

数据库结构：

1. 用户表 (users)

存储论坛的用户信息。

```
CREATE TABLE users (  
  user_id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL UNIQUE,  
  email VARCHAR(100) NOT NULL UNIQUE,  
  password_hash VARCHAR(255) NOT NULL,  
  bio TEXT,  
  avatar_url VARCHAR(255),  
  role ENUM('admin', 'moderator', 'user') DEFAULT 'user',  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  last_login TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
);
```

2. 帖子表 (posts)

存储用户发布的帖子信息。

```
CREATE TABLE posts (  
  post_id INT AUTO_INCREMENT PRIMARY KEY,  
  user_id INT NOT NULL,  
  category_id INT,  
  title VARCHAR(255) NOT NULL,  
  content TEXT NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  views INT DEFAULT 0,  
  FOREIGN KEY (user_id) REFERENCES users(user_id),  
  FOREIGN KEY (category_id) REFERENCES categories(category_id)  
);
```

3. 评论表 (comments)

存储用户在帖子中的评论信息。

```
CREATE TABLE comments (  
  comment_id INT AUTO_INCREMENT PRIMARY KEY,  
  post_id INT NOT NULL,  
  user_id INT NOT NULL,  
  content TEXT NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (post_id) REFERENCES posts(post_id),  
  FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

4. 分类表 (categories)

存储论坛的帖子分类信息。

```
CREATE TABLE categories (  
  category_id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100) NOT NULL UNIQUE,  
  description TEXT,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

5. 帖子标签表 (tags)

存储帖子标签信息，用于给帖子添加标签。

```
CREATE TABLE tags (  
  tag_id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(50) NOT NULL UNIQUE  
);
```

6. 帖子标签关联表 (post_tags)

用于关联帖子和标签的多对多关系。

```
CREATE TABLE post_tags (  
  post_id INT NOT NULL,  
  tag_id INT NOT NULL,  
  PRIMARY KEY (post_id, tag_id),  
  FOREIGN KEY (post_id) REFERENCES posts(post_id),  
  FOREIGN KEY (tag_id) REFERENCES tags(tag_id)  
);
```

7. 私信表 (messages)

用于用户之间的私信交流。

```
CREATE TABLE messages (  
  message_id INT AUTO_INCREMENT PRIMARY KEY,  
  sender_id INT NOT NULL,  
  receiver_id INT NOT NULL,  
  content TEXT NOT NULL,  
  sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (sender_id) REFERENCES users(user_id),  
  FOREIGN KEY (receiver_id) REFERENCES users(user_id)  
);
```

8. 点赞表 (likes)

用于存储帖子和评论的点赞信息。

```
CREATE TABLE likes (  
  like_id INT AUTO_INCREMENT PRIMARY KEY,  
  user_id INT NOT NULL,  
  post_id INT DEFAULT NULL,  
  comment_id INT DEFAULT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (user_id) REFERENCES users(user_id),  
  FOREIGN KEY (post_id) REFERENCES posts(post_id),  
  FOREIGN KEY (comment_id) REFERENCES comments(comment_id)  
);
```

9. 用户权限表 (permissions)

用于存储用户角色和权限的关联信息（扩展功能）。

```
CREATE TABLE permissions (  
  permission_id INT AUTO_INCREMENT PRIMARY KEY,  
  role ENUM('admin', 'moderator', 'user') NOT NULL,  
  permission_name VARCHAR(50) NOT NULL,  
  description TEXT  
);
```

10. 收藏帖子表 (Favorites)

功能概述：

用户可以收藏自己感兴趣的帖子，收藏的帖子可以在用户的个人中心中查看和管理。

表结构设计：

```
CREATE TABLE favorites (  
  favorite_id INT AUTO_INCREMENT PRIMARY KEY, -- 收藏记录的唯一ID  
  user_id INT NOT NULL, -- 收藏的用户ID，关联users表  
  post_id INT NOT NULL, -- 收藏的帖子ID，关联posts表  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- 收藏时间  
  FOREIGN KEY (user_id) REFERENCES users(user_id), -- 外键，关联users表  
  FOREIGN KEY (post_id) REFERENCES posts(post_id) -- 外键，关联posts表  
);
```

表字段解释：

- `favorite_id`: 每条收藏记录的唯一标识。
- `user_id`: 该条收藏记录所属的用户，外键关联 `users` 表的 `user_id`。
- `post_id`: 被收藏的帖子，外键关联 `posts` 表的 `post_id`。
- `created_at`: 用户收藏帖子时的时间戳。

11. 账户积分表 (Points)

功能概述：

该表记录用户在论坛中的积分变动情况。用户通过发帖、评论、下载资源等行为获得积分，并通过该表记录具体的积分来源。

表结构设计：

```
CREATE TABLE points (  
    point_id INT AUTO_INCREMENT PRIMARY KEY, -- 积分记录的唯一ID  
    user_id INT NOT NULL, -- 用户ID，关联users表  
    points INT NOT NULL, -- 积分数值，正数为增加积分，负数为扣除积分  
    action_type VARCHAR(100) NOT NULL, -- 积分变动的原因或行为（例如：发帖、下载资源）  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- 积分变动的时间  
    FOREIGN KEY (user_id) REFERENCES users(user_id) -- 外键，关联users表  
);
```

表字段解释：

- point_id: 每条积分变动的唯一标识。
- user_id: 该条积分记录所属的用户，外键关联 users 表的 user_id。
- points: 积分的数值，正数表示增加积分，负数表示扣除积分。
- action_type: 描述该积分记录的来源，例如“发帖”、“评论”、“资源下载”等。
- created_at: 积分变动的时间戳。

数据库设计说明：

- users 表存储用户的基本信息，包括用户名、邮箱、密码和头像等。role 字段用于区分用户、版主和管理员。
- posts 表存储用户发布的帖子，支持分类功能 (category_id)，可以按分类查看帖子。
- comments 表存储对帖子的评论，每条评论与一个用户和一个帖子相关联。
- categories 表提供帖子分类功能，管理员可以管理不同的帖子分类。
- tags 和 post_tags 表用于给帖子添加多个标签，增强帖子管理和搜索功能。
- messages 表允许用户之间发送私信，支持基本的用户交互。
- likes 表支持对帖子和评论的点赞功能。
- permissions 表可以扩展角色权限管理，管理员可以管理不同角色的权限。

如果有其他功能或更复杂的业务逻辑需求，可以在此基础上进行扩展。

以下是数据库中各个键值的中文解释：

1. 用户表 (users)

- **user_id**: 用户ID, 唯一标识每个用户的编号。
- **username**: 用户名, 用户的登录名或显示名, 必须唯一。
- **email**: 电子邮箱, 用户的电子邮箱地址, 必须唯一。
- **password_hash**: 密码哈希, 用户的密码经过加密后的值。
- **bio**: 个人简介, 用户的个人描述或自我介绍。
- **avatar_url**: 头像URL, 用户头像的图片链接。
- **role**: 角色, 用户在论坛中的权限类型 (如管理员、版主、普通用户)。
- **created_at**: 创建时间, 用户注册的时间。
- **last_login**: 最后登录时间, 用户上次登录的时间。

2. 帖子表 (posts)

- **post_id**: 帖子ID, 唯一标识每个帖子的编号。
- **user_id**: 用户ID, 发帖人的唯一编号, 关联到用户表的 **user_id**。
- **category_id**: 分类ID, 帖子所属分类的编号, 关联到分类表的 **category_id**。
- **title**: 标题, 帖子的标题或主题。
- **content**: 内容, 帖子的正文内容。
- **created_at**: 创建时间, 帖子发布的时间。
- **updated_at**: 更新时间, 帖子的最后一次修改时间。
- **views**: 浏览量, 帖子被浏览的次数。

3. 评论表 (comments)

- **comment_id**: 评论ID, 唯一标识每条评论的编号。
- **post_id**: 帖子ID, 评论所对应的帖子编号, 关联到帖子表的 **post_id**。
- **user_id**: 用户ID, 评论发布者的编号, 关联到用户表的 **user_id**。
- **content**: 内容, 评论的具体内容。
- **created_at**: 创建时间, 评论发布的时间。

4. 分类表 (categories)

- **category_id**: 分类ID, 唯一标识每个分类的编号。
- **name**: 分类名称, 分类的名称。
- **description**: 分类描述, 对分类的简要说明。
- **created_at**: 创建时间, 分类创建的时间。

5. 标签表 (tags)

- **tag_id**: 标签ID, 唯一标识每个标签的编号。
- **name**: 标签名称, 标签的名称 (如“技术”、“新闻”等)。

6. 帖子标签关联表 (post_tags)

- **post_id**: 帖子ID, 关联的帖子编号, 关联到帖子表的 **post_id**。
- **tag_id**: 标签ID, 关联的标签编号, 关联到标签表的 **tag_id**。

7. 私信表 (messages)

- **message_id**: 私信ID, 唯一标识每条私信的编号。
- **sender_id**: 发送者ID, 发送私信的用户编号, 关联到用户表的 **user_id**。
- **receiver_id**: 接收者ID, 接收私信的用户编号, 关联到用户表的 **user_id**。
- **content**: 内容, 私信的具体内容。
- **sent_at**: 发送时间, 私信发送的时间。

8. 点赞表 (likes)

- **like_id**: 点赞ID, 唯一标识每条点赞记录的编号。
- **user_id**: 用户ID, 点赞的用户编号, 关联到用户表的 **user_id**。
- **post_id**: 帖子ID, 点赞的帖子编号, 关联到帖子表的 **post_id**, 可以为空。
- **comment_id**: 评论ID, 点赞的评论编号, 关联到评论表的 **comment_id**, 可以为空。
- **created_at**: 创建时间, 点赞发生的时间。

9. 权限表 (permissions)

- **permission_id**: 权限ID, 唯一标识每个权限的编号。
- **role**: 角色, 定义用户的权限类别 (如管理员、版主、普通用户)。
- **permission_name**: 权限名称, 具体权限的名称。
- **description**: 权限描述, 对该权限的简要说明。表字段解释:

10. 收藏帖子表 (Favorites)

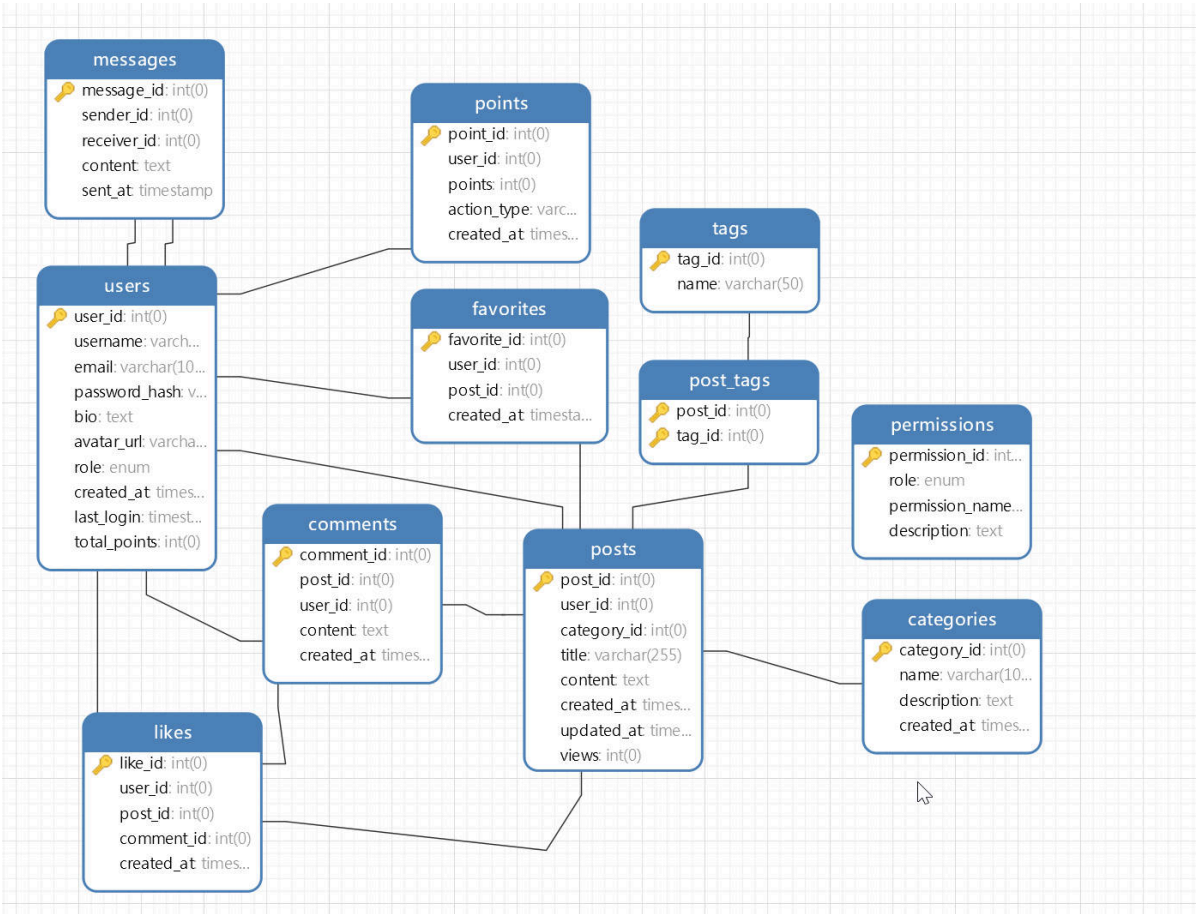
- **favorite_id**: 每条收藏记录的唯一标识。
- **user_id**: 该条收藏记录所属的用户, 外键关联 **users** 表的 **user_id**。
- **post_id**: 被收藏的帖子, 外键关联 **posts** 表的 **post_id**。
- **created_at**: 用户收藏帖子时的时间戳。

11. 账户积分表 (Points)

- **point_id**: 每条积分变动的唯一标识。
- **user_id**: 该条积分记录所属的用户, 外键关联 **users** 表的 **user_id**。
- **points**: 积分的数值, 正数表示增加积分, 负数表示扣除积分。
- **action_type**: 描述该积分记录的来源, 例如“发帖”、“评论”、“资源下载”等。
- **created_at**: 积分变动的时间戳。

键值中文总结

- **主键 (Primary Key, PK)** : 唯一标识每条记录的字段, 如 `user_id` 是用户表的主键, 每个用户都有唯一的ID。
- **外键 (Foreign Key, FK)** : 引用其他表中的主键以建立关系, 如 `user_id` 是帖子表中的外键, 指向用户表中的主键 `user_id`, 表示哪个用户发了该帖子。
- **时间戳 (Timestamp)** : `created_at` 和 `updated_at` 用于记录每条记录的创建时间和更新时间。



数据库关系表 (ER 模型)

1. 用户表 (users)

- **主键:** `user_id`
- **关联关系:**
 - `posts.user_id` 外键指向 `users.user_id`
 - `comments.user_id` 外键指向 `users.user_id`
 - `messages.sender_id` 外键指向 `users.user_id`
 - `messages.receiver_id` 外键指向 `users.user_id`
 - `likes.user_id` 外键指向 `users.user_id`
 - `favorites.user_id` 外键指向 `users.user_id`
 - `points.user_id` 外键指向 `users.user_id`
- **新增字段:** `total_points` 记录用户的总积分

2. 帖子表 (posts)

- 主键: `post_id`
- 外键:
 - `user_id` 外键指向 `users.user_id`
 - `category_id` 外键指向 `categories.category_id`
- 关联关系:
 - `comments.post_id` 外键指向 `posts.post_id`
 - `post_tags.post_id` 外键指向 `posts.post_id`
 - `likes.post_id` 外键指向 `posts.post_id`
 - `favorites.post_id` 外键指向 `posts.post_id`

3. 评论表 (`comments`)

- 主键: `comment_id`
- 外键:
 - `post_id` 外键指向 `posts.post_id`
 - `user_id` 外键指向 `users.user_id`
- 关联关系:
 - `likes.comment_id` 外键指向 `comments.comment_id`

4. 分类表 (`categories`)

- 主键: `category_id`
- 关联关系:
 - `posts.category_id` 外键指向 `categories.category_id`

5. 标签表 (`tags`)

- 主键: `tag_id`
- 关联关系:
 - `post_tags.tag_id` 外键指向 `tags.tag_id`

6. 帖子标签关联表 (`post_tags`)

- 联合主键: `post_id`, `tag_id`
- 外键:
 - `post_id` 外键指向 `posts.post_id`
 - `tag_id` 外键指向 `tags.tag_id`

7. 私信表 (`messages`)

- 主键: `message_id`
- 外键:
 - `sender_id` 外键指向 `users.user_id`
 - `receiver_id` 外键指向 `users.user_id`

8. 点赞表 (`likes`)

- 主键: `like_id`
- 外键:

- `user_id` 外键指向 `users.user_id`
- `post_id` 外键指向 `posts.post_id` (可选)
- `comment_id` 外键指向 `comments.comment_id` (可选)

新增表：

9. 收藏帖子表 (favorites)

- 主键: `favorite_id`
- 外键:
 - `user_id` 外键指向 `users.user_id`
 - `post_id` 外键指向 `posts.post_id`
- 关联关系:
 - 用户和帖子之间的多对多关系（用户可以收藏多个帖子，帖子可以被多个用户收藏）

10. 积分表 (points)

- 主键: `point_id`
- 外键:
 - `user_id` 外键指向 `users.user_id`
- 字段:
 - `points` 记录积分数值
 - `action_type` 表示积分变动的类型（如发帖、评论、下载资源等）
- 关联关系:
 - 用户和积分记录之间的一对多关系（一个用户可以有多个积分变动记录）

ER 模型关系说明：

- 用户表 (`users`) 是核心表，其他表如 帖子表 (`posts`)、评论表 (`comments`)、私信表 (`messages`)、收藏表 (`favorites`) 和 积分表 (`points`) 通过外键与 `users` 表建立了关联。
 - 帖子表 (`posts`) 和 评论表 (`comments`) 通过 `user_id` 与 用户表 (`users`) 关联，同时帖子还与 分类表 (`categories`) 和 标签表 (`tags`) 相关联。
 - 标签表 (`tags`) 和 帖子表 (`posts`) 是多对多关系，通过 帖子标签关联表 (`post_tags`) 来实现。
 - 私信表 (`messages`) 允许用户之间发送私信，具有 `sender_id` 和 `receiver_id` 外键，指向 用户表 (`users`)。
 - 点赞表 (`likes`) 用于存储用户对帖子或评论的点赞，点赞可以与帖子或评论关联。
 - 收藏表 (`favorites`) 用于记录用户对帖子的收藏，实现用户和帖子之间的多对多关系。
 - 积分表 (`points`) 用于记录用户的积分变动，记录用户在论坛中的各种操作（如发帖、下载资源）带来的积分变化。
-

ER 模型示例图

- `users` 表与 `posts`、`comments`、`favorites`、`points` 表之间的关系是**一对多**。
- `posts` 表与 `categories` 表是**多对一**关系。
- `posts` 表与 `tags` 表是**多对多**关系，通过 `post_tags` 表实现。
- `favorites` 表通过 `user_id` 和 `post_id` 建立**多对多**关系，记录用户收藏的帖子。
- `points` 表与 `users` 表建立**一对多**关系，记录每个用户的积分变动。

通过这些关系表设计，数据库结构已经完善，支持用户发帖、评论、收藏帖子、积分管理、标签管理等多项功能，同时保证了系统的可扩展性和高效性。