# 数据库

## 逻辑关系

| **模块** | **主表** | **相关表及逻辑关系** | **关系类型** |
| --- | --- | --- | --- |
| **用户管理** | Users | Privacy\_Settings (1:1), Follows (1), Search\_History (1) | 一对一、一对多 |
| **动态与内容** | Posts | Comments (1), Likes (1), Users (1) | 一对多、一对多 |
| **语音聊天与消息** | Messages, Voice\_Messages | Users (1), Users (1) | 一对多 |
| **通知模块** | Notifications | Users (1), Posts (1), Comments (1) | 一对多 |
| **推荐模块** | Recommendations | Users (1), Posts (1) | 一对多 |
| **短视频模块** | Short\_Videos | Comments (1), Users (1) | 一对多 |
| **直播模块** | Live\_Sessions | Live\_Comments (1), Users (1) | 一对多 |

### 逻辑关系说明：

* **用户管理模块**：Users 表是用户的主表，关联用户的隐私设置和关注关系。每个用户有一个隐私设置 (Privacy\_Settings) 表（1:1），以及多个关注者和粉丝（1通过 Follows 表）。
* **动态与内容模块**：Posts 表管理用户发布的动态，动态可以有多个评论 (Comments)，也可以被多个用户点赞 (Likes)。
* **语音聊天与消息模块**：Messages 和 Voice\_Messages 表管理用户间的文字消息和语音消息。每个用户都可以发送和接收多条消息。
* **通知模块**：Notifications 表存储用户收到的通知，如点赞、评论和关注通知。每个通知都与用户 (Users) 和相关的动态 (Posts) 或评论 (Comments) 关联。
* **推荐模块**：Recommendations 表记录基于算法为用户推荐的动态或短视频。每个推荐内容与用户和动态表相关联。
* **短视频模块**：Short\_Videos 表管理用户上传的短视频。每个视频可以有多个评论 (Comments)，与用户表 (Users) 关联。
* **直播模块**：Live\_Sessions 表存储用户发起的直播信息，每个直播会话可以有多个实时评论 (Live\_Comments)。

## 物理模型

基于MySQL数据库的物理模型，可以根据之前的表格总结，设计出各个表的结构。以下是每个模块的核心表结构示例。

## ### 1. 用户管理模块

### #### \*\*用户表 (Users)\*\*

CREATE TABLE Users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

email VARCHAR(100) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

phone VARCHAR(15),

avatar\_url VARCHAR(255),

bio TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

### #### \*\*关注和粉丝表 (Follows)\*\*

CREATE TABLE Follows (

follower\_id INT,

following\_id INT,

follow\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (follower\_id, following\_id),

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

FOREIGN KEY (following\_id) REFERENCES Users(user\_id)

);

### #### \*\*隐私设置表 (Privacy\_Settings)\*\*

CREATE TABLE Privacy\_Settings (

user\_id INT PRIMARY KEY,

profile\_visibility ENUM('public', 'friends', 'private') DEFAULT 'public',

message\_privacy ENUM('everyone', 'friends', 'none') DEFAULT 'everyone',

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

);

## ### 2. 动态与内容管理模块

### #### \*\*动态表 (Posts)\*\*

CREATE TABLE Posts (

post\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

content TEXT,

image\_url VARCHAR(255),

video\_url VARCHAR(255),

audio\_url VARCHAR(255),

like\_count INT DEFAULT 0,

comment\_count INT DEFAULT 0,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

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

);

### #### \*\*评论表 (Comments)\*\*

CREATE TABLE Comments (

comment\_id INT AUTO\_INCREMENT PRIMARY KEY,

post\_id INT,

user\_id INT,

comment TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

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

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

);

### #### \*\*点赞表 (Likes)\*\*

CREATE TABLE Likes (

post\_id INT,

user\_id INT,

like\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (post\_id, user\_id),

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

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

);

## ### 3. 语音聊天与消息模块

### #### \*\*语音消息表 (Voice\_Messages)\*\*

CREATE TABLE Voice\_Messages (

message\_id INT AUTO\_INCREMENT PRIMARY KEY,

sender\_id INT,

receiver\_id INT,

audio\_url VARCHAR(255),

duration INT,

sent\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

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

FOREIGN KEY (receiver\_id) REFERENCES Users(user\_id)

);

### #### \*\*实时语音聊天会话表 (Voice\_Chat\_Sessions)\*\*

CREATE TABLE Voice\_Chat\_Sessions (

session\_id INT AUTO\_INCREMENT PRIMARY KEY,

user1\_id INT,

user2\_id INT,

start\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

end\_time TIMESTAMP,

FOREIGN KEY (user1\_id) REFERENCES Users(user\_id),

FOREIGN KEY (user2\_id) REFERENCES Users(user\_id)

);

## ### 4. 消息与通知模块

### #### \*\*消息表 (Messages)\*\*

CREATE TABLE Messages (

message\_id INT AUTO\_INCREMENT PRIMARY KEY,

sender\_id INT,

receiver\_id INT,

content TEXT,

message\_type ENUM('text', 'image', 'video', 'audio') DEFAULT 'text',

sent\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

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

FOREIGN KEY (receiver\_id) REFERENCES Users(user\_id)

);

### #### \*\*通知表 (Notifications)\*\*

CREATE TABLE Notifications (

notification\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

notification\_type ENUM('like', 'comment', 'follow', 'message'),

triggered\_by INT,

post\_id INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

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

FOREIGN KEY (triggered\_by) REFERENCES Users(user\_id),

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

);

## ### 5. 短视频模块

### #### \*\*短视频表 (Short\_Videos)\*\*

CREATE TABLE Short\_Videos (

video\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

video\_url VARCHAR(255),

thumbnail\_url VARCHAR(255),

duration INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

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

);

## ### 6. 搜索与推荐模块

### #### \*\*搜索历史表 (Search\_History)\*\*

CREATE TABLE Search\_History (

search\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

search\_query VARCHAR(255),

search\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

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

);

### #### \*\*推荐内容表 (Recommendations)\*\*

CREATE TABLE Recommendations (

recommendation\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

post\_id INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

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

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

);

## ### 7. 直播与互动模块

### #### \*\*直播表 (Live\_Sessions)\*\*

CREATE TABLE Live\_Sessions (

session\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

session\_title VARCHAR(255),

stream\_url VARCHAR(255),

start\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

end\_time TIMESTAMP,

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

);

### #### \*\*直播评论表 (Live\_Comments)\*\*

CREATE TABLE Live\_Comments (

comment\_id INT AUTO\_INCREMENT PRIMARY KEY,

session\_id INT,

user\_id INT,

comment TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (session\_id) REFERENCES Live\_Sessions(session\_id),

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

);

这些表结构确保了该社交App的所有主要功能都能通过MySQL进行高效的存储和管理。你可以根据具体需求进一步优化表的索引、约束和外键关系。