## **1. Viết câu lệnh SQL tạo các bảng**

-- Tạo bảng user

CREATE TABLE "user" (

id SERIAL PRIMARY KEY,

username VARCHAR NOT NULL,

password VARCHAR NOT NULL,

fullname VARCHAR NOT NULL,

avatar VARCHAR,

birthday TIMESTAMP,

created\_time TIMESTAMP NOT NULL DEFAULT NOW()

);

-- Tạo bảng friend

CREATE TABLE friend (

id SERIAL PRIMARY KEY,

sender\_id INT NOT NULL,

receiver\_id INT NOT NULL,

status VARCHAR CHECK (status IN ('pending', 'accepted', 'rejected')) NOT NULL,

created\_time TIMESTAMP NOT NULL DEFAULT NOW(),

FOREIGN KEY (sender\_id) REFERENCES "user"(id),

FOREIGN KEY (receiver\_id) REFERENCES "user"(id)

);

-- Tạo bảng message

CREATE TABLE message (

id SERIAL PRIMARY KEY,

sender\_id INT NOT NULL,

receiver\_id INT NOT NULL,

type VARCHAR NOT NULL CHECK (type IN ('text', 'image', 'video', 'file')),

content VARCHAR,

status VARCHAR CHECK (status IN ('sent', 'pending\_read', 'read')) NOT NULL,

created\_time TIMESTAMP NOT NULL DEFAULT NOW(),

FOREIGN KEY (sender\_id) REFERENCES "user"(id),

FOREIGN KEY (receiver\_id) REFERENCES "user"(id)

);

## **2. Viết câu lệnh SQL insert vào trong 3 table (mỗi table ít nhất 4 records)**

-- Thêm 4 user

INSERT INTO "user" (username, password, fullname, avatar, birthday, created\_time) VALUES

('alice', 'pass123', 'Alice Johnson', 'avatar1.jpg', '1990-05-01', NOW()),

('bob', 'pass456', 'Bob Smith', 'avatar2.jpg', '1989-07-12', NOW()),

('charlie', 'pass789', 'Charlie Brown', NULL, '1992-03-15', NOW()),

('dave', 'pass000', 'Dave Lee', 'avatar4.jpg', '1995-01-20', NOW());

-- Thêm 4 friend relationships

INSERT INTO friend (sender\_id, receiver\_id, status, created\_time) VALUES

(1, 2, 'accepted', NOW()),

(2, 3, 'accepted', NOW()),

(1, 3, 'pending', NOW()),

(2, 4, 'accepted', NOW());

-- Thêm 4 messages

INSERT INTO message (sender\_id, receiver\_id, type, content, status, created\_time) VALUES

(2, 3, 'text', 'Hello Charlie!', 'read', NOW()),

(3, 2, 'text', 'Hey Bob!', 'read', NOW()),

(2, 4, 'image', 'pic.jpg', 'pending\_read', NOW()),

(4, 2, 'video', 'video.mp4', 'sent', NOW());

## **3. Viết câu lệnh SQL query**

### **3.1. Lấy id, username, fullname, avatar của các user có id = 2, 3**

SELECT id, username, fullname, avatar

FROM "user"

WHERE id IN (2, 3);

### **3.2. Lấy các bạn bè của user có id = 2**

SELECT u.id, u.username, u.fullname, u.avatar

FROM friend f

JOIN "user" u ON

(f.sender\_id = u.id AND f.receiver\_id = 2) OR

(f.receiver\_id = u.id AND f.sender\_id = 2)

WHERE f.status = 'accepted' AND u.id != 2;

### **3.3. Lấy tin nhắn của user có id = 2 với bạn bè có id = 3**

SELECT id AS message\_id, sender\_id, receiver\_id, type, status, content, created\_time

FROM message

WHERE

(sender\_id = 2 AND receiver\_id = 3) OR

(sender\_id = 3 AND receiver\_id = 2)

ORDER BY created\_time;

### **3.4. Lấy tin nhắn cuối cùng với tất cả bạn bè của user có id = 2**

SELECT DISTINCT ON (friend\_id)

friend\_id,

id AS message\_id,

type,

status,

content,

sender\_id,

created\_time

FROM (

SELECT m.\*,

CASE

WHEN m.sender\_id = 2 THEN m.receiver\_id

ELSE m.sender\_id

END AS friend\_id

FROM message m

WHERE m.sender\_id = 2 OR m.receiver\_id = 2

) AS sub

ORDER BY friend\_id, created\_time DESC;

### **3.5. Lấy danh sách 10 user có số lượng bạn bè nhiều nhất trong hệ thống**

SELECT

u.id,

u.username,

u.avatar,

u.birthday,

COUNT(f.id) AS friend\_count

FROM "user" u

LEFT JOIN friend f ON (

(f.sender\_id = u.id OR f.receiver\_id = u.id)

AND f.status = 'accepted'

)

GROUP BY u.id, u.username, u.avatar, u.birthday

ORDER BY friend\_count DESC

LIMIT 10;