数据库设计报告

Version 2.0 ● 2021.2.22





**数据库设计报告**

**【融销通**】

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  [ ] 正在修改 | 文件标识： | QST-22软件工程1班-项目代码-DOC-DB |
| 当前版本： | 1.0 |
| 作 者： |  |
| 完成日期： | 2024-11-5 |
| 批 准 人： |  |
| 批准日期： | 2024-11-5 |
| 签 字： |  |

【扛把子小组】

**变更历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **版本** | **变更日期** | **变更内容** | **变更者** |
| 1 | 1.1 | 2024.11.12 | 目录变更 | 崔学童 |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
|  |  |  |  |  |

目录

[1.文档介绍 4](#_Toc182297686)

[1.1文档目的 4](#_Toc182297687)

[1.2 文档范围 4](#_Toc182297688)

[1.3 读者对象 4](#_Toc182297689)

[1.4 参考文献 5](#_Toc182297690)

[1.5术语与缩写解释 5](#_Toc182297691)

[2. 数据库环境说明 6](#_Toc182297692)

[2.1数据库技术选型 6](#_Toc182297693)

[2.2硬件环境 6](#_Toc182297694)

[2.3软件环境 6](#_Toc182297695)

[2.4数据安全性 7](#_Toc182297696)

[2.5数据一致性 7](#_Toc182297697)

[2.6性能优化 7](#_Toc182297698)

[3.数据库的命名规则 8](#_Toc182297699)

[3.1引言 8](#_Toc182297700)

[3.2命名原则 8](#_Toc182297701)

[3.3数据库对象命名规则 8](#_Toc182297702)

[3.4命名规则与机构标准的关系 9](#_Toc182297703)

[3.5结论 10](#_Toc182297704)

[4.系统数据分布 10](#_Toc182297705)

[4.1数据分布特点 10](#_Toc182297706)

[4.1.1分布式存储 10](#_Toc182297707)

[4.1.2数据冗余与备份： 10](#_Toc182297708)

[4.1.3数据一致性： 10](#_Toc182297709)

[4.2数据分布模式 11](#_Toc182297710)

[4.2.1主从复制 11](#_Toc182297711)

[4.2.2分片（Sharding） 11](#_Toc182297712)

[4.2.3分布式数据库 11](#_Toc182297713)

[4.3网络环境下的数据访问 11](#_Toc182297714)

[4.3.1负载均衡 11](#_Toc182297715)

[4.3.2数据缓存 12](#_Toc182297716)

[4.3.3数据同步与复制 12](#_Toc182297717)

[4.4示例说明 12](#_Toc182297718)

[5.逻辑结构设计 12](#_Toc182297719)

[7.安全性设计 21](#_Toc182297720)

[7.1 防止用户直接操作数据库的方法 21](#_Toc182297721)

[7.2 用户帐号密码的加密方法 21](#_Toc182297722)

[7.3 角色与权限 21](#_Toc182297723)

[8.优化 22](#_Toc182297724)

[9.数据库管理与维护说明 22](#_Toc182297725)

[9.1 数据备份 22](#_Toc182297726)

[9.2 权限管理 23](#_Toc182297727)

[9.3 日志管理 23](#_Toc182297728)

[9.4 性能监控 23](#_Toc182297729)

[9.5数据清理和归档 23](#_Toc182297730)

[9.6 索引维护 23](#_Toc182297731)

[9.7安全更新与漏洞修补 24](#_Toc182297732)

[9.8文档管理 24](#_Toc182297733)

# 1.文档介绍

## 1.1文档目的

本文档旨在详细描述“基于数字经济的农产品融销一体平台”（简称“融销通”）的数据库设计。通过明确数据库的结构、关系及各项属性，确保系统数据的有效存储、高效查询和良好维护，从而为平台的融资、销售、种植指导等功能提供坚实的数据支撑。

## 1.2 文档范围

* 数据库设计总体思路
* 主要实体及其属性定义
* 实体间的关系模型
* 数据库表结构设计
* 数据字典
* 数据安全与备份策略
* 索引与性能优化建议

## 1.3 读者对象

本文档面向以下读者：

* 系统开发团队：包括前端、后端、数据库开发人员及测试人员。
* 系统管理员：负责数据库的日常运维和监控。
* 项目经理：需要对数据库设计进行整体把控和评估。
* 相关领域专家：提供农业知识和种植指导的专家，需要了解数据库中存储的农业知识数据结构。

## 1.4 参考文献

[A01] 《SpringBoot框架使用指南》，清华大学出版社，2022年

[A02] 《Vue.js前端开发实战》，机械工业出版社，2021年

[A03] 《数据库系统概论》，高等教育出版社，2019年

[PPP-SYS-ARCH] 项目组，“融销通平台系统架构设计文档”，项目内部文档，2023年

## 1.5术语与缩写解释

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| **B/S架构** | Browser/Server，浏览器/服务器架构 |
| **SpringBoot** | 一种用于简化Spring应用的初始搭建和开发过程的框架 |
| **Vue** | 一种用于构建用户界面的渐进式JavaScript框架 |
| **融销通** | 基于数字经济的农产品融销一体平台的简称 |
| **农户** | 在平台上进行商品买卖、专家咨询、融资申请的用户 |
| **专家** | 提供农业知识管理、处理农业问答信息的用户 |
| **管理员** | 负责管理系统用户、商品等资源的用户 |
| **银行用户** | 进行融资管理的用户 |

# 2. 数据库环境说明

## 2.1数据库技术选型

* 数据库类型：考虑到“融销通”平台需要处理大量用户数据、交易数据以及融资信息，选择关系型数据库（如MySQL5.6.7）作为数据存储的核心，以确保数据的一致性和完整性，并用navicat进行操作。
* 数据库架构：采用主从复制或分布式数据库架构，以提高数据库的可用性和可扩展性。主数据库负责处理读写请求，从数据库负责备份和只读请求，以分散负载并提高性能。

## 2.2硬件环境

* 服务器配置：选择高性能的服务器硬件，包括高速CPU、大容量内存和高速存储设备（如SSD），以确保数据库的高效运行。
* 网络环境：确保服务器之间的网络连接稳定且高速，以减少数据传输延迟并提高系统响应速度。

## 2.3软件环境

* 操作系统：选择稳定且安全的操作系统（如Linux），以提供可靠的运行环境。
* 数据库管理系统（DBMS）：安装并配置MySQL数据库管理系统，以管理数据库用户、权限、备份和恢复等操作。
* 开发工具：使用SpringBoot+Vue前后端分离技术开发，确保平台的前后端能够高效协作并快速响应用户需求。

## 2.4数据安全性

* 数据加密：对敏感数据（如用户密码、融资信息等）进行加密存储，以防止数据泄露。
* 访问控制：实施严格的访问控制策略，确保只有授权用户才能访问和操作数据库。
* 备份与恢复：定期备份数据库数据，并测试备份数据的恢复能力，以应对可能的数据丢失或损坏情况。

## 2.5数据一致性

* 事务处理：使用数据库事务来确保数据的一致性，即在多个操作之间保持数据的原子性、一致性、隔离性和持久性。
* 锁机制：在需要时使用锁机制来防止数据并发访问时的冲突和不一致情况。

## 2.6性能优化

* 索引优化：为常用的查询字段建立索引，以提高查询速度。
* 查询优化：对复杂的查询语句进行优化，以减少查询时间和资源消耗。
* 缓存机制：使用缓存机制来减少数据库的访问次数，提高系统性能。

# 3.数据库的命名规则

## 3.1引言

为确保“基于数字经济的农产品融销一体平台”（简称“融销通”）的数据库设计具有一致性、可读性和可维护性，特制定本数据库的命名规则。本规则旨在明确数据库对象（如表、视图、索引、存储过程、触发器等）的命名方式，以及命名中应遵循的原则。

## 3.2命名原则

* **简洁明了**：命名应简洁明了，避免使用过长或复杂的名称，以便于理解和记忆。
* **有意义**：命名应具有实际意义，能够准确反映数据库对象的用途或功能。
* **一致性**：命名应保持一致性，避免使用不同的命名风格或术语。
* **可读性**：命名应使用易读的字符和词汇，避免使用特殊字符或难以理解的缩写。

## 3.3数据库对象命名规则

* **数据库名**：格式：<项目简称>\_<环境标识>
  + 示例：rongxiaotong\_prod（生产环境）、rongxiaotong\_test（测试环境）
* **表名**：格式：<模块名>\_<功能名>\_<表类型>（可选）
  + 示例：user\_info（用户信息表）、product\_category（产品分类表）
  + 注意：表名应使用复数形式（如orders而非order），以表示表中存储的是多条记录。
* **字段名**：格式：<属性名>\_<数据类型>（可选）
  + 示例：user\_name（用户名）、product\_price（产品价格）
  + 注意：字段名应使用小写字母和下划线进行分隔，避免使用驼峰命名法。
* **视图名**：格式：v\_<模块名>\_<功能名>
  + 示例：v\_user\_orders（用户订单视图）
* **索引名**：格式：idx\_<表名>\_<字段名>\_<类型>（可选）
  + 示例：idx\_user\_info\_user\_name\_unique（用户信息表中的用户名唯一索引）
* **存储过程名**：格式：sp\_<模块名>\_<功能名>
  + 示例：sp\_user\_register（用户注册存储过程）
* **触发器名**：格式：trg\_<表名>\_<触发事件>\_<操作>
  + 示例：trg\_orders\_after\_insert（订单表插入后触发器）

## 3.4命名规则与机构标准的关系

本数据库的命名规则与机构的标准命名规则在大部分情况下保持一致，但在某些细节上可能有所不同。例如，机构标准可能要求所有数据库对象名称都使用大写字母，而本规则则建议使用小写字母和下划线进行分隔，以提高可读性和易读性。此外，本规则还根据“融销通”平台的具体需求，对表名、字段名等进行了更具体的命名规范。这些差异是为了更好地适应本平台的特点和需求，同时确保命名的一致性和可读性。

## 3.5结论

本数据库的命名规则旨在确保“融销通”平台的数据库设计具有一致性、可读性和可维护性。通过遵循这些规则，我们可以确保数据库对象的命名清晰明了、易于理解和记忆，从而提高数据库的设计质量和维护效率。

# 4.系统数据分布

## 4.1数据分布特点

### 4.1.1分布式存储

* 数据被分散存储在多个节点上，每个节点独立管理自己的数据。
* 这种分布方式有助于减少单个节点的负载，提高系统的整体性能和可靠性。

### 4.1.2数据冗余与备份：

* 为了确保数据的安全性和可用性，数据通常会在多个节点上进行冗余存储和备份。
* 这意味着即使某个节点发生故障，其他节点上的数据仍然可以保证系统的正常运行。

### 4.1.3数据一致性：

* 在分布式系统中，保持数据的一致性是一个挑战。
* 通常通过分布式事务、数据复制和同步等技术来确保数据在不同节点之间的一致性。

## 4.2数据分布模式

### 4.2.1主从复制

* 主节点负责处理读写请求，并将数据变化同步到从节点。
* 从节点通常只处理只读请求，以分担主节点的负载。
* 这种模式在读写分离和负载均衡方面非常有效。

### 4.2.2分片（Sharding）

* 数据被划分为多个片段，每个片段存储在不同的节点上。
* 通过分片技术，可以水平扩展系统的存储和计算能力。
* 这种模式适用于大规模数据的存储和处理。

### 4.2.3分布式数据库

* 如Cassandra、HBase等，它们将数据分散存储在多个节点上，并提供跨节点的数据查询和更新功能。
* 这些数据库通常具有高度的可扩展性和容错性。

## 4.3网络环境下的数据访问

### 4.3.1负载均衡

* 通过负载均衡器将用户的请求分发到不同的节点上，以平衡负载并提高系统的响应速度。
* 负载均衡器可以根据节点的负载情况、网络延迟等因素来智能地选择最佳的节点。

### 4.3.2数据缓存

* 在网络节点上设置数据缓存层，以加速数据的访问速度。
* 缓存层可以存储常用的数据或查询结果，从而减少对数据库的访问次数。

### 4.3.3数据同步与复制

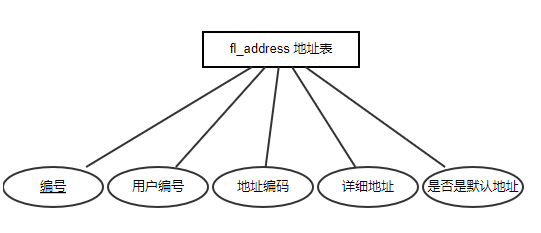
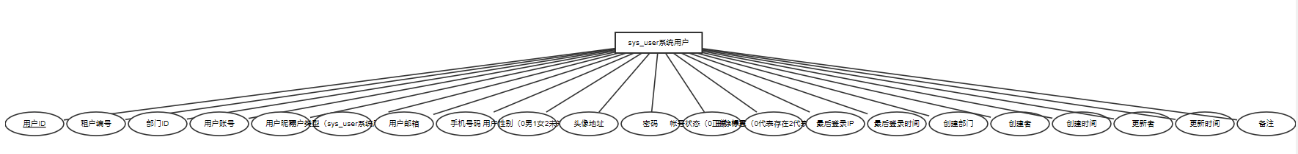
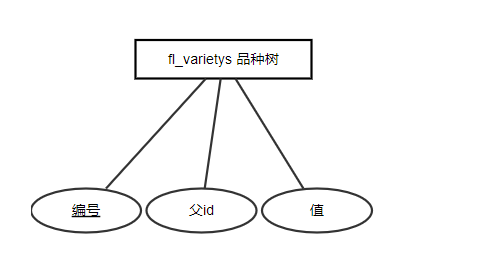
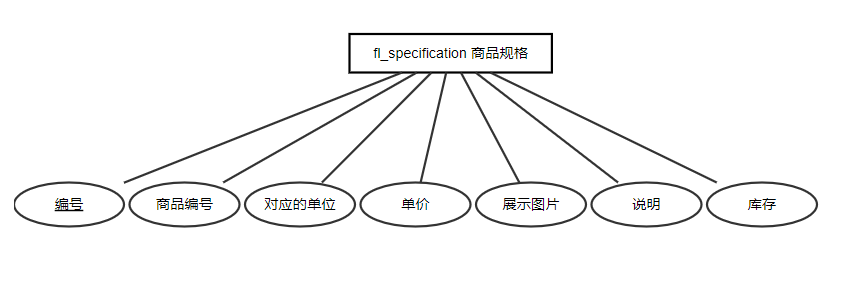
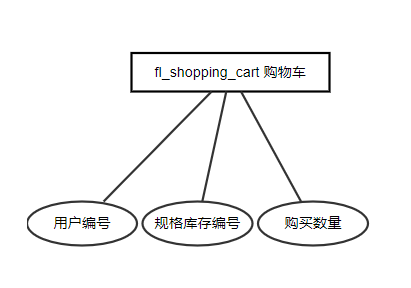
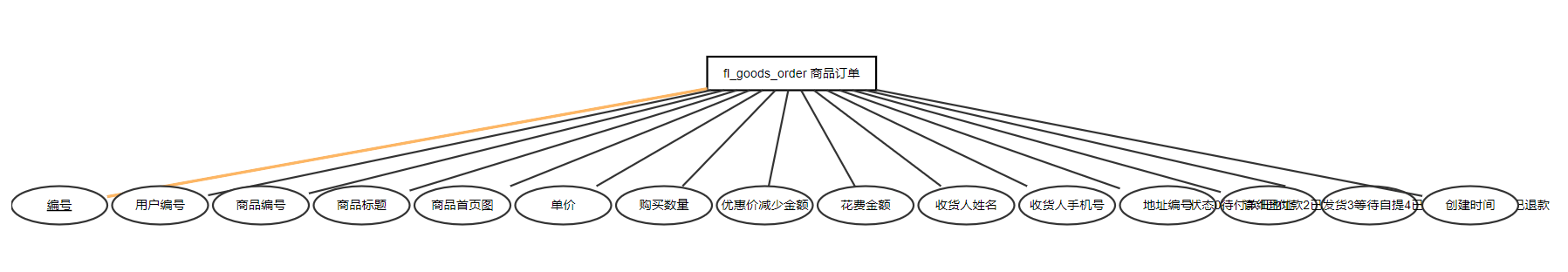
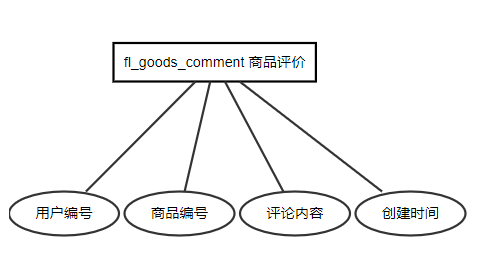
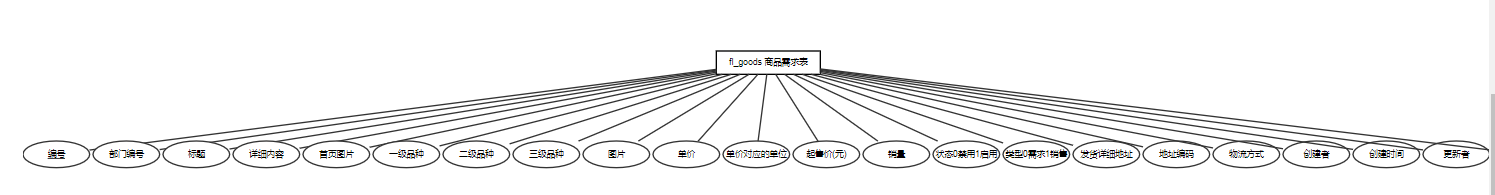
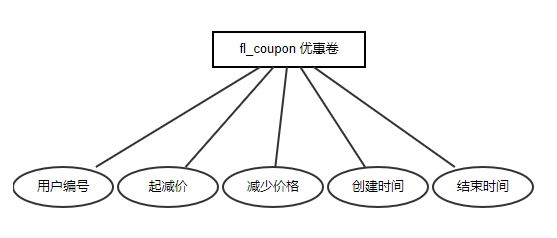
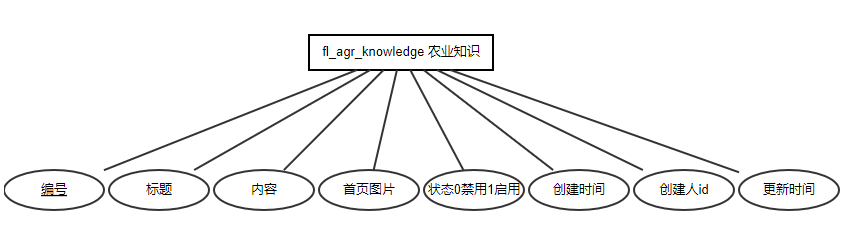
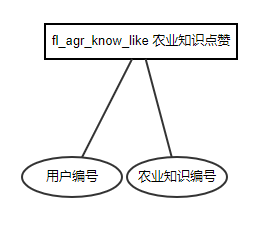
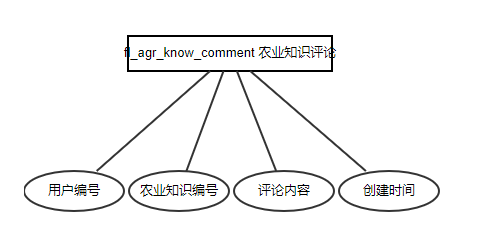
* 在分布式系统中，数据同步与复制是确保数据一致性和可用性的关键。
* 通过定期或实时的数据同步与复制操作，可以确保不同节点上的数据保持一致。

## 4.4示例说明

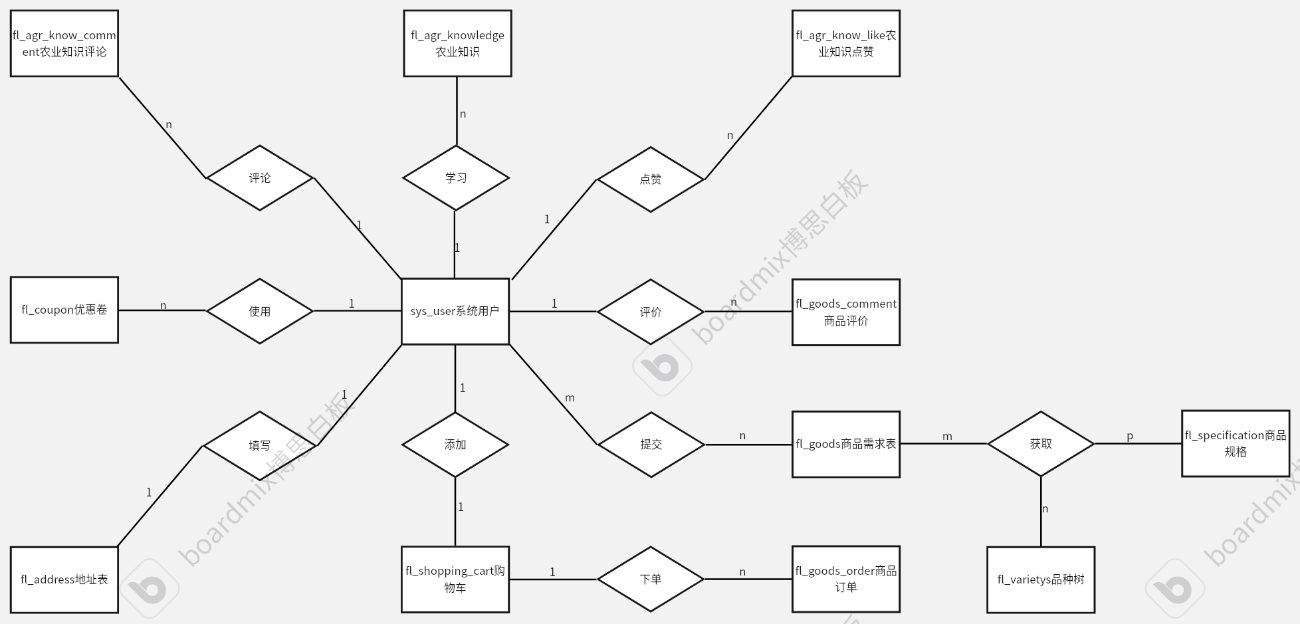
以一个基于数字经济的农产品融销一体平台为例（简称“融销通”）：

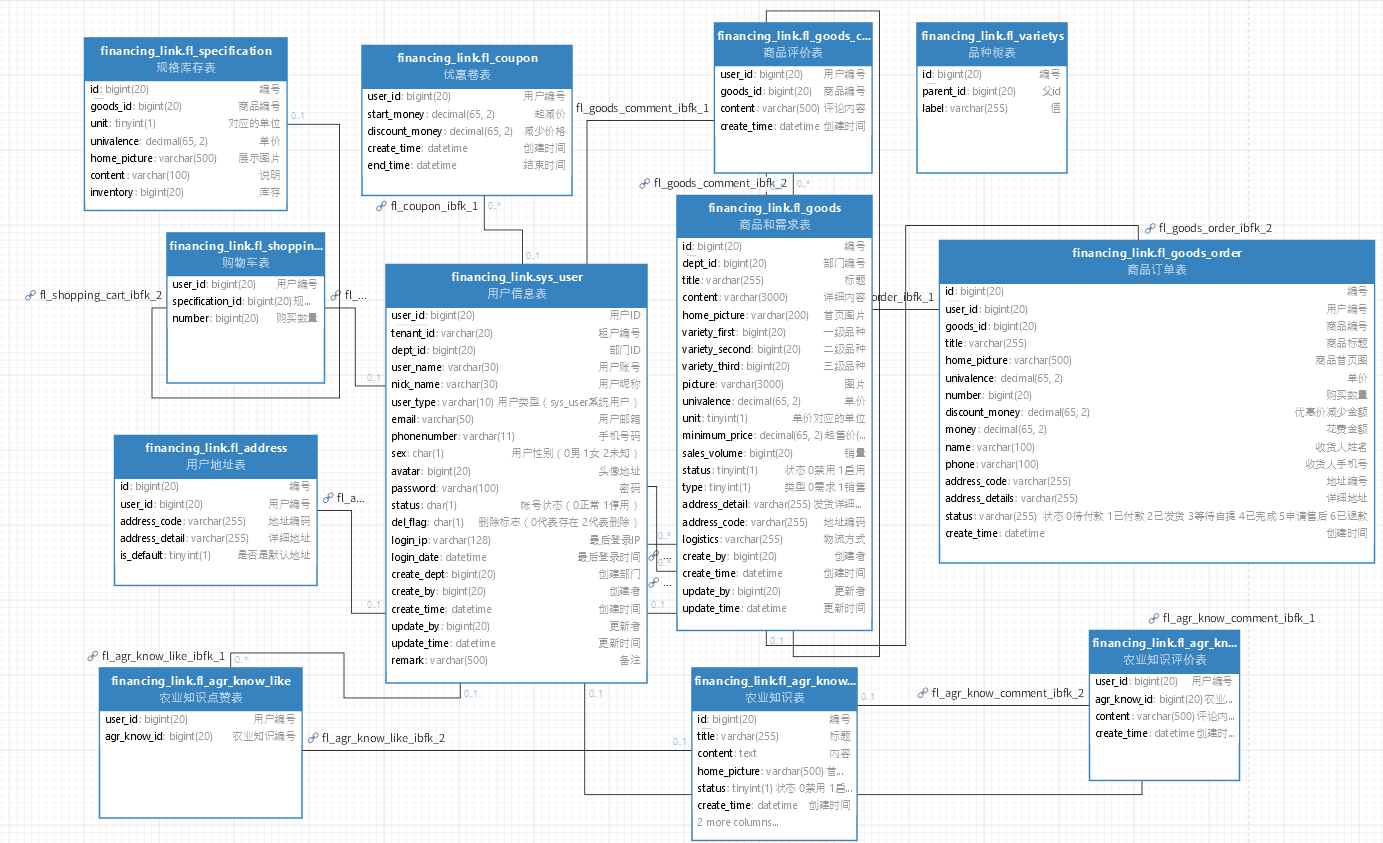
* **用户数据**：用户的个人信息、账户信息等可能被存储在多个节点上，以确保数据的可用性和安全性。
* **产品信息**：农产品的详细信息、价格、库存等也可能被分片存储在不同的节点上，以提高查询和更新速度。
* **交易数据**：用户的交易记录、订单信息等可能被存储在具有高可用性和一致性的分布式数据库中，以确保数据的完整性和可靠性。

# 5.逻辑结构设计

以下为E-R图的实体，其中包含属性，如下所示：  
  


总E-R图：

6.物理结构设计



|  |  |  |
| --- | --- | --- |
| *表名* | *中文名称* | *功能说明* |
| *sys\_user* | *用户信息表* | 所有用户信息 |
| *fl\_goods* | *商品和需求表* | 所有商品和需求 |
| *fl\_specification* | *规格库存表* | 商品对应的规格库存 |
| *fl\_agr\_knowledge* | *农业知识表* | 所有的农业知识 |
| *fl\_agr\_know\_like* | *农业知识点赞表* | 农业知识对应的点赞 |
| *fl\_agr\_know\_comment* | *农业知识评论表* | 农业知识对应的评论 |
| *fl\_goods\_comment* | *商品评价表* | 商品对于的评价 |
| *fl\_shopping\_cart* | *购物车表* | 所有的购物车 |
| *fl\_coupon* | *优惠劵表* | 所有用户的优惠卷 |
| *fl\_goods\_order* | *商品订单表* | 商品对应的订单 |
| *fl\_address* | *用户地址表* | 用户对应的地址 |
| *fl\_varietys* | *种类树表* | 商品和需求对应的品种种类树 |

表1:fl\_address

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | id | bigint | 20 | 是 | 编号 |
| 2 | user\_id | bigint | 20 |  | 用户编号 |
| 3 | address\_code | varchar | 255 |  | 地址编码 |
| 4 | address\_detail | varchar | 255 |  | 详细地址 |
| 5 | is\_default | tinyint | 1 |  | 是否是默认地址 |

表2:fl\_agr\_know\_comment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | user\_id | bigint | 20 |  | 用户编号 |
| 2 | agr\_know\_id | bigint | 20 |  | 农业知识编号 |
| 3 | content | varchar | 500 |  | 评论内容 |
| 4 | create\_time | datetime |  |  | 创建时间 |

表3:fl\_agr\_know\_like

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | user\_id | bigint | 20 |  | 用户编号 |
| 2 | agr\_know\_id | bigint | 20 |  | 农业知识编号 |

表4:fl\_agr\_knowledge

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | id | bigint | 20 | 是 | 编号 |
| 2 | title | varchar | 255 |  | 标题 |
| 3 | content | text |  |  | 内容 |
| 4 | home\_picture | varchar | 500 |  | 首页图片 |
| 5 | status | tinyint | 1 |  | 状态0禁用1启用 |
| 6 | create\_time | datetime |  |  | 创建时间 |
| 7 | create\_by | bigint | 20 |  | 创建人id |
| 8 | update\_time | datetime |  |  | 更新时间 |

表5:fl\_coupon

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | user\_id | bigint | 20 |  | 用户编号 |
| 2 | start\_money | decimal | 65,2 |  | 起减价 |
| 3 | discount\_money | decimal | 65,2 |  | 减少价格 |
| 4 | create\_time | datetime |  |  | 创建时间 |
| 5 | end\_time | datetime |  |  | 结束时间 |

表6:fl\_goods

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | id | bigint | 20 | 是 | 编号 |
| 2 | dept\_id | bigint | 20 |  | 部门编号 |
| 3 | title | varchar | 255 |  | 标题 |
| 4 | content | varchar | 3000 |  | 详细内容 |
| 5 | home\_picture | varchar | 200 |  | 首页图片 |
| 6 | variety\_first | bigint | 20 |  | 一级品种 |
| 7 | variety\_second | bigint | 20 |  | 二级品种 |
| 8 | variety\_third | bigint | 20 |  | 三级品种 |
| 9 | picture | varchar | 3000 |  | 图片 |
| 10 | univalence | decimal | 65,2 |  | 单价 |
| 11 | unit | tinyint | 1 |  | 单价对应的单位 |
| 12 | minimum\_price | decimal | 65,2 |  | 起售价(元) |
| 13 | sales\_volume | bigint | 20 |  | 销量 |
| 14 | status | tinyint | 1 |  | 状态0禁用1启用 |
| 15 | type | tinyint | 1 |  | 类型0需求1销售 |
| 16 | address\_detail | varchar | 255 |  | 发货详细地址 |
| 17 | address\_code | varchar | 255 |  | 地址编码 |
| 18 | logistics | varchar | 255 |  | 物流方式 |
| 19 | create\_by | bigint | 20 |  | 创建者 |
| 20 | create\_time | datetime |  |  | 创建时间 |
| 21 | update\_by | bigint | 20 |  | 更新者 |
| 22 | update\_time | datetime |  |  | 更新时间 |

表7:fl\_goods\_comment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | user\_id | bigint | 20 |  | 用户编号 |
| 2 | goods\_id | bigint | 20 |  | 商品编号 |
| 3 | content | varchar | 500 |  | 评论内容 |
| 4 | create\_time | datetime |  |  | 创建时间 |

表8:fl\_goods\_order

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | id | bigint | 20 | 是 | 编号 |
| 2 | user\_id | bigint | 20 |  | 用户编号 |
| 3 | goods\_id | bigint | 20 |  | 商品编号 |
| 4 | title | varchar | 255 |  | 商品标题 |
| 5 | home\_picture | varchar | 500 |  | 商品首页图 |
| 6 | univalence | decimal | 65,2 |  | 单价 |
| 7 | number | bigint | 20 |  | 购买数量 |
| 8 | discount\_money | decimal | 65,2 |  | 优惠价减少金额 |
| 9 | money | decimal | 65,2 |  | 花费金额 |
| 10 | name | varchar | 100 |  | 收货人姓名 |
| 11 | phone | varchar | 100 |  | 收货人手机号 |
| 12 | address\_code | varchar | 255 |  | 地址编号 |
| 13 | address\_details | varchar | 255 |  | 详细地址 |
| 14 | status | varchar | 255 |  | 状态0待付款1已付款2已发货3等待自提4已完成5申请售后6已退款 |
| 15 | create\_time | datetime |  |  | 创建时间 |

表9:fl\_shopping\_cart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | user\_id | bigint | 20 |  | 用户编号 |
| 2 | specification\_id | bigint | 20 |  | 规格库存编号 |
| 3 | number | bigint | 20 |  | 购买数量 |

表10:fl\_specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | id | bigint | 20 | 是 | 编号 |
| 2 | goods\_id | bigint | 20 |  | 商品编号 |
| 3 | unit | tinyint | 1 |  | 对应的单位 |
| 4 | univalence | decimal | 65,2 |  | 单价 |
| 5 | home\_picture | varchar | 500 |  | 展示图片 |
| 6 | content | varchar | 100 |  | 说明 |
| 7 | inventory | bigint | 20 |  | 库存 |

表11:fl\_varietys

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | id | bigint | 20 | 是 | 编号 |
| 2 | parent\_id | bigint | 20 |  | 父id |
| 3 | label | varchar | 255 |  | 值 |

表12:sys\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 主键 | 说明 |
| 1 | user\_id | bigint | 20 | 是 | 用户ID |
| 2 | tenant\_id | varchar | 20 |  | 租户编号 |
| 3 | dept\_id | bigint | 20 |  | 部门ID |
| 4 | user\_name | varchar | 30 |  | 用户账号 |
| 5 | nick\_name | varchar | 30 |  | 用户昵称 |
| 6 | user\_type | varchar | 10 |  | 用户类型（sys\_user系统用户） |
| 7 | email | varchar | 50 |  | 用户邮箱 |
| 8 | phonenumber | varchar | 11 |  | 手机号码 |
| 9 | sex | char | 1 |  | 用户性别（0男1女2未知） |
| 10 | avatar | bigint | 20 |  | 头像地址 |
| 11 | password | varchar | 100 |  | 密码 |
| 12 | status | char | 1 |  | 帐号状态（0正常1停用） |
| 13 | del\_flag | char | 1 |  | 删除标志（0代表存在2代表删除） |
| 14 | login\_ip | varchar | 128 |  | 最后登录IP |
| 15 | login\_date | datetime |  |  | 最后登录时间 |
| 16 | create\_dept | bigint | 20 |  | 创建部门 |
| 17 | create\_by | bigint | 20 |  | 创建者 |
| 18 | create\_time | datetime |  |  | 创建时间 |
| 19 | update\_by | bigint | 20 |  | 更新者 |
| 20 | update\_time | datetime |  |  | 更新时间 |
| 21 | remark | varchar | 500 |  | 备注 |

# 7.安全性设计

***提示：提高软件系统的安全性应当从“管理”和“设计”两方面着手。****这里仅考虑数据库的安全性设计。*

## 7.1 防止用户直接操作数据库的方法

***提示：****用户只能用帐号登陆到应用软件，通过应用软件访问数据库，而没有其他途径操作数据库。*

## 7.2 用户帐号密码的加密方法

- 哈希算法：采用如`BCrypt`或`Argon2`算法，对用户密码进行单向哈希加密。这些算法支持加盐并具有强大的抗破解性。`BCrypt`是比较流行的密码加密方式，在Java中可以通过`BCryptPasswordEncoder`实现。

- 不存储明文密码：在数据库中只存储加密后的哈希值，不存储密码的明文。

- 传输层加密：使用HTTPS协议加密数据传输，确保用户密码在网络传输过程中不会被拦截。

## 7.3 角色与权限

|  |  |  |
| --- | --- | --- |
| 角色 | 可以访问的表与列 | 操作权限 |
| 管理员 | sys\_user   所有用户信息列 | 检索、更新、删除 |
| fl\_goods          商品信息列 | 创建、检索、更新、删除 |
| fl\_specification  所有规格和库存信息列 | 创建、检索、更新、删除 |
| fl\_agr\_knowledge   所有知识信息列 | 创建、检索、更新、删除 |
| fl\_agr\_know\_like   所有点赞信息列 | 检索、删除 |
| fl\_agr\_know\_comment  所有评论列 | 检索、删除 |
| fl\_goods\_comment   所有商品评价信息列 | 检索、删除 |
| fl\_shopping\_cart   所有购物车信息列 | 检索、删除 |
| fl\_coupon         所有优惠券信息列 | 创建、检索、更新、删除 |
| fl\_goods\_order     所有订单信息列 | 检索、更新、删除 |
| fl\_address         所有用户地址信息列 | 检索、删除 |
| fl\_varietys    种类树信息列 | 创建、检索、更新、删除 |

# 8.优化

|  |  |  |
| --- | --- | --- |
| 优先级 | 优化对象（目标） | 措施 |
| 高 | 查询响应速度 | 添加索引以加速高频查询字段，减少全表扫描；使用缓存技术。 |
| 中 | 数据存储空间 | 针对大表进行分区处理，或者压缩历史数据表。 |
| 低 | 数据冗余和完整性 | 适当反规范化，提高查询效率，减少连接操作带来的消耗。 |

折中方案：

若查询速度与数据空间产生冲突时，如过多索引增加了存储空间，可仅为高频查询字段创建索引；或将频繁查询的冗余字段添加至数据表中，减少连接操作。

具体优化措施

- 优化数据库参数：调整数据库内存分配、连接池等参数，提高整体性能。

- 反规范化：在高频使用的表中，将关联信息存储到同一表中，减少连接操作。

- 数据分片与归档：将历史数据归档至独立存储中，提升主表查询效率。

# 9.数据库管理与维护说明

## 9.1 数据备份

   - 定期执行全量和增量备份，建议每日进行增量备份，每周进行全量备份，并将备份文件存储在独立的安全位置。

   - 对备份数据进行加密，避免数据泄露。

   - 定期测试备份恢复流程，确保备份数据的可用性。

## 9.2 权限管理

   - 基于最小权限原则，按照用户角色分配数据库访问权限。

   - 定期审查用户权限，确保用户仅能访问和操作所需的资源。

   - 对管理员、开发人员、和普通用户设置不同的权限级别，以防止未经授权的数据访问。

## 9.3 日志管理

   - 记录所有数据库操作日志，包括数据增删改、用户登录、异常操作等。

   - 日志文件定期归档，以便于日后追踪问题和进行审计。

   - 设置日志清理策略，将过期的日志文件定期删除或归档到冷存储，释放存储空间。

## 9.4 性能监控

   - 使用数据库监控工具或内置监控（如数据库自带的性能视图）实时跟踪数据库负载、查询响应时间、锁等待等指标。

   - 定期分析慢查询日志，对长时间运行的查询进行优化，或设置索引提升查询效率。

   - 监控并优化连接池参数，避免资源浪费或连接耗尽。

## 9.5数据清理和归档

   - 定期清理无效或过期数据，如历史交易记录、旧的交互信息等，以减少数据量。

   - 将不常用的历史数据归档到单独的表或冷存储中，以提升主表的查询效率。

   - 设计自动清理机制，定期清理临时表和缓存数据，释放空间。

## 9.6 索引维护

   - 定期检查和重建索引，避免由于频繁的增删改操作导致索引碎片化。

   - 优化索引策略，仅为高频查询列设置索引，以平衡查询速度和存储空间。

## 9.7安全更新与漏洞修补

   - 定期更新数据库软件，应用安全补丁，以防止已知的安全漏洞被利用。

   - 定期进行安全扫描和审计，确保数据库环境符合安全合规要求。

## 9.8文档管理

   - 维护详细的数据库架构文档、备份恢复流程文档和权限配置文档，以便于后续运维和开发团队参考。

   - 在数据库变更时及时更新相关文档，保持一致性，为后期编写用户手册奠定基础。