**实体集**

用户表

用户ID（自增），昵称（唯一性约束），头像URL，账号（唯一性约束），密码（加密存储），邮箱（邮箱注册）（唯一性约束）、状态（是否封禁、注销等）

管理员表

管理员ID（自增），账号（唯一性约束），密码（加密存储），权限等级

种类表（存储帖子和感兴趣的种类）

种类ID（自增），种类名（唯一性约束）、种类简介、种类图片URL

帖子表

帖子ID，帖子标题，帖子种类（源自种类表），发布时间，帖子正文，帖子作者ID（源自用户表），点赞数，点踩数，状态

评论表

评论ID，评论正文，评论人ID（源自用户表），评论帖子ID（源自帖子表），发布时间

举报用户表

举报ID，举报原因，举报人ID（源自用户表），被举报人ID（源自用户表），举报时间，举报状态（是否审核完毕）

举报帖子表

举报ID，举报原因，举报人ID（源自用户表），被举报帖子ID（源自帖子表），举报时间，举报状态（是否审核完毕）

反馈表

反馈ID，反馈内容，反馈人ID（源自用户表），反馈时间，反馈状态（是否审核完毕）

**联系集**

用户关注表

用户ID（源自用户表），关注的人的ID（源自用户表）

用户收藏表

用户ID（源自用户表），收藏的帖子的ID（源自帖子表）

用户喜好表

用户ID（源自用户表），喜爱的种类ID（源自种类表）

**DDL**

-- 用户表

CREATE TABLE user (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

nickname VARCHAR(50) NOT NULL UNIQUE,

avatar\_url VARCHAR(255),

account VARCHAR(50) NOT NULL UNIQUE,

password\_hash VARCHAR(255) NOT NULL, -- 加密存储的密码

email VARCHAR(100) NOT NULL UNIQUE, -- 邮箱注册

status ENUM('active', 'banned', 'deleted') DEFAULT 'active'

);

-- 管理员表

CREATE TABLE admin (

admin\_id INT AUTO\_INCREMENT PRIMARY KEY,

account VARCHAR(50) NOT NULL UNIQUE,

password\_hash VARCHAR(255) NOT NULL, -- 加密存储的密码

permission\_level TINYINT(1) DEFAULT 0 -- 使用TINYINT而不是ENUM

);

-- 种类表（存储帖子和感兴趣的种类）

CREATE TABLE category (

category\_id INT AUTO\_INCREMENT PRIMARY KEY,

category\_name VARCHAR(100) NOT NULL UNIQUE,

category\_description TEXT,

category\_image\_url VARCHAR(255)

);

-- 帖子表

CREATE TABLE post (

post\_id INT AUTO\_INCREMENT PRIMARY KEY,

post\_title VARCHAR(255) NOT NULL,

category\_id INT NOT NULL,

FOREIGN KEY (category\_id) REFERENCES category(category\_id),

publish\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,

post\_content TEXT NOT NULL,

author\_id INT NOT NULL,

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

like\_count INT DEFAULT 0,

dislike\_count INT DEFAULT 0,

INDEX (publish\_time),

INDEX (author\_id)

);

-- 评论表

CREATE TABLE comment (

comment\_id INT AUTO\_INCREMENT PRIMARY KEY,

comment\_content TEXT NOT NULL,

commenter\_id INT NOT NULL,

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

post\_id INT NOT NULL,

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

publish\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,

like\_count INT DEFAULT 0,

dislike\_count INT DEFAULT 0,

INDEX (publish\_time),

INDEX (post\_id)

);

-- 举报用户表

CREATE TABLE user\_report (

report\_id INT AUTO\_INCREMENT PRIMARY KEY,

report\_reason TEXT NOT NULL,

reporter\_id INT NOT NULL,

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

reported\_user\_id INT NOT NULL,

FOREIGN KEY (reported\_user\_id) REFERENCES user(user\_id),

report\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,

status ENUM('pending', 'reviewed') DEFAULT 'pending'

);

-- 举报帖子表

CREATE TABLE post\_report (

report\_id INT AUTO\_INCREMENT PRIMARY KEY,

report\_reason TEXT NOT NULL,

reporter\_id INT NOT NULL,

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

reported\_post\_id INT NOT NULL,

FOREIGN KEY (reported\_post\_id) REFERENCES post(post\_id),

report\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,

status ENUM('pending', 'reviewed') DEFAULT 'pending'

);

-- 反馈表

CREATE TABLE feedback (

feedback\_id INT AUTO\_INCREMENT PRIMARY KEY,

feedback\_content TEXT NOT NULL,

feedback\_user\_id INT NOT NULL,

FOREIGN KEY (feedback\_user\_id) REFERENCES user(user\_id),

feedback\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,

status ENUM('pending', 'reviewed') DEFAULT 'pending'

);

-- 用户关注表

CREATE TABLE user\_follow (

user\_id INT NOT NULL,

follow\_user\_id INT NOT NULL,

PRIMARY KEY (user\_id, follow\_user\_id),

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

FOREIGN KEY (follow\_user\_id) REFERENCES user(user\_id)

);

-- 用户收藏表

CREATE TABLE user\_favorite (

user\_id INT NOT NULL,

favorite\_post\_id INT NOT NULL,

PRIMARY KEY (user\_id, favorite\_post\_id),

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

FOREIGN KEY (favorite\_post\_id) REFERENCES post(post\_id)

);

-- 用户喜好表

CREATE TABLE user\_preference (

user\_id INT NOT NULL,

preferred\_category\_id INT NOT NULL,

PRIMARY KEY (user\_id, preferred\_category\_id),

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

FOREIGN KEY (preferred\_category\_id) REFERENCES category(category\_id)

);