**{中南饿了么}数据库设计说明书**

|  |  |  |  |
| --- | --- | --- | --- |
| 状态： | 草稿 | 标识号： | ICSS(HN)-Template-DataBase |
|  | 评审 | 当前版本： | **1.0** |
|  | 初始版 | 前一版本： |  |
|  | 修订版 | 发布日期： | **2024-12-28** |

**修改历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **作者** | **修改内容** | **更请请求号** |
| 2006-5-11 | 1.0 | 高海云 | 初始版 |  |

**目录**

HYPERLINK \l "\_Toc117147443"1 概述 5

HYPERLINK \l "\_Toc117147444"1.1 目的 5

HYPERLINK \l "\_Toc117147445"1.2 范围 5

HYPERLINK \l "\_Toc117147446"1.3 参考文档 5

HYPERLINK \l "\_Toc117147447"1.4 术语与缩写解释 5

HYPERLINK \l "\_Toc117147448"2 设计约束 5

HYPERLINK \l "\_Toc117147449"3 系统总体结构 5

HYPERLINK \l "\_Toc117147450"3.1 子系统N的结构与功能 5

HYPERLINK \l "\_Toc117147451"4 开发环境的配置 6

HYPERLINK \l "\_Toc117147452"5 运行环境的配置 6

HYPERLINK \l "\_Toc117147453"6 测试环境的配置 6

HYPERLINK \l "\_Toc117147454"7 其它 6

# 概述

## 目的

提示：本节描述编制本数据库设计说明书的目的。说明本系统“是什么”，描述本系统的主要功能。

## 范围

提示：本节描述该数据库设计说明书的应用范围等。

# 数据库环境说明

***提示：***

*（1）说明所采用的数据库系统，设计工具，编程工具等*

*（2）详细配置*

Redis:

版本：6.0.16

设计工具：redismanager

配置：redis-server：localhost:6379；

Mysql：

版本：8.0.40-0ubuntu0.22.04.1 for Linux on x86\_64 ((Ubuntu))

设计工具：navicat, tabby

配置：mysqld：localhost:3306；

# 数据库的命名规则

***提示：***

1. *完整并且清楚的说明本数据库的命名规则。*

全小写加下划线

# 逻辑设计

***提示：****数据库设计人员根据需求文档，创建与数据库相关的那部分实体关系图（ERD）。如果采用面向对象方法（OOAD），这里实体相当于类（class）。*

## 实体关系图

User(用户) -----> UserID(用户ID）

## 实体说明

### User实体

实体描述：系统的注册用户信息

属性：

id: 自增主键

uid: 用户唯一标识

name: 用户名

email: 邮箱

passwd: 密码

主键：(id, uid)

业务规则：

用户名和邮箱不可重复

邮箱需要验证

密码需要加密存储

### UserID实体

实体描述：用户ID生成器

属性：

id: 自增主键

主键：id

业务规则：

生成用户的uid

## 数据表设计

### 4.3.1 user表

### -----------------------------------------------

|字段名 |类型 |长度|是否必填 |说明 |

|id |int |11 |是 |自增主键 |

|uid |int |11 |是 |用户唯一标识|

|name |varchar |255 |是 |用户名 |

|email |varchar |255 |是 |邮箱 |

|passwd |varchar |255 |是 |密码 |

### -----------------------------------------------

### 4.3.2 user\_id表

### -------------------------------------

|字段名|类型 |长度|是否必填|说明 |

|id |int |11 |是 |自增主键 |

### -------------------------------------

## 缓存设计

Redis缓存：

Key格式：CODE\_PREFIX + email

Value：验证码

过期时间：短期存储(用于邮箱验证码)

使用场景：

重置密码时的邮箱验证

用户注册时的邮箱验证

## SQL语句

## 用户注册存储过程:

## CALL reg\_user(name, email, passwd, @result)

## 验证用户密码:

SELECT \* FROM user WHERE email = ? AND passwd = ?

更新用户密码:

UPDATE user SET passwd = ? WHERE name = ?

# 物理设计

***提示：***

*（1）主要是设计表结构。一般地，实体对应于表，实体的属性对应于表的列，实体之间的关系成为表的约束。逻辑设计中的实体大部分可以转换成物理设计中的表，但是它们并不一定是一一对应的。*

*（2）对表结构进行规范化处理（第三范式）。*

## 表汇总

|  |  |
| --- | --- |
| 表名 | 功能说明 |
| user | 存储用户基本信息，包含认证信息 |
| user\_id | 用于生成用户唯一标识的自增序列表 |

## 表A

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | user | | |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 |
| id | int(11) | 非空 | 主键，自增 |
| uid | int(11) | 非空 | 主键，外键关联user\_id表 |
| name | varchar(225) | 非空 | 唯一索引 |
| email | varchar(225) | 非空 | 唯一索引 |
| passwd | varchar(225) | 非空 | 存储加密后的密码 |
| 补充说明 | 1. id和uid构成联合主键，确保数据唯一性  2. name和email设置唯一索引，防止重复注册  3. passwd存储加密后的密码字符串，不存储明文  4. 所有字段均设置为非空，确保数据完整性 | | |

## 表B

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | user\_id | | |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 |
| id | int(11) | 非空 | 主键，自增 |
| 补充说明 | 作为id生成器，在一定程度上防止sql注入 | | |

# 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。*

## 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，而没有其它途径操作数据库。*

*1. 网络访问控制*

*MySQL服务器仅监听本地地址(127.0.0.1)*

*数据库端口(3306)只允许应用服务器访问*

*Redis服务配置密码访问：password = whx051021*

*2. 分层访问设计*

*客户端无法直接访问数据库，必须通过以下路径：*

*客户端 → GateServer(50001) → StatusServer/PurchaseServer → 数据库*

*所有数据库操作通过MysqlDao和MysqlManager封装：*

*3.安全传输*

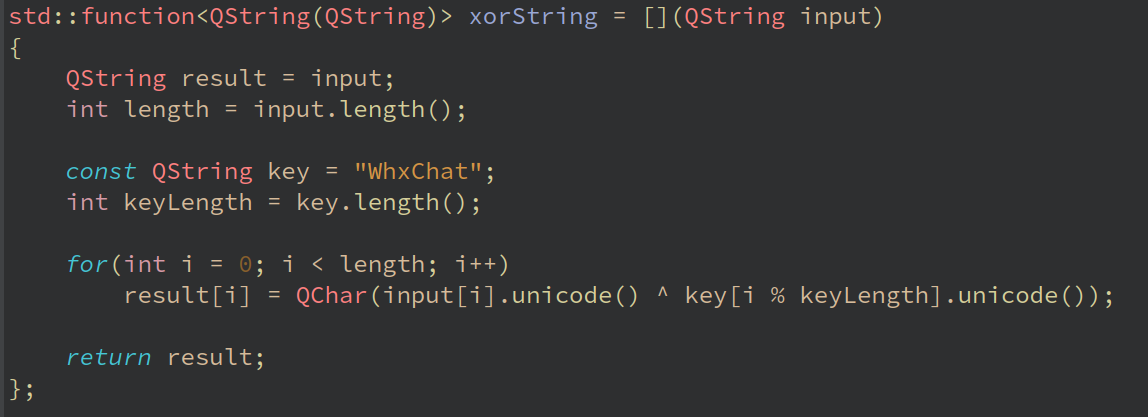
*密码加密存储*

*使用验证码和Token机制*

*服务间通过gRPC加密通信*

## 用户帐号密码的加密方法

***提示：****对用户帐号的密码进行加密处理，确保在任何地方都不会出现密码的明文。*



# 优化

***提示：****分析并优化数据库的“时－空”效率，尽可能地“提高处理速度”并且“降低数据占用空间”。*

*（1）分析“时－空”效率的瓶颈，找出优化对象（目标），并确定优先级。*

*（2）当优化对象（目标）之间存在对抗时，给出折衷方案。*

*（3）给出优化的具体措施，例如优化数据库环境参数，对表格进行反规范化处理等。*

|  |  |  |
| --- | --- | --- |
| 优先级 | 优化对象（目标） | 措施 |
|  |  |  |
|  |  |  |
|  |  |  |

# 数据库管理与维护说明

***提示：****在设计数据库的时候，及时给出管理与维护本数据库的方法，有助于将来撰写出正确完备的用户手册。*