

# 数据库设计说明书

团队：银河护胃队

学院：计算机与大数据学院

专业：数据科学与大数据技术

年级：2022级

2024 年 10 月 31 日

# 目录

## 目录

数据库设计说明书 .....	1
1. 引言 .....	3
1.1 编写目的 .....	3
1.2 范围 .....	3
1.3 项目背景 .....	3
1.4 术语解释 .....	3
1.5 参考资料 .....	3
2. 数据库环境说明 .....	3
2.1 采用的数据库系统 .....	3
3. 结构设计 .....	4
3.1 概念结构设计 .....	4
3.2 逻辑结构设计 .....	5
3.3 物理结构设计 .....	8
4. 数据字典设计 .....	8
4.1 基本表设计 .....	8
5. 对象关系映射 .....	13
6. 安全性设计 .....	16
7. 性能优化 .....	17

## 1. 引言

### 1.1 编写目的

本文档旨在详细描述智能健康饮食推荐系统的数据库设计，包括结构设计、数据字典设计、安全性设计和性能优化，以确保数据库系统能够高效、安全地支持系统功能。

### 1.2 范围

本文档覆盖智能健康饮食推荐系统数据库的各个方面，包括概念结构、逻辑结构、物理结构、数据字典、安全性和性能优化。

### 1.3 项目背景

软件名称：养食记

开发团队：银河护胃队

项目简介：智能健康饮食推荐系统是一个基于人工智能技术的平台，它通过分析用户的身体情况和健康目标，提供个性化的饮食建议和外卖推荐。系统需要处理大量的用户数据、食物营养信息和外卖商品信息。

### 1.4 术语解释

- AI大数据模型：用于分析用户数据和提供个性化推荐的人工智能算法。
- 个性化饮食建议：根据用户的身体情况、要求和健康目标定制的饮食方案
- 菜品推荐：基于用户需求和位置信息推荐的菜品和商家。

### 1.5 参考资料

《数据库设计说明书》国际规范文本

## 2. 数据库环境说明

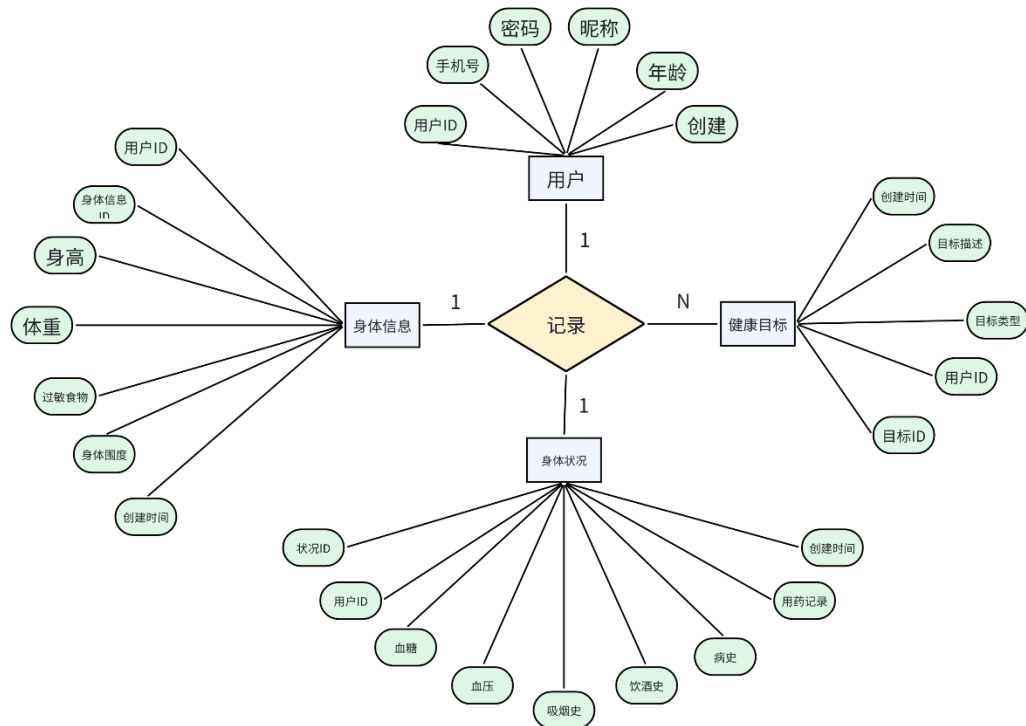
### 2.1 采用的数据库系统

Mysql 8.0

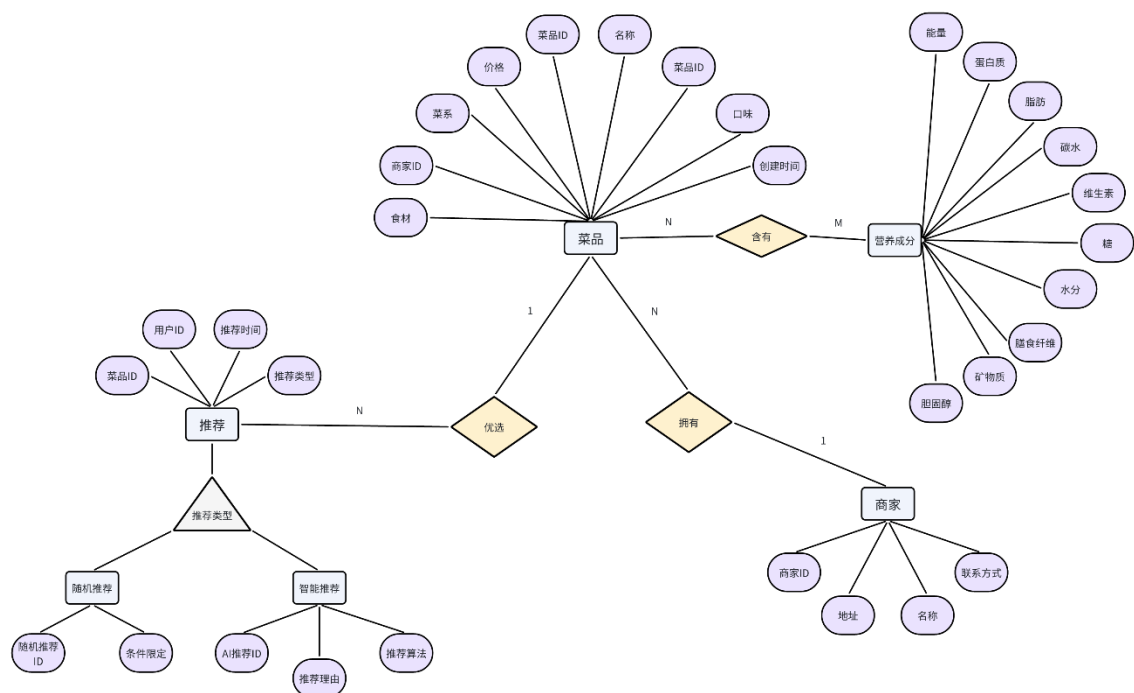
### 3. 结构设计

#### 3.1 概念结构设计

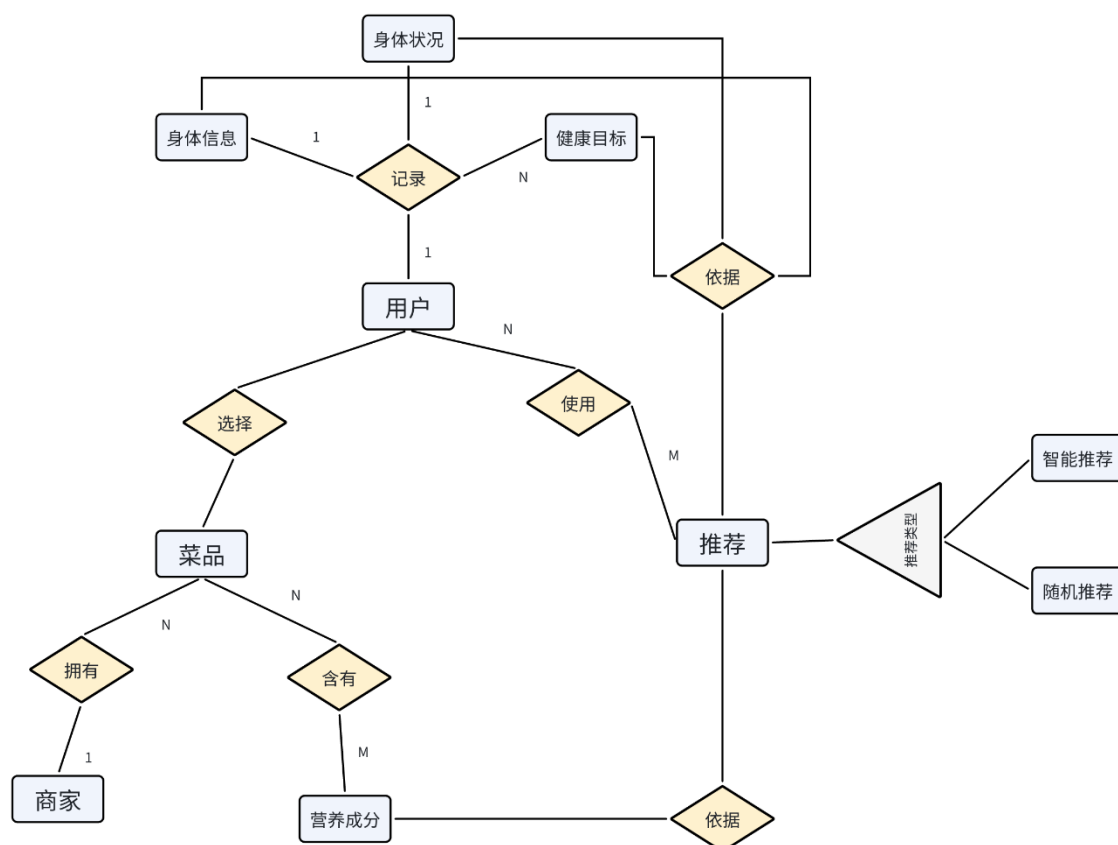
- 用户 E-R 图



- 菜品E-R 图



- 业务服务 E-R 图



## 3.2 逻辑结构设计

### 1. 用户账号信息表

- **主键:** 用户ID
- **关系:**
  - ①与 身体信息表 存在 1 对 1 关系, 一个用户可以有多个身体信息记录;
  - ②与 身体状况 存在 1 对 1 关系, 一个用户可以有多个身体状态;
  - ③与 健康目标表 存在 1 对 n 关系, 一个用户可以有多个健康目标记录。
- **业务说明:** 存储个人账户的基本信息包括账号, 密码, 手机号码等。

### 2. 身体信息表

- **主键:** 身体信息ID
- **关系:**
  - ①与 用户账号信息表 存在 n 对 1 关系: 每个身体信息记录属于一个用户;
- **业务说明:** 记录用户的身体健康信息, 如身高、体重等。

### 3. 健康目标表

- 主键：目标ID

- 关系：

①与 用户账号信息表 存在 n 对 1 关系：每个健康目标记录属于一个用户；

### 4. 健用户饮食习惯表

- 主键：用户饮食习惯ID

- 关系：

①与 用户信息表 存在 n 对 1 关系：多个饮食习惯记录属于一个用户；

### 5. 健康目标表

- 主键：目标ID

- 关系：

①与 用户账号信息表 存在 n 对 1 关系：每个健康目标记录属于一个用户；

- 业务说明：记录用户设定的健康目标；

### 6. 菜品信息表

- 主键：菜品ID

- 关系：

①与 商家信息表 存在 n 对 1 关系，每个商家含有多个菜品；

②与 食材信息表 存在 1 对 n 关系，一个菜品可以含有多种食材；

- 业务说明：存储菜品的详细信息，包括菜品名称，营养成分，包含食材等。

### 7. 商家信息表

- 主键：商家ID

- 关系：

①与 食材信息表 存在 1 对 n 关系，一个商家可以销售多种菜品；

- 业务说明：存储商家的基本信息，包括名称，地址，菜品库等。

### 8. 食材表

- 主键：食材ID

- 关系：

①与 菜品 存在 n 对 1 关系，一个菜品可以包含多种食材；

- 业务说明：记录每个种食材的详细信息，包括食材名称，营养成分，食材类别等。

## 9. 用户打卡表

- **主键：**用户打卡ID
- **关系：**
  - ①与 用户信息表 存在 n 对 1 关系，一个用户可以含有多个打卡记录表（不同月份，年份）；
- **业务说明：**记录用户某一段时间内的打卡信息。

## 10. 用户AI推荐历史表（聊天记录）

- **主键：**用户ID
- **关系：**
  - ①与 用户信息表 存在 n 对 1 关系，一个用户拥有多个推荐历史记录；
- **业务说明：**存储每场比赛的详细信息，包括参赛选手、比分和比赛时间。

## 11. 用户评价表

- **主键：**评价ID
- **关系：**
  - ①与 用户信息表 存在 n 对 1 关系，一个用户可以有多条用户评价；
  - ②与 菜品信息表 存在 n 对 1 关系，一个菜品可以拥有多个评价。
- **业务说明：**记录评价的内容和基本信息，包括评分，评价时间，评价用户，菜品等。

## 12. 每日摄入表

- **主键：**每日摄入ID
- **关系：**
  - ①与 用户信息表 存在 n 对 1 关系，一个用户可以有多条每日摄入记录；
- **业务说明：**记录用户每天的能量，食物，水分摄入信息。

## 13. 身体维度记录表

- **主键：**用户ID
- **关系：**
  - ①与 用户信息表 存在 1对 1 关系，每个用户拥有一个用户维度记录。
- **业务说明：**存储用户的身体维度信息。

### 3.3 物理结构设计

#### 数据访问方式

为提高对会员信息和比赛记录的查询速度，可以为频繁访问的列创建联合索引。这将减少全表扫描的发生，快速定位所需数据，从而加快查询响应，优化查询，提高系统效率。

## 4. 数据字典设计

### 4.1 基本表设计

#### 1. 用户账号信息表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
userId	int unsigned	[v]	[ ]	PRI				用户id
account	varchar(255)	[v]	[ ]					账号
password	varchar(255)	[v]	[ ]					密码
phoneNum	bigint	[ ]	[ ]					手机号码
avatar_url	varchar(255)	[ ]	[ ]					头像url
nickname	varchar(255)	[ ]	[ ]					昵称
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdatedAt	datetime	[ ]	[ ]					更新时间

#### 2. 用户身体信息表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
userId	int unsigned	[v]	[ ]	PRI				用户id
age	int	[ ]	[ ]					年龄
sex	enum('男','女')	[ ]	[ ]					性别
height	float	[ ]	[ ]					身高(m)
weight	float	[ ]	[ ]					体重(kg)
weight_target	float	[ ]	[ ]					目标体重(kg)
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdatedAt	datetime	[ ]	[ ]					更新时间



### 3. 用户健康信息表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
userId	int unsigned	[v]	[ ]	PRI				用户id
blood_sugar	varchar(255)	[ ]	[ ]					血糖 (XX/XX mmol/L)
blood_pressure	varchar(255)	[ ]	[ ]					血压 (XX/XX mmHg)
allergens	json	[ ]	[ ]					过敏源
RecentMedications	text	[ ]	[ ]					近期用药记录
past_medical_history	text	[ ]	[ ]					既往病史
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdatedAt	datetime	[ ]	[ ]					更新时间

### 4. 体重记录表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
weight_record_id	int unsigned	[v]	[ ]	PRI				体重记录id
userId	int unsigned	[v]	[ ]					用户id
new_weight	Float	[ ]	[ ]					更新体重
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdatedAt	datetime	[ ]	[ ]					更新时间

### 5. 用户饮食习惯表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
userId	int unsigned	[v]	[ ]	PRI				用户id
diet_goals	json	[ ]	[ ]					饮食目标
taste_preference	json	[ ]	[ ]					喜好口味
avoid_certain_food	json	[ ]	[ ]					忌口
fitness_planning	json	[ ]	[ ]					健身规划
exercise_habits	json	[ ]	[ ]					运动习惯
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdatedAt	datetime	[ ]	[ ]					更新时间

## 6. 菜品信息表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
dishId	int unsigned	[v]	[ ]	PRI				菜品id
dish_name	varchar(255)	[ ]	[ ]					菜品名称
ingredients	json	[ ]	[ ]					食材
Ingredients	json	[ ]	[ ]					食材及含量
nutrition	json	[ ]	[ ]					营养成分及含量
flavor	varchar(255)	[ ]	[ ]					口味
Cuisine	varchar(255)	[ ]	[ ]					菜系
price	decimal(10,2)	[ ]	[ ]					价格
MerchantId	int unsigned	[ ]	[ ]					所属商家id
dish_image	varchar(255)	[ ]	[ ]					菜品图片url
score	float	[ ]	[ ]					用户评分
allergens	json	[ ]	[ ]					过敏成分
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdateAt	datetime	[ ]	[ ]					更新时间

## 7. 商家信息表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
MerchantId	int unsigned	[v]	[ ]	PRI				商家id
Merchant_name	varchar(255)	[ ]	[ ]					商家名称
address	varchar(255)	[ ]	[ ]					地址
menu	json	[ ]	[ ]					菜单
inco	varchar(255)	[ ]	[ ]					商家图标
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdateAt	datetime	[ ]	[ ]					更新时间

## 8. 用户打卡表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
Log_Id	int unsigned	[v]	[ ]	PRI				打卡记录id
userId	int unsigned	[v]	[ ]					用户id
year	date	[ ]	[ ]					打卡年份
mouth	date	[ ]	[ ]					打卡月份
log_days	json	[ ]	[ ]					打卡日期
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdateAt	datetime	[ ]	[ ]					更新时间

## 9. 用户AI推荐历史（聊天记录）

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
userId	int unsigned	[v]	[ ]	PRI				用户id
contain	text	[ ]	[ ]					聊天内容
CreatedAt	datetime	[ ]	[ ]					创建时间
Initiating_object	enum('AI','user')	[ ]	[ ]					发起对象

## 10. 用户评价表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
appraiseId	int unsigned	[v]	[ ]	PRI				评价id
userId	int unsigned	[v]	[ ]					用户id
dishId	int unsigned	[v]	[ ]					菜品id
score	float	[ ]	[ ]					评分
appraise	text	[ ]	[ ]					评价
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdateAt	datetime	[ ]	[ ]					更新时间

## 11. 食材表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
Ingredientid	int unsigned	[v]	[ ]	PRI				食材id
Ingredient_name	varchar(255)	[ ]	[ ]					食材名称
efficacy	json	[ ]	[ ]					功效
Contraindications	json	[ ]	[ ]					禁忌
types	varchar(255)	[ ]	[ ]					食材类型
Nutritions	json	[ ]	[ ]					营养成分
unit_heat	int	[ ]	[ ]					单位热量
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdateAt	datetime	[ ]	[ ]					更新时间

## 12. 每日摄入表

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
id	int unsigned	[v]	[ ]	PRI				摄入表id
userId	int unsigned	[v]	[ ]					用户id
date	datetime	[ ]	[ ]					日期
water_target	int	[ ]	[ ]					饮水目标(ml)
have_drunk	int	[ ]	[ ]					已喝水量(ml)
power_target	int	[ ]	[ ]					能量摄入目标(kcal)
have_eaten	int	[ ]	[ ]					已摄入能量(kcal)
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdateAt	datetime	[ ]	[ ]					更新时间

## 13. 身体维度记录

列名	#数据类型	非空	自增	键	默认	额外的	表达式	注释
id	int unsigned	[v]	[ ]	PRI				围度记录id
userId	int unsigned	[v]	[ ]					用户id
waist	float	[ ]	[ ]					腰围(cm)
thigh	float	[ ]	[ ]					大腿围(cm)
calf	float	[ ]	[ ]					小腿围(cm)
bust	float	[ ]	[ ]					胸围(cm)
hips	float	[ ]	[ ]					臀围(cm)
arm	float	[ ]	[ ]					臂围(cm)
CreatedAt	datetime	[ ]	[ ]					创建时间
UpdateAt	datetime	[ ]	[ ]					更新时间

## 5. 对象关系映射

### 1. 用户账号信息表

```
public class UserAccount {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long userId; // 用户id  
    private String account; // 账号  
    private String password; // 密码  
    private Long phoneNum; // 手机号码  
    private String avatarUrl; // 头像url  
    private String nickname; // 昵称  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

### 2. 用户身体信息表

```
public class UserHealthInfo {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long userId; // 用户id  
    private Integer age; // 年龄  
    private String sex; // 性别  
    private Float height; // 身高(m)  
    private Float weight; // 体重(kg)  
    private Float weightTarget; // 目标体重(kg)  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

### 3. 用户健康信息表

```
public class UserHealthInfo {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long userId; // 用户id  
    private String bloodSugar; // 血糖 (XX/XX mmol/L)  
    private String bloodPressure; // 血压 (XX/XX mmHg)  
    private String allergens; // 过敏源  
    private String recentMedications; // 近期用药记录  
    private String pastMedicalHistory; // 既往病史  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

### 4. 体重记录表

```
public class WeightRecord {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long weightRecordId; // 体重记录id  
    private Long userId; // 用户id  
    private Float newWeight; // 更新体重  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 5. 用户饮食习惯表

```
public class UserDietHabit {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long userId; // 用户id  
    private String dietGoals; // 饮食目标  
    private String tastePreference; // 喜好口味  
    private String avoidCertainFood; // 忌口  
    private String fitnessPlanning; // 健身规划  
    private String exerciseHabits; // 运动习惯  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 6. 菜品信息表

```
public class DishInfo {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long dishId; // 菜品id  
    private String dishName; // 菜品名称  
    private String ingredients; // 食材  
    private String ingredientsDetails; // 食材及含量  
    private String nutrition; // 营养成分及含量  
    private String flavor; // 口味  
    private String cuisine; // 菜系  
    private BigDecimal price; // 价格  
    private Long merchantId; // 所属商家id  
    private String dishImage; // 菜品图片url  
    private Float score; // 用户评分  
    private String allergens; // 过敏成分  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 7. 商家信息表

```
public class MerchantInfo {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long merchantId; // 商家id  
    private String merchantName; // 商家名称  
    private String address; // 地址  
    private String menu; // 菜单  
    private String icon; // 商家图标  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 8. 用户打卡表

```
public class UserCheckIn {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long logId; // 打卡记录id  
    private Long userId; // 用户id  
    private LocalDate year; // 打卡年份  
    private LocalDate month; // 打卡月份  
    private String logDays; // 打卡日期  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间 }  
}
```

## 9. 用户AI推荐历史表

9. 用户AI推荐历史表 (UserAIHistory)

```
public class UserAIHistory {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long userId; // 用户id  
    private String contain; // 聊天内容  
    private LocalDateTime createdAt; // 创建时间  
    private String initiatingObject; // 发起对象  
}
```

## 10. 用户评价表

```
public class UserReview {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long appraiseId; // 评价id  
    private Long userId; // 用户id  
    private Long dishId; // 菜品id  
    private Float score; // 评分  
    private String appraise; // 评价  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 11. 食材表

```
public class IngredientInfo {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long ingredientId; // 食材id  
    private String ingredientName; // 食材名称  
    private String efficacy; // 功效  
    private String contraindications; // 禁忌  
    private String types; // 食材类型  
    private String nutritions; // 营养成分  
    private Integer unitHeat; // 单位热量  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 12. 每日摄入表

```
public class DailyIntake {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id; // 摄入表id  
    private Long userId; // 用户id  
    private LocalDateTime date; // 日期  
    private Integer waterTarget; // 饮水目标(ml)  
    private Integer haveDrunk; // 已喝水量(ml)  
    private Integer powerTarget; // 能量摄入目标(kcal)  
    private Integer haveEaten; // 已摄入能量(kcal)  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 13. 身体维度记录表

```
public class BodyMeasurement {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id; // 围度记录id  
    private Long userId; // 用户id  
    private Float waist; // 腰围(cm)  
    private Float thigh; // 大腿围(cm)  
    private Float calf; // 小腿围(cm)  
    private Float bust; // 胸围(cm)  
    private Float hips; // 臀围(cm)  
    private Float arm; // 臂围(cm)  
    private LocalDateTime createdAt; // 创建时间  
    private LocalDateTime updatedAt; // 更新时间  
}
```

## 6. 安全性设计

### 1. 数据库访问控制

**角色基础的权限管理：**实施基于角色的访问控制（RBAC），确保不同用户仅能访问其所需的数据。

**定期权限审查：**定期审查和更新用户权限，以防止权限滥用。

### 2. 数据加密

**敏感数据加密：**对敏感数据（如用户个人信息、健康数据）进行加密存储，使用 AES 等加密算法。

**数据传输加密：**在数据传输过程中采用 SSL/TLS 加密，确保数据在网络传输中的安全性。

### 3. 数据备份与恢复

**定期备份策略：**定期进行全量和增量备份，确保数据的持久性和可恢复性。

**数据恢复计划：**制定明确的数据恢复计划，定期进行恢复演练，确保在数据丢失或损坏时可以快速恢复。

### 4. 审计与监控

**数据库审计实施：**实施数据库审计，记录所有数据访问和操作日志，便于追踪和分析潜在的安全事件。

**安全事件监控：**设置监控系统，实时监控数据库性能和安全事件，及时发现异常活动。

### 5. SQL 注入防护

**参数化查询使用：**使用参数化查询和预编译语句，避免直接拼接 SQL 语句。



**安全测试周期：**定期进行安全测试，发现和修复 SQL 注入漏洞。

## 6. 数据完整性保护

**数据完整性约束：**使用约束（如主键、外键、唯一性约束）确保数据的完整性。

**事务管理实施：**实施事务管理，确保数据操作的原子性和一致性。

## 7. 数据库防火墙和网络隔离

**数据库防火墙部署：**部署数据库防火墙，阻止可疑的访问请求和攻击行为。

**网络隔离策略：**将数据库服务器与外部网络隔离，限制直接访问，增强安全性。

# 7. 性能优化

## 1. 索引优化

**索引创建与维护：**根据查询频率和数据访问模式，创建适当的索引，提高查询性能。

**索引分析与重建：**定期分析和重建索引，以优化数据库的读取效率。

## 2. 查询优化

**执行计划分析：**通过分析执行计划，识别并优化性能瓶颈的查询语句。

**查询简化：**避免使用不必要的复杂查询，简化 SQL 语句，减少计算负担。

## 3. 数据库配置

**配置参数调整：**根据硬件环境和应用需求，调整数据库配置参数（如缓冲区大小、连接数等）以提升性能。

**性能监测与评估：**定期监测和评估数据库性能，调整配置以应对不断变化的负载。

## 4. 缓存策略

**数据缓存机制：**实施数据缓存机制，将频繁访问的数据存储在内存中，减少数据库的直接查询。

**缓存技术应用：**使用内容分发网络（CDN）和应用缓存（如 Redis）提升数据访问速度和系统响应能力。