数据库设计说明书

一、 引言

1. 编写目的

数据库表结构设计在软件项目开发过程中扮演着至关重要的角色。一个经过深思熟虑且精心规划的数据库架构不仅能够极大地提升项目的开发效率，还能显著简化后期系统维护工作，并为未来可能出现的功能需求扩展预留足够的灵活性与空间。因此，本文件旨在全面地从多个角度出发，对历年卷（或特定项目）所需的数据存储解决方案进行详尽的设计规划。这份文档将成为指导团队成员理解、实现以及维护整个系统数据层面内容的关键参考资料。
通过编写这样一份详细的数据库设计说明书，我们力求达到以下几个目标：

- **统一标准**：明确所有相关数据对象（如表格、字段等）的标准命名规则，确保一致性和可读性。

- **范围界定**：清晰定义每个数据实体所涵盖的信息边界，避免冗余同时保证信息完整性。
- **编码规范**：制定合理有效的数据类型选择指南及编码实践建议，以优化性能并减少错误发生的可能性。
  此外，该文档还承担着促进项目组内部沟通协作的重要职责。它为全体参与者提供了一个共同遵循的基础框架，其中包括但不限于：
- 开发过程中应遵守的最佳实践指南；
- 各个子系统之间以及不同职能团队成员间如何有效交换信息的具体规定；
- 统一采用的数据格式标准，这有助于消除误解，提高工作效率。

《数据库设计说明书》是面向包括但不限于数据库设计师、数据库管理员以及参与该项目的所有成员的一份核心资料。它不仅提供了关于如何构建高效稳定数据库环境的专业知识，同时也强调了在整个软件生命周期内保持良好沟通与合作的重要性。通过严格遵守这份文档中提出的各项原则与建议，我们相信能够极大程度上保障软件产品的质量，加速项目进度，并最终满足甚至超越客户的期望值。

预期的读者：数据库设计师，数据库管理员，项目成员

2. 背景说明

项目概述

- **项目名称**：卷卷福
- **开发者团队**：先天软工圣体队
- **目标用户群体**：福州大学的学生

项目背景

随着信息技术的快速发展以及高校学生对于便捷高效学习工具需求的增长，“卷卷福”应运而生。该项目旨在通过结合最新的软件工程技术与创新的设计理念，为福州大学的学生提供一个集成了资料分享、在线交流等功能于一体的综合服务平台。它不仅能够帮助学生们更好地组织日常的学习生活，还能够在一定程度上促进师生之间以及同学之间的沟通与合作。

3. 定义

词汇名称	词汇定义	备注

数据库	用来保存系统数据的后台应用软件	数据处理的主要内容是数据的存储、查询、修改、排序和统计等
C/S	客户/服务器	Client/Server的缩写
表	不同字段汇总成的集合	由行（row）和列(column)组成，是一个二维的网格结构，每个列都是一个字段
E-R图	实体-联系图，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型	Entity -Relationship 的缩写
UML图	UML是一个通用的标准建模语言，可以对任何具有静态结构和动态行为的系统进行建模，而且适用于系统开发的不同阶段，从需求规格描述直至系统完成后的测试和维护	

4. 参考资料

1. 卷卷福选题报告

2024秋软工实践《先天软工圣体队》团队展示与《卷卷福》选题报告 - DriOgon - 博客园

2. 《数据库系统概论》（第6版）

3. 数据库设计说明书的编写

https://blog.csdn.net/lzj_1314/article/details/92689586

二、外部设计

1. 标识符和状态

数据库软件的名称：Mysql

数据库的名称：juanjuanfu

表名	标识符或名称	描述信息	状态

用户表（Users）	UserID	用来保存用户信息	使用
资源表（Resources）	ResouceID	用来保存历年卷信息	使用
资料库表（Librarirs）	LibraryID	用来保存资源库信息	使用
交互记录表（Interactions）	InteractionID	用来保存历年卷下载、上传信息	使用
评论表（Comments）	CommentID	用来保存发布的评论	使用
排行榜表（Rankings）	RankingID	用来保存更新排行榜	使用
发帖表（Posts）	PostID	用来保存发布的帖子	使用
社区表（Communities）	CommunityID	用来保存社区信息	使用

2. 使用它的程序

应用程序	访问的数据表	版本号
登录注册界面	用户表	1.0
社区界面	发帖表，社区表，评论表，排行榜表	1.0
资料库界面	资料库表，资源表	1.0
我的界面	交互记录表，发帖表	1.0

3. 约定

文卷	记录	数据项的命名约定
用户表	Users	无
资源表	Resources	无

资料库表	Librarirs	无
交互记录表	Interactions	无
评论表	Comments	无
排行榜表	Rankings	无
发帖表	Posts	无
社区表	Communities	无

4. 支持软件

支持软件名称	版本号	主要功能
Mysql	5.7	建立数据库并提供数据库维护与管理功能
Django	2.2.1	后端框架，为小程序提供数据库的技术支持
Python	3.6.5	代码编写
Javascript		脚本语言，提供HTTP的技术支持
navicat		管理员工具

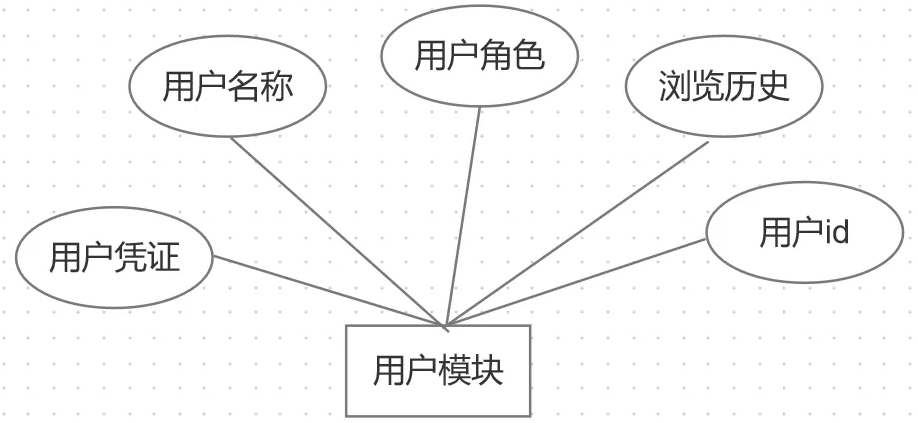
三、结构设计

1.　E-R图和模块设计

说明本数据库将反映的现实世界中的实体、属性和它们之间的关系等的原始数据形式，包括各数据项、记录、系、文卷的标识符、定义、类型、度量单位和值域，建立本数据库的每一幅用户视图。

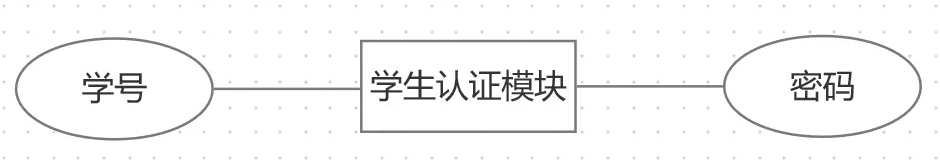
1.1用户模块

用户信息（用户凭证、用户名称、浏览历史、用户id、用户角色）



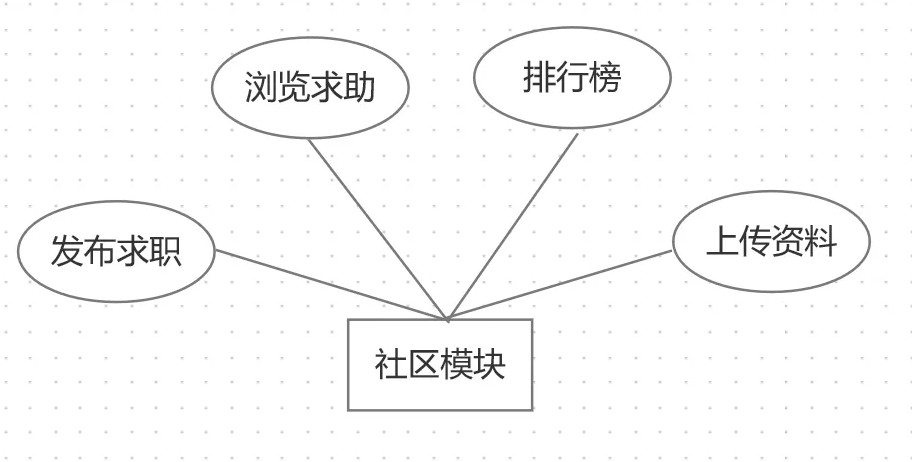
1.2学生认证模块

学生认证（学号、密码）



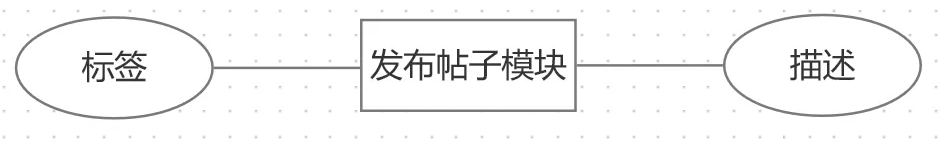
1.3社区模块

社区（浏览求助、上传资料、发布求职、排行榜）



1.4发布帖子模块

发布帖子（标签、描述）

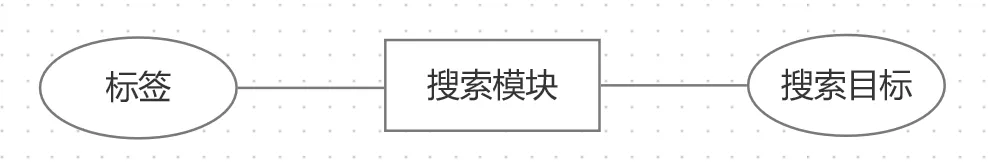


1.5评论模块

发表评论（用户id、评论内容）

1.6搜索模块

搜索（搜索目标、标签）

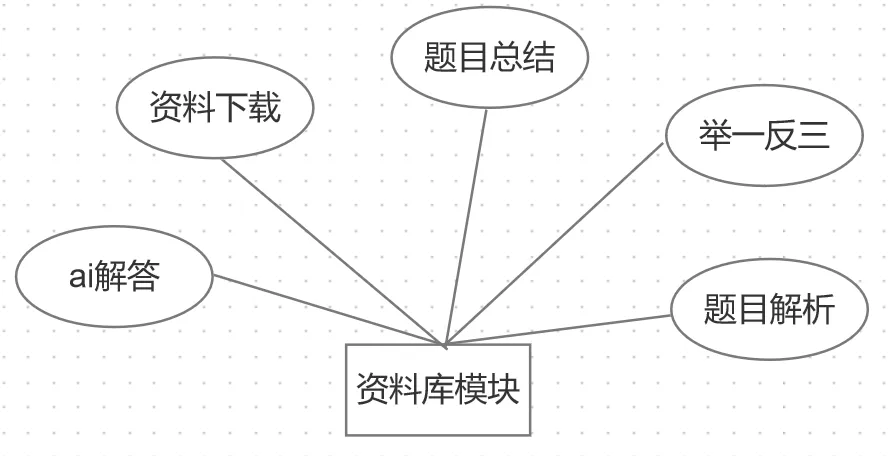


其中属性约束为：

搜索目标为系统提供的标签选项，用户不能自己添加

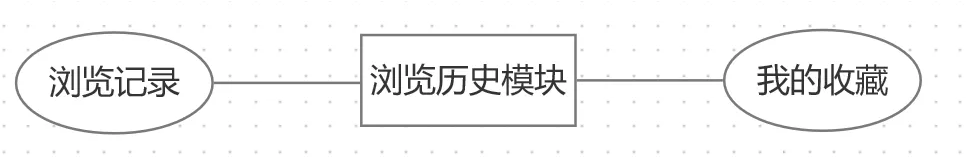
1.7资料库模块

资料库（题目总结、题目解析、举一反三、ai解答、资料下载）

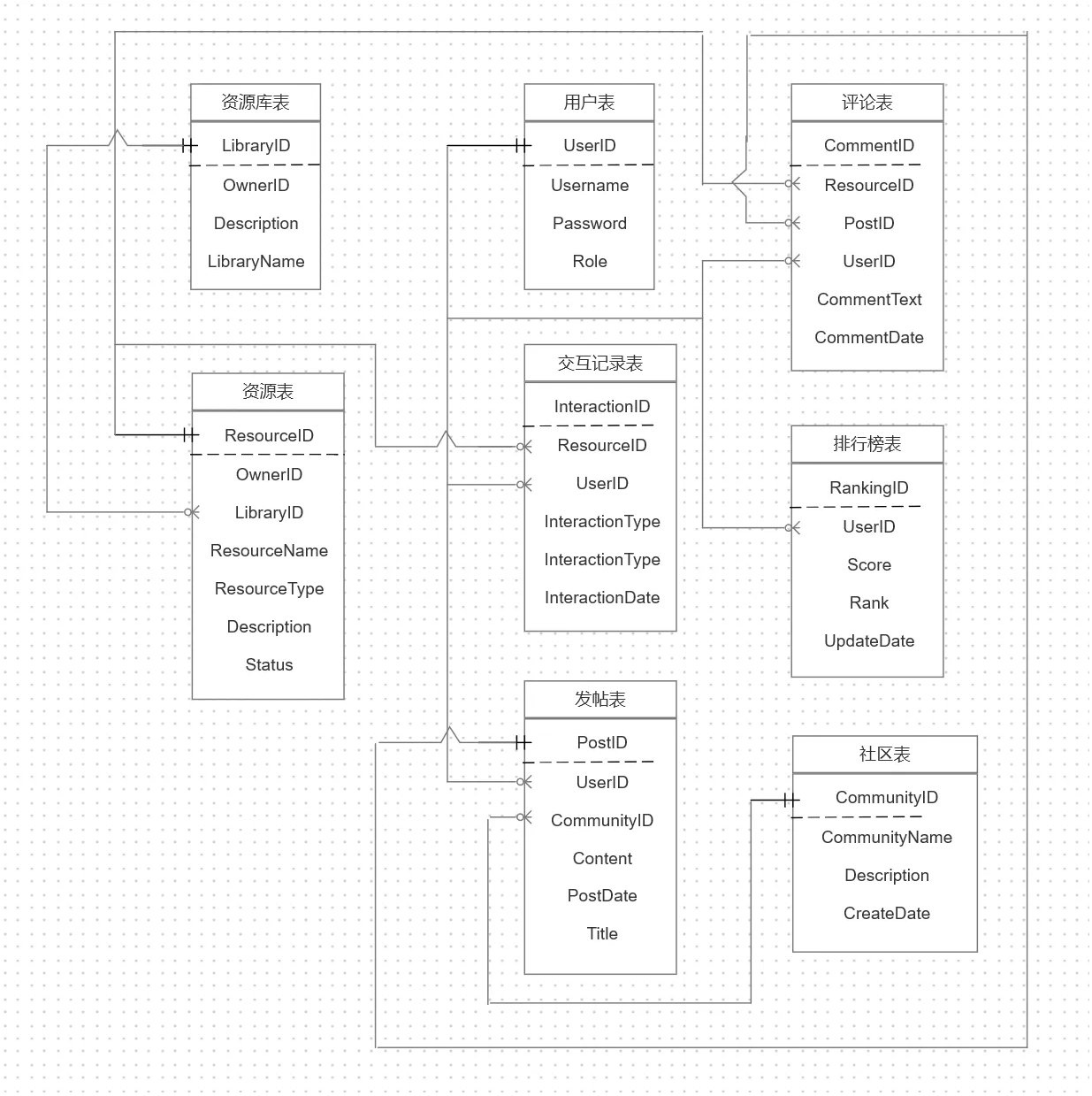


1.8浏览历史模块

浏览历史（浏览记录、我的收藏）



1.9全局E-R图



2. 逻辑结构设计

2.1 用户表 (Users)

字段名	数据类型	长度	主键	非空	描述

UserID	INT	–	是	是	用户唯一标识
Username	VARCHAR	50	否	是	用户名
Password	VARCHAR	255	否	是	用户密码（加密存储）
Role	VARCHAR	20	否	否	用户角色（用户、管理员等）

2.2 资源表 (Resources)

字段名	数据类型	长度	主键	非空	描述
ResourceID	INT	–	是	是	资源唯一标识
ResourceName	VARCHAR	100	否	是	资源名称
ResourceType	VARCHAR	50	否	是	资源类型
OwnerID	INT	–	否	是	资源拥有者（用户ID）
LibraryID	INT	–	否	是	资料库ID
Description	TEXT	–	否	否	资源描述
Status	VARCHAR	20	否	是	资源状态

2.3 资料库表 (Libraries)

字段名	数据类型	长度	主键	非空	描述
LibraryID	INT	–	是	是	资料库唯一标识
LibraryName	VARCHAR	100	否	是	资料库名称
Description	TEXT	–	否	否	资料库描述

OwnerID	INT	–	否	是	资料库拥有者（用户 ID）

2.4 交互记录表 (Interactions)

字段名	数据类型	长度	主键	非空	描述
InteractionID	INT	–	是	是	交互记录唯一标识
ResourceID	INT	–	否	是	资源ID（指向资源表）
UserID	INT	–	否	是	用户ID（指向用户表）
InteractionType	VARCHAR	50	否	是	交互类型（上传、下载等）
InteractionDate	DATETIME	–	否	是	交互日期

2.5 评论表 (Comments)

字段名	数据类型	长度	主键	非空	描述
CommentID	INT	–	是	是	评论唯一标识
ResourceID	INT	–	否	是	资源ID（指向资源表）
PostID	INT	–	是	是	帖子ID（指向帖子表）
UserID	INT	–	否	是	用户ID（指向用户表）
CommentText	TEXT	–	否	是	评论内容
CommentDate	DATETIME	–	否	是	评论日期

2.6 排行榜表 (Rankings)

字段名	数据类型	长度	主键	非空	描述
RankingID	INT	–	是	是	排行榜唯一标识
UserID	INT	–	否	是	用户ID（指向用户表）
Score	INT	–	否	是	用户得分
Rank	INT	–	否	是	用户在排行榜中的排名
UpdateDate	DATETIME	–	否	是	排行榜更新时间

2.7 发帖表 (Posts)

字段名	数据类型	长度	主键	非空	描述
PostID	INT	–	是	是	帖子唯一标识
UserID	INT	–	否	是	用户ID（指向用户表）
Title	VARCHAR	150	否	是	帖子标题
Content	TEXT	–	否	是	帖子内容
PostDate	DATETIME	–	否	是	帖子发布日期
CommunityID	INT	–	否	是	社区ID（指向社区表）

2.8 社区表 (Communities)

字段名	数据类型	长度	主键	非空	描述

CommunityID	INT	–	是	是	社区唯一标识
CommunityName	VARCHAR	100	否	是	社区名称
Description	TEXT	–	否	否	社区描述
CreateDate	DATETIME	–	否	是	社区创建日期

3. 对象关系映射

下面使用 SQL语言 设计并得到数据库和相应的表

3.1创建Users表

Plain Text

```
CREATE TABLE Users (
    UserID INT PRIMARY KEY NOT NULL,
    Username VARCHAR(50) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    Role VARCHAR(20)
);

```

3.2创建Resources表

```
CREATE TABLE Resources (
    ResourceID INT PRIMARY KEY NOT NULL,
    ResourceName VARCHAR(100) NOT NULL,
    ResourceType VARCHAR(50) NOT NULL,
    OwnerID INT NOT NULL,
    LibraryID INT NOT NULL,
    Description TEXT,
    Status VARCHAR(20) NOT NULL
);
```

3.3创建Libraries表

```
CREATE TABLE Libraries (
    LibraryID INT PRIMARY KEY NOT NULL,
    LibraryName VARCHAR(100) NOT NULL,
    Description TEXT,
    OwnerID INT NOT NULL
);
```

3.4创建Interactions表

```
CREATE TABLE Interactions (
    InteractionID INT PRIMARY KEY NOT NULL,
    ResourceID INT NOT NULL,
    UserID INT NOT NULL,
    InteractionType VARCHAR(50) NOT NULL,
    InteractionDate DATETIME NOT NULL
);
```

3.5创建Comments表

```
CREATE TABLE Comments (
    CommentID INT PRIMARY KEY NOT NULL,
    ResourceID INT NOT NULL,
    PostID INT NOT NULL,
    UserID INT NOT NULL,
    CommentText TEXT NOT NULL,
    CommentDate DATETIME NOT NULL
);

```

3.6.创建Rankings表

```
CREATE TABLE Rankings (
    RankingID INT PRIMARY KEY NOT NULL,
    UserID INT NOT NULL,
    Score INT NOT NULL,
    Rank INT NOT NULL,
    UpdateDate DATETIME NOT NULL
);

```

3.7创建Posts表

```
CREATE TABLE Posts (
    PostID INT PRIMARY KEY NOT NULL,
    UserID INT NOT NULL,
    Title VARCHAR(150) NOT NULL,
    Content TEXT NOT NULL,
    PostDate DATETIME NOT NULL,
    CommunityID INT NOT NULL
);
```

3.8创建Communities表

```
CREATE TABLE Communities (
    CommunityID INT PRIMARY KEY NOT NULL,
    CommunityName VARCHAR(100) NOT NULL,
    Description TEXT,
    CreateDate DATETIME NOT NULL
);
```

四、运用设计

1. 安全保密设计

数据库由专门数据库管理员对数据库操作，需要注意以下几项安全问题：访问安全、网络安全、传输安全、备份安全、数据安全

以下是为数据库设计的安全规则，以确保系统安全性、数据完整性和用户隐私保护：

1.1基于用户角色的访问控制

访问者	权限	访问类型

管理员	全部权限	全表的查询、插入、更新、删除
普通用户	社区、发布讨论、搜索、查询等	POST

1.2存储加密

为了确保用户数据的安全性和隐私性，对于存储在数据库中的敏感信息，比如用户的密码和个人信息等，采取加密存储的方式是非常必要的。这不仅有助于保护用户免受潜在的数据泄露威胁，还能增强整个系统的安全性。

1. **选择合适的加密算法**：首先需要选择一个强大且被广泛认可的加密算法来对敏感信息进行加密处理。对于密码而言，推荐使用像bcrypt、scrypt或Argon2这样的哈希函数而不是简单的MD5或SHA-1。这些专门设计用于密码存储的算法能够提供更强的抗破解能力，并且通常会引入加盐机制（salt），即向每个密码添加随机数据后再进行哈希处理，这样即使两个用户设置了相同的密码，它们在数据库中也会表现为不同的哈希值。
2. **安全地管理密钥**：如果使用了对称加密技术来保护其他类型的个人信息，则还需要妥善保管好用来解密数据的密钥。理想情况下，应将密钥存储在一个与主要数据库隔离的安全位置，并限制对其访问权限。
3. **实现多层次的安全措施**：除了直接对数据本身进行加密之外，还应该考虑实施其他形式的安全防护策略，例如设置强密码策略、启用双因素认证等，以此来进一步提高账户安全性。
4. **定期更新加密方案**：随着技术的发展，新的攻击手段不断出现，因此重要的是要定期审查和更新所使用的加密技术和流程，确保它们始终处于行业最佳实践水平之上。
   通过上述措施，可以有效地保护存储于数据库内的敏感信息不受未经授权的访问或窃取，从而维护用户的利益及企业信誉。

1.3异常行为监控

在监控系统中，对异常行为的监测是确保信息安全的重要环节之一。这类异常行为包括但不限于多次失败的登录尝试、未经授权的访问尝试等情形。当用户连续多次输入错误密码或使用无效凭证尝试登录时，这可能表明有人正在试图通过暴力破解等方式非法进入系统；而未经授权的访问尝试则指的是未被授权的个人或程序试图获取受保护资源的行为，这也是一种常见的安全威胁。

面对这些异常情况，管理员应当迅速响应并采取适当的措施来防止潜在的安全风险进一步扩大。首先，可以立即锁定相关账户以阻止更多的非法登录尝试，并对涉及的IP地址进行调查和必要时加以封锁。其次，对于任何疑似未经授权的访问活动，应详细记录下来，并启动内部审查流程以查明事件的具体原因及其影响范围。此外，加强系统的安全性设置也是非常必要的一步，比如启用双因素认证机制、

定期更新密码策略以及增强防火墙规则等，以此提高整体防护水平。

总之，在发现任何可疑活动后及时介入处理，并通过持续优化安全管理措施来降低未来发生类似问题的风险，这是每个组织都必须重视的任务。同时，也建议企业加强对员工的信息安全意识培训，让他们了解如何识别潜在威胁及正确应对方法，从而形成一个全面的安全防御体系。

1.4 定期进行数据库备份

定期进行数据库备份是一项至关重要的任务，它能够帮助企业或个人在面对数据丢失或系统故障时迅速恢复信息。这一过程不仅有助于保护重要资料免受意外删除、硬件损坏或是恶意软件攻击的影响，而且还能确保业务连续性，减少因数据不可用而造成的潜在经济损失和服务中断。为了实现这一点，建议制定一个详细的备份计划，包括但不限于选择合适的备份频率（如每日、每周）、确定需要备份的关键数据集以及采用可靠的存储解决方案来保存这些备份副本。此外，还应该定期测试备份文件的有效性和完整性，以保证当真正需要使用它们来进行恢复操作时，可以顺利地完成整个过程。总之，通过实施有效的数据库备份策略，组织和个人都能够更好地应对突发情况，维护其数字资产的安全与稳定。

1.5更新数据库及系统组件

为了保障系统和数据的安全性，企业或组织需要定期对数据库软件以及服务器操作系统进行更新。这不仅仅包括安装最新的补丁来修复已知的安全漏洞，还意味着要升级到最新版本的软件，以利用其中增强的安全特性、性能改进以及其他功能上的优化。通过保持所有组件处于最新状态，可以有效降低遭受恶意攻击的风险，同时也能提高整体系统的稳定性和效率。此外，建议启用自动更新机制（如果适用），这样可以在第一时间获得官方发布的安全更新，进一步加强防护措施。对于那些无法立即执行全面更新的情况，则应密切关注相关产品的安全公告，并根据实际情况尽快安排必要的维护工作。总之，持续关注并及时响应厂商提供的任何安全信息，是维护一个健康且安全的IT环境的关键步骤之一。