Name: Clinton Omari

RegNo: EB3/56416/21

CREATE DATABASE IF NOT EXISTS linkup;

USE linkup;

-- Users table (must be created before referencing it)

CREATE TABLE users (

id VARCHAR(255) PRIMARY KEY,

firstName VARCHAR(255) NOT NULL,

lastName VARCHAR(255) NOT NULL,

phoneNumber VARCHAR(20) NOT NULL,

email VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL,

profile VARCHAR(255),

coverImage VARCHAR(255),

isActive TINYINT(1) DEFAULT 0,

isUpdated TINYINT(1) DEFAULT 0,

isDelete TINYINT(1) DEFAULT 0,

createdAt DATETIME NOT NULL,

updatedAt DATETIME,

role VARCHAR(50) DEFAULT 'student'

);

-- Forums table (assumed for FK usage)

CREATE TABLE forums (

id VARCHAR(255) PRIMARY KEY,

name VARCHAR(255) NOT NULL,

description TEXT,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP

);

-- Attachments

CREATE TABLE attachments (

id VARCHAR(255) PRIMARY KEY,

title VARCHAR(255) NOT NULL,

description TEXT NOT NULL,

company\_name VARCHAR(255) NOT NULL,

application\_link VARCHAR(255) NOT NULL,

created\_by VARCHAR(255) NOT NULL,

created\_at DATETIME NOT NULL,

updated\_at DATETIME,

is\_deleted TINYINT(1) DEFAULT 0,

FOREIGN KEY (created\_by) REFERENCES users(id)

);

-- User Attachments

CREATE TABLE user\_attachments (

id VARCHAR(255) PRIMARY KEY,

user\_id VARCHAR(255) NOT NULL,

attachment\_id VARCHAR(255) NOT NULL,

applied\_at DATETIME NOT NULL,

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

FOREIGN KEY (attachment\_id) REFERENCES attachments(id)

);

-- Connections

CREATE TABLE connections (

id VARCHAR(255) PRIMARY KEY,

senderId VARCHAR(255) NOT NULL,

receiverId VARCHAR(255) NOT NULL,

status VARCHAR(50) DEFAULT 'pending',

createdAt DATETIME NOT NULL,

updatedAt DATETIME,

FOREIGN KEY (senderId) REFERENCES users(id),

FOREIGN KEY (receiverId) REFERENCES users(id)

);

-- Forum Members

CREATE TABLE forum\_members (

id VARCHAR(255) PRIMARY KEY,

forum\_id VARCHAR(255) NOT NULL,

user\_id VARCHAR(255) NOT NULL,

joined\_at DATETIME NOT NULL,

FOREIGN KEY (forum\_id) REFERENCES forums(id),

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

);

-- Forum Messages

CREATE TABLE forum\_messages (

id VARCHAR(255) PRIMARY KEY,

forum\_id VARCHAR(255) NOT NULL,

sender\_id VARCHAR(255) NOT NULL,

message TEXT,

file\_url VARCHAR(255),

file\_name VARCHAR(255),

sent\_at DATETIME NOT NULL,

FOREIGN KEY (forum\_id) REFERENCES forums(id),

FOREIGN KEY (sender\_id) REFERENCES users(id)

);

-- Direct Messages

CREATE TABLE messages (

id VARCHAR(255) PRIMARY KEY,

sender\_id VARCHAR(255) NOT NULL,

receiver\_id VARCHAR(255) NOT NULL,

message TEXT NOT NULL,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (sender\_id) REFERENCES users(id),

FOREIGN KEY (receiver\_id) REFERENCES users(id)

);

-- Password Reset Codes

CREATE TABLE PasswordResetCodes (

id VARCHAR(255) PRIMARY KEY,

email VARCHAR(255) NOT NULL,

resetCode VARCHAR(10) NOT NULL,

expirationTime DATETIME NOT NULL,

createdAt DATETIME DEFAULT UTC\_TIMESTAMP

);

CREATE INDEX idx\_password\_reset\_codes\_email ON PasswordResetCodes(email);

-- Posts

CREATE TABLE Posts (

id VARCHAR(255) PRIMARY KEY,

user\_id VARCHAR(255) NOT NULL,

content TEXT,

image\_url VARCHAR(255),

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

updated\_at DATETIME,

is\_deleted TINYINT(1) DEFAULT 0,

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

);

-- Post Likes

CREATE TABLE PostLikes (

id VARCHAR(255) PRIMARY KEY,

post\_id VARCHAR(255) NOT NULL,

user\_id VARCHAR(255) NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (post\_id) REFERENCES Posts(id),

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

CONSTRAINT UC\_PostLike UNIQUE (post\_id, user\_id)

);

-- Post Comments

CREATE TABLE PostComments (

id VARCHAR(255) PRIMARY KEY,

post\_id VARCHAR(255) NOT NULL,

user\_id VARCHAR(255) NOT NULL,

content TEXT NOT NULL,

created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

updated\_at DATETIME,

is\_deleted TINYINT(1) DEFAULT 0,

FOREIGN KEY (post\_id) REFERENCES Posts(id),

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

);

-- Resources

CREATE TABLE resources (

id VARCHAR(255) PRIMARY KEY,

title VARCHAR(255) NOT NULL,

description TEXT,

quantity INT,

isDeleted TINYINT(1) DEFAULT 0,

createdAt DATETIME NOT NULL,

updatedAt DATETIME

);

-- Borrowed Books

CREATE TABLE borrowed\_books (

id VARCHAR(255) PRIMARY KEY,

user\_id VARCHAR(255) NOT NULL,

resource\_id VARCHAR(255) NOT NULL,

borrowed\_date DATE NOT NULL,

return\_date DATE NOT NULL,

is\_returned TINYINT(1) DEFAULT 0,

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

FOREIGN KEY (resource\_id) REFERENCES resources(id)

);

