DB-Project-Part 1

ER diagram：(先用着，但建议换个新的)

图示, 示意图

AI 生成的内容可能不正确。

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),

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');

*-------------------------------*

*-- 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)*

*-------------------------------*

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！');