**实验六 数据库系统的设计**

XX班 100002000001 许愿

**【实验目的】**

要求学生在掌握数据库原理以及熟悉SQL Sever软件的基础上，设计一个SQL Sever的大学生大创组队信息平台系统的数据库, 为今后进一步开发数据库应用系统奠定基础。

**【实验环境】**

Windows 11, SQL Server 2022 + Mermaid（生成E-R图）

**【实验内容】**

根据大学生在日常中所遇到的大创等比赛组队的需求，设计出一套组队信息平台所使用的数据库，完成需求分析、概念结构设计和逻辑结构设计。

**【实验步骤】**

1. **需求分析阶段**

大学生大创组队信息平台涉及到以下方面的**业务过程**：

## **用户登录/注册/注销/填写个人信息**

用户首次使用本平台时，可选择注册账号。注册时需填写昵称和密码，并绑定联系方式。注册成功后，用户可以在个人信息界面补充或修改个人资料。后续使用中，用户可通过账号密码登录。若不再使用，用户可选择注销账号。

**所需数据**：用户ID，用户昵称，用户密码，用户联系方式，用户其他信息

## **项目信息展示**

为方便申请人深入了解项目，项目发起后，平台会展示发起人填写的信息。游客登录后即可查看各项目的详细内容。

**所需数据：**项目ID、项目名称、项目背景、项目简介、项目内容、项目状态、立项时间、预计结项时间、当前项目进度

## **项目信息分类检索**

系统设置了项目信息分类检索功能，便于项目参与者寻找适合自己的项目。

**所需数据：**用户输入的检索数据、项目ID、项目名称、项目背景、项目内容

## **用户个人信息查看**

用户在登录账号后，可以查看自己的个人信息。

**所需数据：**用户ID、账号创建时间、昵称、性别、联系方式、就读院校、专业、年级、个人简介、个人积分

## **用户对项目评论**

在每一个项目完成后，此项目的参与者可以对其进行评论，以便于后续的用户查看此项目时了解。

**所需数据：**评价内容、评价时间、评价人、评价类型（好/坏）、评价的项目ID、项目的具体信息

## **用户之间私聊沟通**

用户之间可以主动发起私聊，方便客户与用户进行精准交流。

**所需数据：**用户**ID**

## **项目发起人操作项目**

项目发起人可以上传一个指定项目，该项目创建后其他人可选择加入，项目发起人有权限对其项目进行删除或编辑操作，删除后项目为不可加入状态，编辑后更新显示最新内容，可对项目具体名称、背景、简介等进行修改。

**所需数据：**用户ID、项目ID、项目名称、项目背景、项目简介、项目内容、项目状态、立项时间、预计结项时间、当前项目进度

## **项目发起人发布/撤回组队需求**

项目发起人在登录自己的账号后，可以在项目界面发布组队需求，修改组队需求信息以及撤回项目组队的需求。

**所需数据**：项目发起人ID、需求ID、需求发起时间、组队目的、组队类型、人数要求、需求状态、需求细则

## **项目发起人同意组队申请**

在用户申请加入某项目后，若该项目的发起人同意该用户的加入，则可选择同意该用户的组队申请，该用户则成为该项目成员中的一员。

**所需数据：**项目ID、用户ID、用户昵称、需求状态

## **参与者申请参与组队**

用户如果看到了相应的组队需求并且想加入组队，那么可以在平台上申请参与组队。

**所需数据：**组队需求ID、用户ID、用户联系方式、用户专业、用户年级

## **参与者参与/退出项目**

用户在浏览项目内容之后，可以选择参与该项目。当用户已成为项目参与者时，可以选择退出该项目。

**所需数据：**用户ID、项目ID

## **管理员审核上传的项目信息**

用户上传项目信息后，待审核的项目将按照一定的规则分配给任意一个管理员账号，此时项目为待审核，管理员浏览该项目的基本信息并给出审核结果。通过后，项目才能正式发布。如果不通过，系统发送审核不通过的信息，项目发布失败。

**所需数据：**项目ID、项目信息、项目发布时间、审核状态、处理人ID、处理时间

## **管理员操作公告**

管理员可以在公告栏创建/发布/删除公告，公告可以用来作为推广和发布通知。

**所需数据：管理员ID、公告ID、**公告发布时间、公告持续时间、公告标题、公告内容

## **平台记录/展示用户参与的历史项目信息**

系统会记录用户参与过的历史项目，方便用户找到已参与的项目并回顾其内容。其他用户也可通过参与用户的历史记录，详细了解该项目。

**所需数据：**用户ID、项目ID、项目名称、项目背景、项目简介、项目内容、项目状态、立项时间、预计结项时间、当前项目进度

## **用户在项目结束后与项目成员互评打分**

用户在项目结束后可以对项目成员进行互评打分，包含评价和评价时间。

**所需数据：**用户ID、项目ID、评价内容、评价时间

## **平台记录项目进度**

对于每一个平台发布的项目，如果项目相关负责人在平台提交了相关项目的进度，则平台会进行记录。

**所需数据：**项目ID 、项目状态 、当前项目进度

## **项目发起人对参与者分配个人任务**

在项目进行的过程中，项目发起人给每个项目参与者分配相应的任务。

**所需数据：**项目ID、用户ID、任务内容

所涉及到的**数据流**如下：

#### 数据流1：用户信息

来源：注册账号、填写个人资料

去向：录入系统

组成：用户ID、密码、账号创建时间、昵称、性别、联系方式、就读院校、专业、年级、个人简介

#### 数据流2：项目信息

来源：用户在发布项目时填写或是在项目发布后对项目信息进行修改

去向：录入系统

组成：项目ID、项目名称、项目背景、项目简介、项目内容、项目状态、立项时间、预计结项时间、当前项目进度

#### 数据流3：管理员信息

来源：注册账号、编辑管理员信息

去向：录入系统、管理员表

组成：管理员ID、管理员用户名、管理员密码、管理员账号创建时间、管理员联系方式

#### 数据流4：项目评价信息

来源：用户结束项目后对其的评价

去向：录入系统

组成：评价内容、评价时间、评价人、评价类型、评价的项目

#### 数据流5：组队需求信息

来源：组队发起

去向：录入系统

组成：**需求ID、**需求发起时间、组队目的、组队类型、人数要求、需求细则、需求状态

1. **概念结构设计阶段**

## 实体集及其属性

#### 用户

记录用户的具体信息，是进行本数据库管理的最基本和最重要的实体集。属性包括主码用户ID，每个用户只有一个ID；用户密码需要进行加密，不能在数据库中明文显示；账户创建时间是用户注册账号的时间；昵称、性别、联系方式、就读院校、专业、年级、个人简介均由用户自行输入。

#### 项目

用来存储项目从被发起到立项直到结束的全部信息。在项目没有结束的时候也存在修改项目信息的情况。属性包括：**项目ID、**项目名称、项目背景、项目简介、项目内容、项目状态、立项时间、预计结项时间、当前项目进度，其中项目ID为主码。

#### 管理员

管理员拥有高于用户的权限，可以对项目、用户、公告等进行管理，可进行审核、管理等功能，并在系统中对相应操作进行记录。属性包括：管理员ID、密码、账号创建时间、权限、昵称、性别、联系方式。主码为：管理员ID。

#### 项目评价

用来记录某一项的项目所获得的评价，包括项目评价内容，项目评价时间等。属性包括：项目评价ID、项目评价类型、项目评价时间、评价内容。

#### 组队需求

如果用户有组队需求，那么他就可以在平台上发布这个需求，因此组队需求这个实体集用于记录组队需求有关的信息，包括如下的属性：需求ID、需求发起时间、组队目的、组队类型、人数要求、需求细则、需求状态，其中需求ID为主码。

#### 公告

是由管理员发布，全体用户可看到的具有公示性的通知，用来发布一些重要事件。属性包括公告ID、公告标题、公告内容，后两者由管理员自行取拟。

#### 标签

用来对项目进行分类，当用户上传项目时会添加相应的标签。属性为：标签名称、标签类别，其中标签名称为主码。

## 实体集之间的联系

#### 用户和项目的参与关系

用户可以在项目结题后发表项目评价。一位用户可以对于不同的项目发表不同的项目评价，但针对每个项目评价，有且仅有一位用户能够发表，因此用户与项目评价是一对多的关系。

#### 用户和项目评价的发表关系

每一位用户可以对于不同的项目发表不同的项目评价，但针对每个项目评价，有且仅有一位用户能够发表。因此用户与项目评价是一对多的关系。

#### 用户和组队需求的关系

用户在组队之前可以在系统中发出组队需求。一个用户可以发出多个组队需求，而一个组队需求却只能由一个用户发起，因此用户和组队需求之间是一对多的关系。

#### 用户和用户之间的关系

用户与用户之间可以存在关系，比如好友、关注、拉黑等关系。一个用户与多个用户之间可以产生关系，因此用户与用户是多对多的关系。

#### 项目评价和项目的关系

一条项目评价只能针对某一个项目进行评论，而一个项目可能有着多条评论，因此项目评价和项目之间是多对一的关系。

#### 组队需求和项目的对应关系

组队需求只能与一个项目相对应，一个项目只能有一个组队需求，因此组队需求和项目是一对一的关系。

#### 项目和标签的关系

标签便于对项目进行分类和检索，一个项目可以有多个标签，一个标签可以对应多个项目，因此项目与标签是多对多关系。

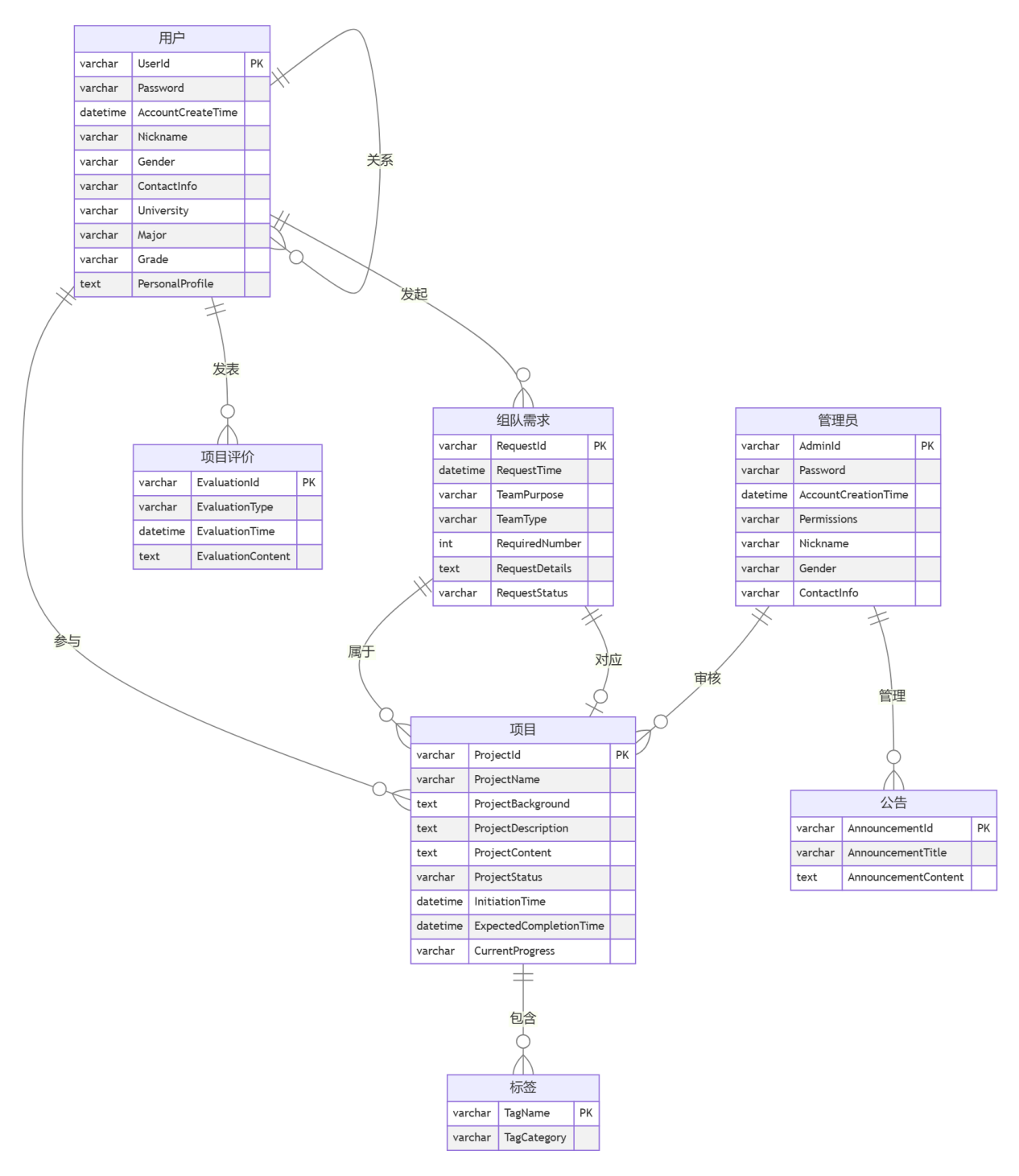
#### 管理员和公告的管理关系

管理员可以对公告进行管理。一位管理员可以管理多个公告，一条公告也可以由多位管理员管理，因此管理员与公告是多对多的关系。

#### 管理员和项目的审核关系

管理员可以对上传的项目进行审核。一位管理员可以审核多个项目，而一个项目是由一位管理员审核的，因此管理员和项目是一对多的关系。

## 相应的E-R图



1. **逻辑结构设计阶段**

## 关系模式的设计

用户表user(UserId, Password, AccountCreateTime, Nickname, Gender, ContactInfo, University, Major, Grade, PersonalProfile)

项目表 project(ProjectId, ProjectName, ProjectBackground, ProjectDescription, ProjectContent, ProjectStatus, InitiationTime, ExpectedCompletionTime, CurrentProgress, AuditResult, AuditTime)

组队需求表 grouping\_requirement(RequireId, RequestTime, TeamPurpose, TeamType, RequiredNumber, RequestDetails, RequestStatus, CreatorId, ProjectId)

公告表 announcement(AnnouncementId, AnnouncementTitle, AnnouncementContent, AdminId)

项目评价表 project\_evaluation(ProjectEvaId, EvaluatorId, ProjectId, EvaType, EvaTime, EvaContent)

管理员表 administrator(AdminId, Password, AccountCreationTime, Nickname, Gender, ContactInfo)

用户-项目参与表 user\_project(UserId, ProjectId, Duty)

好友表 friend(UserId, RelatedUserId, relation)

标签表 tag(TagName, TagCategory)

项目-标签表 project\_tag(ProjectId, TagName)

## 表的设计

该环节依据前述E-R图再调整。

### 用户表

user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| UserId | bigint |  | 用户ID | 主键，其实可以考虑使用int类型，满足绝大多数情况。使用bigint类型能满足极端情况。 |
| Password | varchar | 20 | 用户密码 | 用户密码需要进行加密，可以考虑使用hash和加salt值 |
| AccountCreateTime | datetime |  | 注册时间 | 添加默认值即可在用户生成时由DBMS填充此列 |
| Nickname | varchar | 20 | 用户名 | 由用户自行填写 |
| Gender | char | 2 | 用户性别 | 约束，男或女 |
| ContactInfo | varchar | 15 | 联系方式 | 电话号码，若使用邮箱则需适当增长 |
| University | varchar | 30 | 用户学校 |  |
| Major | varchar | 15 | 用户专业 |  |
| Grade | varchar | 10 | 用户年级 | 可以用int类型，此处考虑到灵活性选用varchar |
| PersonalProfile | varchar | 200 | 个人简介 |  |

### 项目表

project表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| ProjectId | bigint |  | 项目ID | 主键，与上述情况同理 |
| ProjectName | varchar | 60 | 项目名称 |  |
| ProjectBackground | varchar | 100 | 项目背景 |  |
| ProjectDescription | varchar | 300 | 项目简介 |  |
| ProjectContent | varchar | 10000 | 项目内容 |  |
| ProjectStatus | varchar | 10 | 项目状态 | 项目进度状态 |
| InitiationTime | datetime |  | 项目立项时间 |  |
| ExpectedCompletionTime | datetime |  | 项目预计结项时间 |  |
| CurrentProgress | varchar | 200 | 当前项目进度 | 对项目进度的描述 |
| AuditResult | char | 1 | 审核结果 | 缺省为0，1为通过，0为不通过 |
| AuditTime | datetime |  | 审核时间 |  |

### 组队需求表

grouping\_requirement表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| RequireId | bigint |  | 组队需求ID | 主键，与上述情况同理 |
| RequestTime | datetime |  | 需求发布时间 |  |
| TeamPurpose | varchar | 80 | 组队目的 | 描述组队目的 |
| TeamType | varchar | 16 | 组队类型 | 如：竞赛组队，课程项目组队 |
| RequiredNumber | varchar | 10 | 人数要求 | 记录人数区间，如：3-5 |
| RequestDetails | varchar | 1000 | 需求细则 | 描述组队需求的详细内容 |
| RequestStatus | char | 1 | 需求状态 | 缺省为0，0为无应答，1为招收中，2为已结束 |
| CreatorId | bigint |  | 发起人用户ID | 外键，参照user表的UserId属性 |
| ProjectId | bigint |  | 关联项目ID | 外键，参照project表的ProjectId属性 |

### 公告表

announcement表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| AnnouncementId | bigint |  | 公告ID | 主键，与上述情况同理 |
| AnnouncementTitle | varchar | 100 | 公告标题 |  |
| AnnouncementContent | text | 10000 | 公告内容 |  |
| AdminId | bigint | 20 | 管理该公告的管理员ID | 外键，参照administrator表的AdminId属性 |

### 项目评价表

project\_evaluation表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| ProjectEvaId | bigint |  | 项目评价ID | 主键，与上述情况同理 |
| EvaluatorId | varchar | 20 | 进行评价的用户 | 外键，参照user表的UserId属性 |
| ProjectId | varchar | 20 | 评价的项目ID | 外键,参照project表的ProjectId属性 |
| EvaType | char | 1 | 项目评价星级 | 好评还是差评，5为好评，1差评 |
| EvaTime | datetime |  | 评价时间 |  |
| EvaContent | varchar | 2000 | 评价内容 |  |

### 管理员表

administrator表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| AdminId | bigint |  | 管理员ID | 主键，与上述情况同理 |
| Password | varchar | 20 | 管理员密码 | 同上，需要进行加密 |
| AccountCreationTime | datetime |  | 注册时间 |  |
| Nickname | varchar | 20 | 用户名 |  |
| Gender | char | 1 | 性别（男/女） | 非空（缺省为1，表示男性） |
| ContactInfo | varchar | 15 | 联系方式 | 手机号，若为邮箱则需增长 |

### 用户-项目参与表

user\_project表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| UserId | bigint |  | 用户ID | 主键，外键，参照user表中的UserId属性 |
| ProjectId | bigint |  | 项目ID | 主键，外键，参照project表中的ProjectId属性 |
| Duty | varchar | 50 | 项目成员的职责 |  |

### 好友表

friend表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| UserId | bigint |  | 主体用户ID | 主键，外键，参照user表中的UserId属性 |
| RelatedUserId | bigint |  | 联系用户ID | 主键，外键，参照user表中的UserId属性 |
| relation | varchar | 10 | 用户之间关系的类型 | 如好友等 |

### 标签表

Tag表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| TagName | varchar | 20 | 标签名称 | 主键 |
| TagCategory | varchar | 10 | 标签类别 |  |

### 项目-标签表

project\_tag表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 说明 | 备注 |
| ProjectId | bigint |  | 项目ID | 主键，外键，参照project表中的ProjectId属性 |
| TagName | varchar | 20 | 标签内容 | 主键，外键，参照label表中的TagName属性 |

## 范式判断

用户表user(UserId, Password, AccountCreateTime, Nickname, Gender, ContactInfo, University, Major, Grade, PersonalProfile)

项目表 project(ProjectId, ProjectName, ProjectBackground, ProjectDescription, ProjectContent, ProjectStatus, InitiationTime, ExpectedCompletionTime, CurrentProgress, AuditResult, AuditTime)

组队需求表 grouping\_requirement(RequireId, RequestTime, TeamPurpose, TeamType, RequiredNumber, RequestDetails, RequestStatus, CreatorId, ProjectId)

公告表 announcement(AnnouncementId, AnnouncementTitle, AnnouncementContent, AdminId)

项目评价表 project\_evaluation(ProjectEvaId, EvaluatorId, ProjectId, EvaType, EvaTime, EvaContent)

管理员表 administrator(AdminId, Password, AccountCreationTime, Nickname, Gender, ContactInfo)

用户-项目参与表 user\_project(UserId, ProjectId, Duty)

好友表 friend(UserId, RelatedUserId, relation)

标签表 tag(TagName, TagCategory)

项目-标签表 project\_tag(ProjectId, TagName)

① **1NF：属性不可分割。**

由于每个元组的每个分量都满足原子性（属性不可再分），故满足1NF。

② **2NF：不存在非主属性对候选码的部分依赖。**

1. 用户表的候选码为UserId，后者均为全部依赖。
2. 项目表的候选码为ProjectId，后者均为全部依赖。
3. 组队需求表的候选码为RequireId，后者均为全部依赖。
4. 公告表的候选码为AnnouncementId，后者均为全部依赖。
5. 项目评价表的候选码为ProjectEvaId，后者均为全部依赖。
6. 管理员表的候选码为AdminId，后者均为全部依赖。
7. 用户-项目参与表的候选码为UserId和ProjectId，而Duty缺了两者之一均不成立，故不存在部分依赖。
8. 好友表的候选码为UserId和RelatedUserId，而relation缺了两者之一均不成立，故不存在部分依赖。
9. 标签表的候选码为TagName和TagCategory，不存在非主属性。

综上所述，本关系模式满足2NF。

③ **3NF：不存在非主属性对候选码的传递依赖。**

1. 用户表的候选码为UserId，后者信息均与候选码直接相关，使用候选码可推出所有非主属性，故不存在传递依赖。
2. 项目表的候选码为ProjectId，后者信息均与候选码直接相关，使用候选码可推出所有非主属性，故不存在传递依赖。
3. 组队需求表的候选码为RequireId，后者信息均与候选码直接相关，使用候选码可推出所有非主属性，故不存在传递依赖。
4. 公告表的候选码为AnnouncementId，后者信息均与候选码直接相关，使用候选码可推出所有非主属性，故不存在传递依赖。
5. 项目评价表的候选码为ProjectEvaId，考虑到一位用户能做出多次评价，后者信息均与候选码直接相关，使用候选码可推出所有非主属性，故不存在传递依赖。
6. 管理员表的候选码为AdminId，后者信息均与候选码直接相关，使用候选码可推出所有非主属性，故不存在传递依赖。
7. 用户-项目参与表的候选码为UserId和ProjectId，Duty与候选码直接相关，故不存在传递依赖。
8. 好友表的候选码为UserId和RelatedUserId，relation与候选码直接相关，故不存在传递依赖。
9. 标签表的候选码为TagName和TagCategory，不存在非主属性。

综上所述，本关系模式满足3NF。

**【实验结果与分析】**

根据上文已初步完成该数据库系统的设计，证明了对该数据库系统的逻辑结构设计满足3NF范式。此时的逻辑数据库依然有很多优化空间，例如具体采用怎样的DBMS（SQLite/MySQL/Oracle）、怎样添加锁以处理数据一致性问题、在数据量不断膨胀时怎样优化查询效率、怎样配置索引以提高数据库性能等等。

在构建相应的逻辑数据库后，完成前端、后端的设计即可将其部署，尝试投入使用。但同样面对问题：本身有服务器的情况下可以使用Django作为前后端框架，但难以处理并发问题；使用Serverless函数及Github Page能够将项目部署的流程极简化，但在站点访问量增大之后成本也不断提高——这是两难的问题，还需在以后不断思考。