VARCHAR(50),

amount\_paid DECIMAL(10,2),

confirmation\_code VARCHAR(100) UNIQUE,

FOREIGN KEY (user\_id) REFERENCES users(user\_id),

FOREIGN KEY (event\_id) REFERENCES events(event\_id)

);

-- Notifications

CREATE TABLE notifications (

notification\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

title VARCHAR(255) NOT NULL,

content TEXT NOT NULL,

type VARCHAR(50), -- 'system', 'review', 'event', 'deadline'

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

read\_at TIMESTAMP,

is\_read BOOLEAN DEFAULT false,

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

-- Event Schedules

CREATE TABLE schedules (

schedule\_id INT AUTO\_INCREMENT PRIMARY KEY,

event\_id INT,

title VARCHAR(255) NOT NULL,

description TEXT,

start\_time TIMESTAMP NOT NULL,

end\_time TIMESTAMP NOT NULL,

location VARCHAR(255),

speaker VARCHAR(255),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

last\_updated TIMESTAMP,

FOREIGN KEY (event\_id) REFERENCES events(event\_id)

);

-- Analytics

CREATE TABLE analytics\_logs (

log\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

action\_type VARCHAR(100) NOT NULL,

action\_details JSON,

ip\_address VARCHAR(45),

user\_agent TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

-- Announcements

CREATE TABLE announcements (

announcement\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

content TEXT NOT NULL,

published\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

expires\_at TIMESTAMP,

priority INT DEFAULT 0,

created\_by INT,

FOREIGN KEY (created\_by) REFERENCES users(user\_id)

);

-- Contact Information

CREATE TABLE contact\_info (

contact\_id INT AUTO\_INCREMENT PRIMARY KEY,

department VARCHAR(100),

email VARCHAR(255),

phone VARCHAR(50),

address TEXT,

contact\_person VARCHAR(100),

is\_primary BOOLEAN DEFAULT false

);

-- Create indexes for better query performance

CREATE INDEX idx\_users\_email ON users(email);

CREATE INDEX idx\_abstracts\_user\_id ON abstracts(user\_id);

CREATE INDEX idx\_abstracts\_status ON abstracts(status);

CREATE INDEX idx\_registrations\_user\_id ON registrations(user\_id);

CREATE INDEX idx\_registrations\_event\_id ON registrations(event\_id);

CREATE INDEX idx\_notifications\_user\_id ON notifications(user\_id);

CREATE INDEX idx\_schedules\_event\_id ON schedules(event\_id);