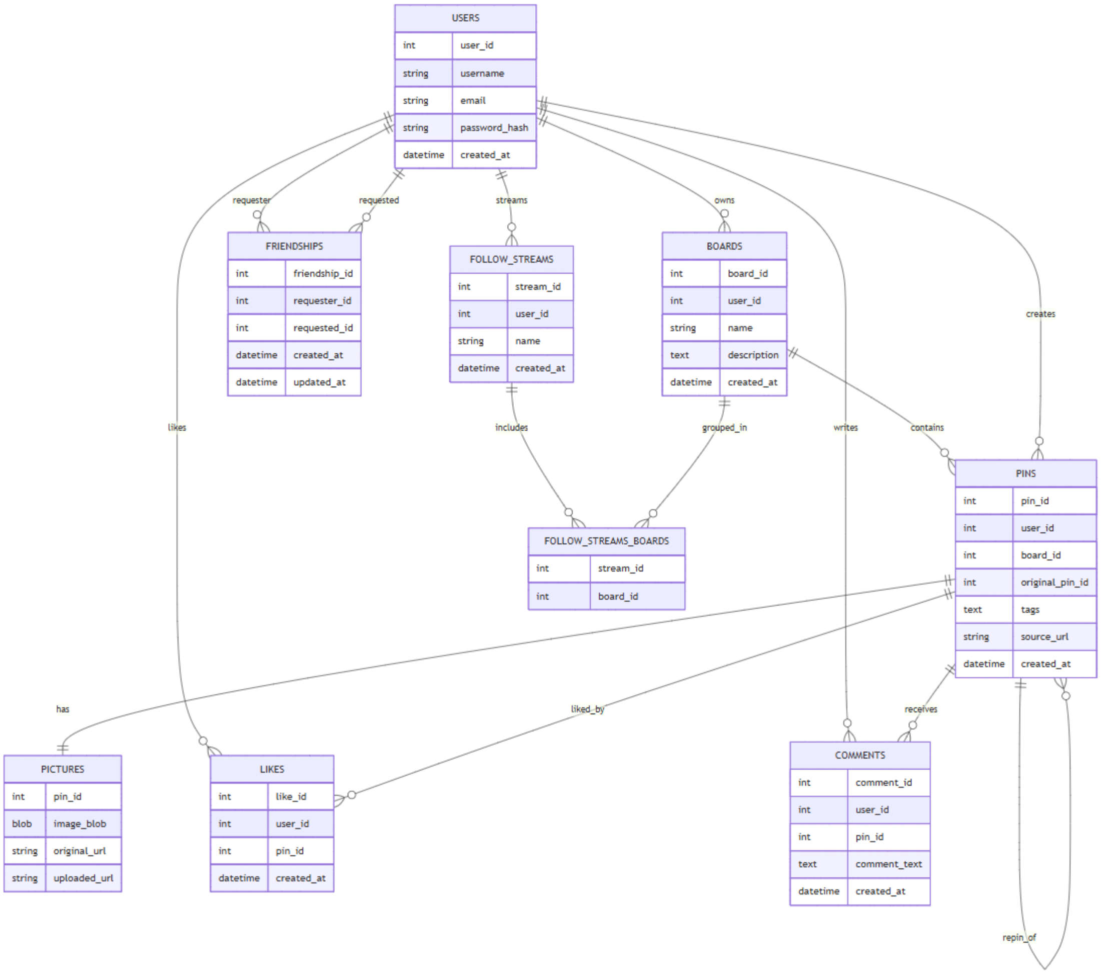
DB-Project-Part 1

ER diagram：(先用着，但建议换个新的)



Relational schema (keys & constraints) with database system：

Primary Key, *Foreign Key*

Users: user_id, username, email, password_hash, created_at;

Boards: board_id, *user_id(Users(user_id))*, name, description, created_at;

Pins: pin_id, *user_id(Users(user_id))*, *board_id(Boards(board_id))*, *original_pin_id(Pins(pin_id))*, tags, source_url, created_at;

Pictures: *pin_id(Pins(pin_id))*, image_blob, original_url, uploaded_url;

Friendships: friendship_id, *requester_id(Users(user_id))*, *requested_id(Users(user_id))*, created_at, updated_at;

FollowStreams: stream_id, *user_id(Users(user_id))*, name, created_at;

FollowStreamsBoards: *stream_id(FollowStreams(stream_id)), board_id(Boards(board_id))*;

Like: like_id, *user_id(Users(user_id)), pin_id(Pins(pin_id)),* created_at;

Comments: comment_id, *user_id(Users(user_id)), pin_id(Pins(pin_id))*, comment_text, created_at;

```
-- Users Table
CREATE TABLE Users (
    user_id SERIAL PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Boards Table
CREATE TABLE Boards (
    board_id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE
);

-- Pins Table
CREATE TABLE Pins (
    pin_id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    board_id INT NOT NULL,
    original_pin_id INT,
    tags TEXT,
    source_url VARCHAR(255),
```

```sql
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE,
    FOREIGN KEY (board_id) REFERENCES Boards(board_id) ON DELETE CASCADE,
    FOREIGN KEY (original_pin_id) REFERENCES Pins(pin_id) ON DELETE CASCADE
);

-- Pictures Table
CREATE TABLE Pictures (
    pin_id INT PRIMARY KEY,
    image_blob BYTEA NOT NULL,
    original_url VARCHAR(255),
    uploaded_url VARCHAR(255),
    FOREIGN KEY (pin_id) REFERENCES Pins(pin_id) ON DELETE CASCADE
);

-- Friendships Table
CREATE TABLE Friendships (
    friendship_id SERIAL PRIMARY KEY,
    requester_id INT NOT NULL,
    requested_id INT NOT NULL,
    status VARCHAR(10) NOT NULL CHECK (status IN ('pending', 'accepted', 'declined')),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (requester_id, requested_id),
    FOREIGN KEY (requester_id) REFERENCES Users(user_id) ON DELETE CASCADE,
    FOREIGN KEY (requested_id) REFERENCES Users(user_id) ON DELETE CASCADE
);

-- FollowStreams Table
CREATE TABLE FollowStreams (
    stream_id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    name VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE
);

-- FollowStreamBoards Table
CREATE TABLE FollowStreamBoards (
    stream_id INT NOT NULL,
    board_id INT NOT NULL,
    PRIMARY KEY (stream_id, board_id),
    FOREIGN KEY (stream_id) REFERENCES FollowStreams(stream_id) ON DELETE CASCADE,
    FOREIGN KEY (board_id) REFERENCES Boards(board_id) ON DELETE CASCADE
);

-- Likes Table
CREATE TABLE Likes (
    like_id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    pin_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (user_id, pin_id),
    FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE,
    FOREIGN KEY (pin_id) REFERENCES Pins(pin_id) ON DELETE CASCADE
);

-- Comments Table
CREATE TABLE Comments (
    comment_id SERIAL PRIMARY KEY,
    user_id INT NOT NULL,
    pin_id INT NOT NULL,
    comment_text TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE,
    FOREIGN KEY (pin_id) REFERENCES Pins(pin_id) ON DELETE CASCADE
);
```

Pins and Repins: Repins reference the original pin_id. Deleting the original will delete all repins followed by it. Pin’s tags stored as a comma-separated string in Pins for simplicity, enabling keyword search via LIKE.

Image Storage: Each image is stored once per original pin. Repins reference the same image blob. Follow streams are private, but boards/pins are public.

Examples for testing: （主要是丢给 gpt 生产的）

```
-- 1. Users (5 users)
INSERT INTO Users (user_id, username, email, password_hash) VALUES
  (1, 'erica',   'erica@example.com',   'hash_erica'),
  (2, 'timmy',   'timmy@example.com',   'hash_timmy'),
  (3, 'alice',   'alice@example.com',   'hash_alice'),
  (4, 'bob',     'bob@example.com',     'hash_bob'),
  (5, 'charlie', 'charlie@example.com', 'hash_charlie');

-- 2. Boards (7 boards)
INSERT INTO Boards (board_id, user_id, name, description) VALUES
  (1, 1, 'Furniture',          'Antique & modern pieces'),
  (2, 1, 'Dream Vacations',    'Beaches and hidden gems'),
  (3, 2, 'Super Dinosaurs',    'Everything T-Rex & friends'),
  (4, 2, 'Pirates',            'Arrr! Ships, maps, treasure'),
  (5, 3, 'Monsters',           'Creepy-cute creatures'),
  (6, 4, 'Tech',       'Latest tech innovations'),
  (7, 5, 'Nature Photography', 'Stunning landscapes and wildlife');

------------------------------
```

```sql
-- 3. Pins (12 pins: 10 original, 2 repins)
------------------------------
-- Original Pins
INSERT INTO Pins (pin_id, user_id, board_id, tags, source_url, created_at)
VALUES
  (1, 1, 1, 'couch,brown,modern', 'https://example.com/sofa.jpg', CURRENT_TIMESTAMP),
  (2, 1, 2, 'beach,sand,sea',   'https://example.com/beach.jpg',CURRENT_TIMESTAMP),
  (3, 2, 3, 'dinosaur,trex',    'https://example.com/trex.png',        CURRENT_TIMESTAMP),
  (4, 2, 4, 'pirate,ship',      'https://example.com/pirate.png',      CURRENT_TIMESTAMP),
  (7, 3, 5, 'monster,cute',     'https://example.com/cute_monster.jpg',CURRENT_TIMESTAMP),
  (8, 1, 2, 'mountain,alpine',  'https://example.com/alps.jpg',        CURRENT_TIMESTAMP),
  (9, 4, 6, 'phone,gadget',     'https://example.com/new_phone.png',   CURRENT_TIMESTAMP),
  (10,5, 7, 'forest,sunrise',   'https://example.com/forest_sunrise.jpg',CURRENT_TIMESTAMP);

-- Repins (引用原始 pin_id)
INSERT INTO Pins (pin_id, user_id, board_id, original_pin_id, created_at) VALUES
  (5, 2, 4, 2, CURRENT_TIMESTAMP),   -- Timmy repins Erica's beach to Pirates

  (6, 3, 5, 3, CURRENT_TIMESTAMP),   -- Alice repins Timmy's dinosaur to Monsters
  (11,5, 7, 2, CURRENT_TIMESTAMP),   -- Charlie repins Erica's beach to Nature Photography
  (12,4, 6, 3, CURRENT_TIMESTAMP);   -- Bob repins Timmy's dinosaur to Tech Gadgets

------------------------------
-- 4. Pictures (仅原始 Pins 存储图片)
------------------------------
INSERT INTO Pictures (pin_id, image_blob, original_url, uploaded_url) VALUES
  (1,  '\xDEADBEEF', 'https://example.com/sofa.jpg',         NULL),
  (2,  '\xDEADBEEF', 'https://example.com/beach.jpg',        NULL),
  (3,  '\xDEADBEEF', 'https://example.com/trex.png',         NULL),
  (4,  '\xDEADBEEF', 'https://example.com/pirate.png',       NULL),
  (7,  '\xDEADBEEF', 'https://example.com/cute_monster.jpg', NULL),
  (8,  '\xDEADBEEF', 'https://example.com/alps.jpg',         NULL),
  (9,  '\xDEADBEEF', 'https://example.com/new_phone.png',    NULL),
  (10, '\xDEADBEEF', 'https://example.com/forest_sunrise.jpg', NULL);

------------------------------
-- 5. Friendships (4 条关系)
------------------------------
```

```sql
INSERT INTO Friendships (friendship_id, requester_id, requested_id, status)
 VALUES
  (1, 1, 2, 'accepted'),     -- Erica ↔ Timmy
  (2, 2, 3, 'accepted'),     -- Timmy → Alice
  (3, 3, 4, 'accepted'),     -- Alice ↔ Bob
  (4, 5, 1, 'pending');      -- Charlie → Erica (等待接受)

------------------------------
-- 6. Follow Streams
------------------------------
-- Timmy 'Monsters and Dinosaurs'
INSERT INTO FollowStreams (stream_id, user_id, name) VALUES (1, 2, 'Monsters and Dinosaurs');
INSERT INTO FollowStreamBoards (stream_id, board_id) VALUES
  (1, 3), (1, 5), (1, 1), (1, 2);

-- Alice 'Design & Nature'
INSERT INTO FollowStreams (stream_id, user_id, name) VALUES (2, 3, 'Design & Nature');
INSERT INTO FollowStreamBoards (stream_id, board_id) VALUES
  (2, 1), (2, 7);

------------------------------
-- 7. Likes
------------------------------
INSERT INTO Likes (like_id, user_id, pin_id) VALUES
  (1, 2, 1),  -- Timmy like Erica sofa
  (2, 1, 3),  -- Erica like Timmy dio
  (3, 3, 2),  -- Alice like Erica bleach
  (4, 4, 7),  -- Bob like Alice monster
  (5, 5, 8),  -- Charlie like Erica 的雪山
  (6, 1, 10); -- Erica like Charlie 的森林日出

------------------------------
-- 8. Comments (5 条评论)
------------------------------
INSERT INTO Comments (comment_id, user_id, pin_id, comment_text) VALUES
  (1, 2, 1, 'Cute couch!'),
  (2, 1, 3, 'Rawr-some picture!'),
  (3, 5, 8, '这张雪山照片太震撼了！'),
  (4, 4, 3, '龙哥就是龙！'),
  (5, 3, 10, 'Man！');
```