**《风》**

**数据库设计说明书**

**所属学院：至诚学院**

**团队名称：风雨五组**

**指导教师：傅明建**

**项目时间：2021-2022学年**

**目 录**

[版本信息 - 2 -](#_Toc11937)

[第一章 引言 - 3 -](#_Toc14320)

[1.1编写目的 - 3 -](#_Toc22205)

[1.2背景 - 3 -](#_Toc2137)

[1.3参考资料 - 3 -](#_Toc20206)

[第二章 外部设计 - 4 -](#_Toc6309)

[2.1标识符和状态 - 4 -](#_Toc5131)

[2.2命名约定 - 4 -](#_Toc17773)

[2.2.1表命名约定 - 4 -](#_Toc32508)

[2.2.2字段命名规范 - 4 -](#_Toc32733)

[2.3设计约定 - 4 -](#_Toc28800)

[第三章 结构设计 - 5 -](#_Toc4342)

[3.1 概念结构设计 - 5 -](#_Toc13285)

[3.1.1实体和属性的定义 - 5 -](#_Toc27609)

[3.1.2局部E-R图 - 7 -](#_Toc6313)

[3.1.3全局E-R图 - 8 -](#_Toc12130)

[3.2 逻辑结构设计 - 9 -](#_Toc557)

[（1）用户关系模式 - 9 -](#_Toc14746)

[（2）管理员关系模式 - 9 -](#_Toc26070)

[（3）订单关系模式 - 9 -](#_Toc975)

[（4）反馈关系模式 - 9 -](#_Toc13131)

[（5）帮助关系模式 - 9 -](#_Toc29777)

[3.3 物理结构设计 - 9 -](#_Toc16320)

[第四章 运用设计 - 10 -](#_Toc31520)

[4.1安全保密设计 - 10 -](#_Toc21605)

[4.2数据库设计 - 11 -](#_Toc14615)

[4.2.1创建表 - 11 -](#_Toc16751)

[第五章 数据库验收标准 - 12 -](#_Toc11600)

[5.1数据库验收步骤 - 12 -](#_Toc8796)

**版本信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 修订人 | 修改日期 | 备注 |
| 第1版 | 风雨五组 | 2022/4/20 | 初版 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **引言**

**1.1编写目的**

本数据库设计说明书是关于跑腿代拿APP的数据库设计，主要包括系统需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库设计、安全保密设计等。

本数据库设计说明书的预期读者为：用户，开发人员，测试人员

本数据库设计说明书是根据系统需求分析设计所编写的。

**1.2背景**

随着电商以及网络的快速发展，快递已经与我们的日常生活密不可分，网络信息时代，人们的日常生活跟网络息息相关，尤其是当代大学生，很早就开始用手机，习惯于网购的生活，取快递似乎已成为我们日常生活中难以割舍的一部分。

大学生活中，我们往往会因为各种原因或者碰到各种各样的突发状况而没空去领取我们的快递，比如天气不好，驿站太远等等。身为大学生的我们在日常生活中也经常有这种不想去拿快递的烦恼。不仅仅是快递，日常学习生活任务繁忙，但是总不可避免的可能需要去什么地方取一下东西，从用户需求上来看，跑腿代拿平台十分具有必要性和可行性。

**1.3参考资料**

[1][《校易数据库设计说明书》](https://github.com/carry-code-succeed/Campus_Transaction/blob/master/doc/%E6%A0%A1%E6%98%93%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1%E8%AF%B4%E6%98%8E%E4%B9%A6/%E6%A0%A1%E6%98%93%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1%E8%AF%B4%E6%98%8E%E4%B9%A6.d)

[2][《风-需求设计说明书》](https://github.com/2021FYWZ/wind/blob/master/%E9%9C%80%E6%B1%82%E8%A7%84%E6%A0%BC%E8%AF%B4%E6%98%8E%E4%B9%A6.md)

[3][《数据库表结构设计的几条标准》](https://www.cnblogs.com/wyq178/p/8549715.html)

1. **外部设计**

**2.1标识符和状态**

数据库软件的名称：SQL Server

数据库的名称：WIND\_SQL

**2.2命名约定**

**2.2.1表命名约定**

（1）字母全部大写原则  
所有数据库对象命名字母全部大写。  
（2）字符范围原则  
只能使用英文字母、下划线、数字进行命名，首位字符必须是英文字母。  
（3）分段命名原则  
命名中多个单词间采用下划线分隔，以便阅读同时方便某些工具对数据库对象的映射。如XXX\_XXX\_XXX，但不限于三段式。

（4）简单命名原则  
命名尽可能简单，避免太长的命名，尽量使用缩写形式，但是缩写也要能够表达命名的含义。

**2.2.2字段命名规范**

（1）必须有一主键，主键不直接用ID，而是表名+ID

（2）常用的字段name，不直接用name，而是表名+Name

（3）避免中文拼音

（4）避免名称过长

（5）避免保留字

**2.3设计约定**

（1）表必须要有主键。

（2）一个字段只表示一个含义。

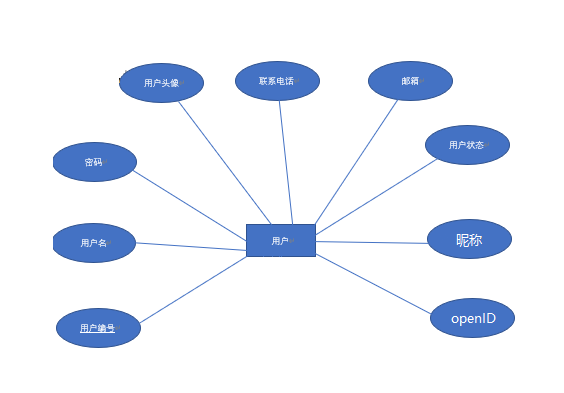
（3）禁止使用复杂数据类型（数组，自定义类型等）。

1. **结构设计**

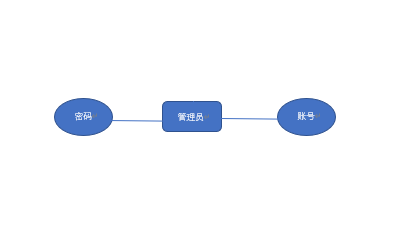
**3.1 概念结构设计**

**3.1.1实体和属性的定义**

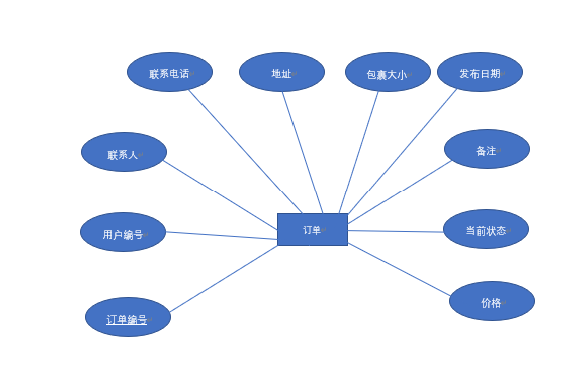
**（1）用户关系模块**



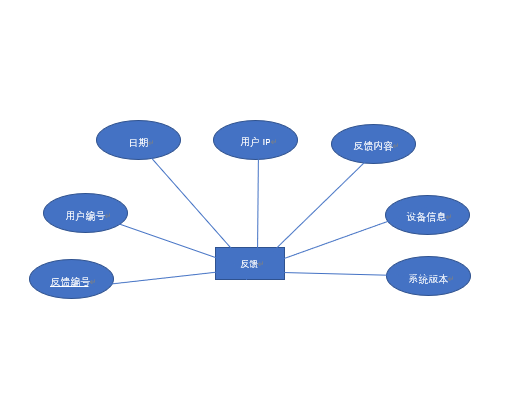
**（2）管理员关系模块**



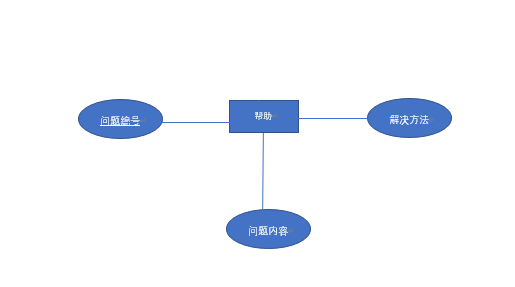
**（3）订单关系模块**



**（4）反馈关系模块**

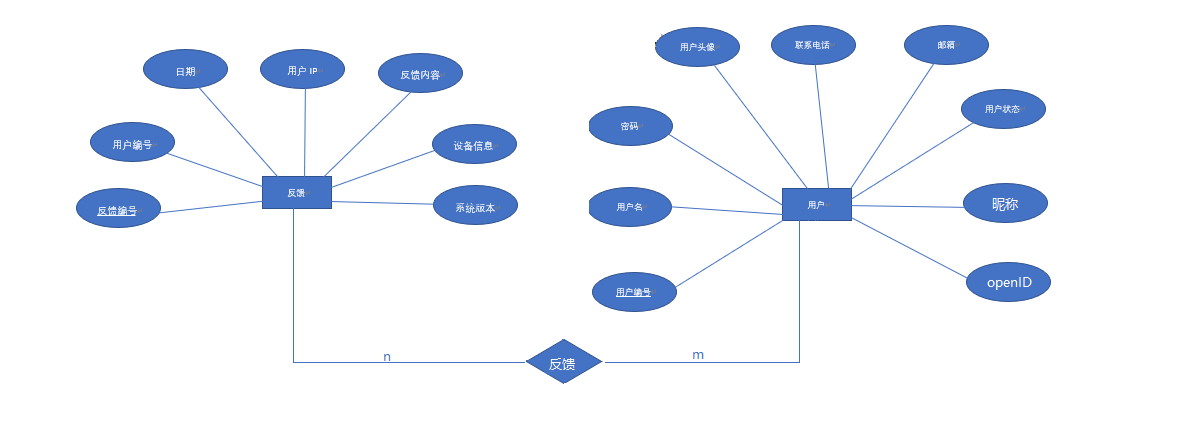


**（5）帮助关系模块**

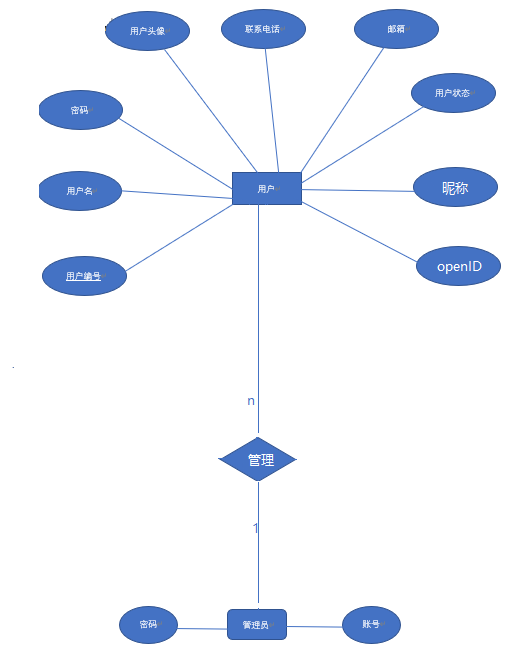


**3.1.2局部E-R图**

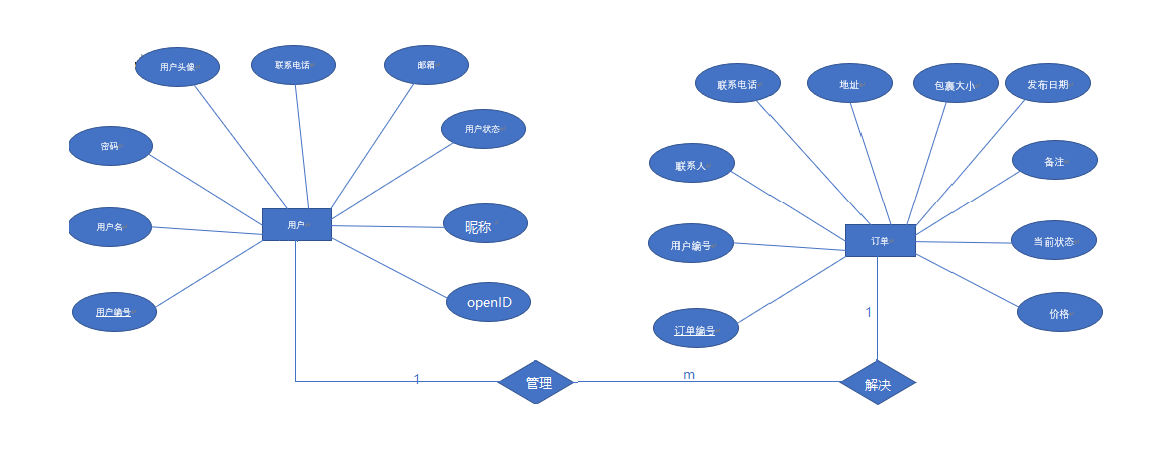
**（1）用户与反馈关系**



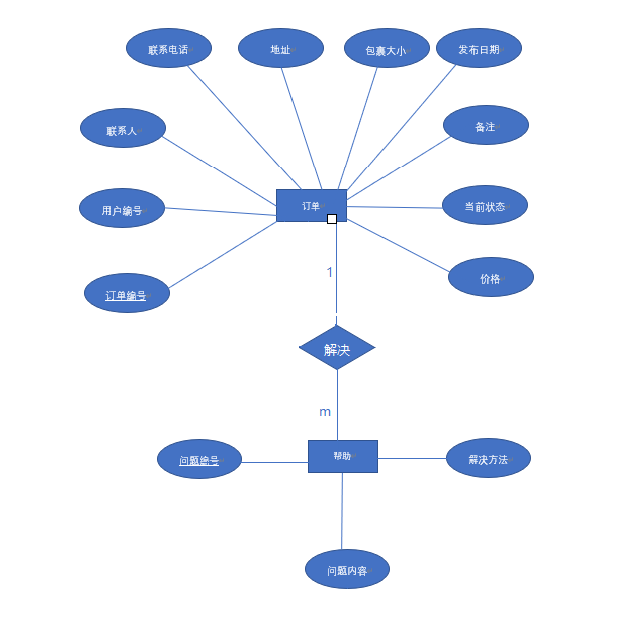
**（2）用户与管理员关系**



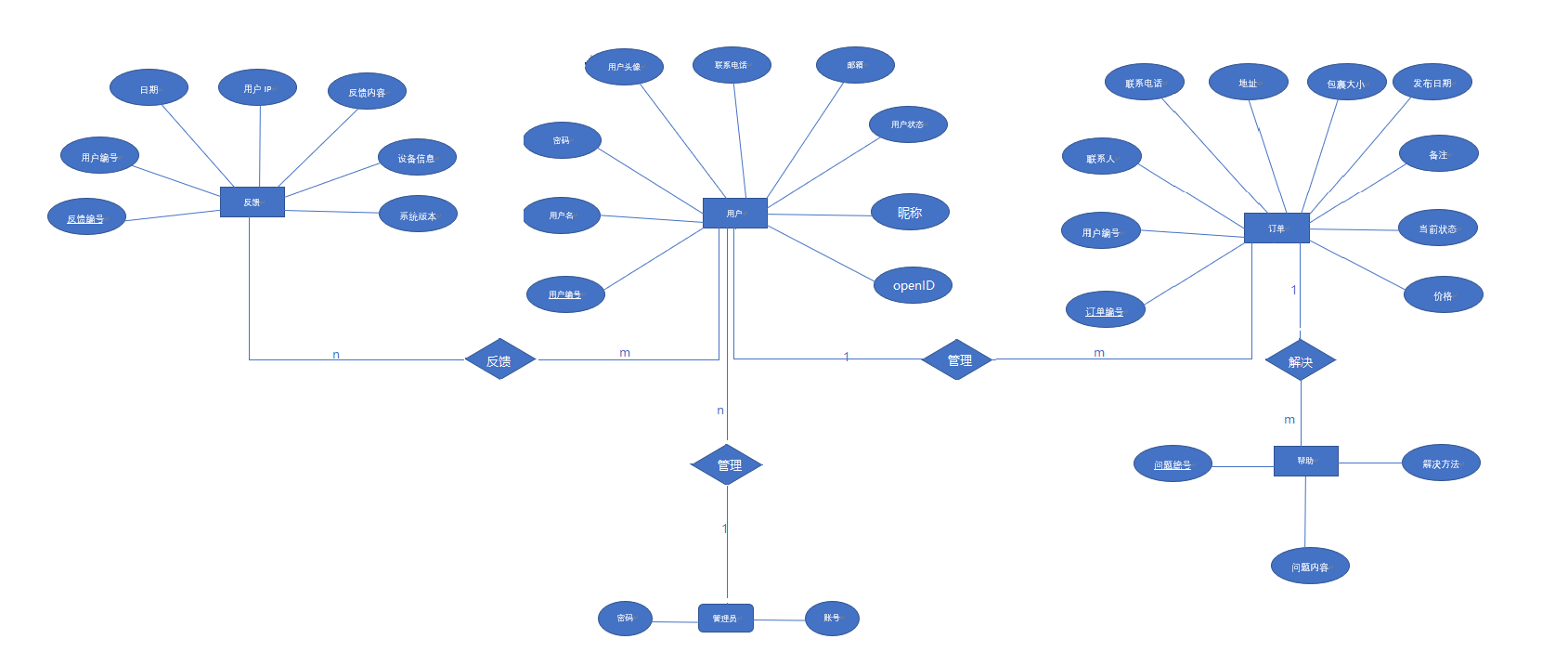
**（3）用户与订单关系**



**（4）订单与帮助关系**



**3.1.3全局E-R图**



**3.2 逻辑结构设计**

**（1）用户关系模式**

用户关系模式（用户编号，用户名，昵称，密码，用户头像，联系电话，邮箱，用户状态，openID）；

**（2）管理员关系模式**

管理员关系模式（管理员账号，密码）；

**（3）订单关系模式**

订单关系模式（订单编号，用户编号，联系人，联系电话，地址，价格，包裹大小，发布日期，备注，当前状态）；

**（4）反馈关系模式**

反馈关系模式（反馈编号，用户编号，日期，用户IP，反馈内容，设备信息，系统版本）；

**（5）帮助关系模式**

帮助关系模式（问题编号，问题内容，解决方法）；

**3.3 物理结构设计**

表1 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段标识 | 字段名 | 数据类型 | 说明 |
| 用户编号 | u\_id | bigint | PK,NOT NULL |
| 用户名 | u\_name | varchar(64) | NOT NULL |
| 密码 | u\_pwd | varchar(64) | NOT NULL |
| 昵称 | u\_nickname | varchar(64) |  |
| 用户头像 | u\_img | varchar(50) |  |
| 联系电话 | u\_phone | varchar(20) |  |
| 邮箱 | u\_email | varchar(64) |  |
| 用户状态 | u\_state | varchar(64) |  |
| OpenId | u\_openid | varchar(64) |  |

表2 管理员表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段标识 | 字段名 | 数据类型 | 说明 |
| 管理员账号 | m\_id | bigint | PK,NOT NULL |
| 密码 | m\_name | varchar(64) | NOT NULL |

表3订单表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段标识 | 字段名 | 数据类型 | 说明 |
| 订单编号 | o\_id | bigint | PK,NOT NULL |
| 用户编号 | u\_id | bigint | FK,NOT NULL |
| 联系人 | o\_contacts | varchar(64) | NOT NULL |
| 联系电话 | o\_phone | varchar(20) | NOT NULL |
| 地址 | o\_address | varchar(64) | NOT NULL |
| 价格 | o\_price | numeric(4,2) | NOT NULL |
| 包裹大小 | o\_package | varchar(64) | NOT NULL |
| 发布日期 | o\_date | date | NOT NULL |
| 当前状态 | o\_state | varchar(64) | NOT NULL |
| 备注 | o\_remark | varchar(64) |  |

表4 反馈表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段标识 | 字段名 | 数据类型 | 说明 |
| 反馈编号 | f\_id | bigint | PK,NOT NULL |
| 用户编号 | u\_id | bigint | FK,NOT NULL |
| 日期 | f\_date | date | NOT NULL |
| 用户IP | f\_ip | varchar(50) | NOT NULL |
| 反馈内容 | f\_content | varchar(64) | NOT NULL |
| 设备信息 | f\_info | varchar(64) | NOT NULL |
| 系统版本 | f\_version | varchar(64) | NOT NULL |

表5 帮助表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段标识 | 字段名 | 数据类型 | 说明 |
| 问题编号 | h\_id | bigint | PK,NOT NULL |
| 问题内容 | h\_content | varchar(64) | NOT NULL |
| 解决方法 | h\_method | varchar(64) | NOT NULL |

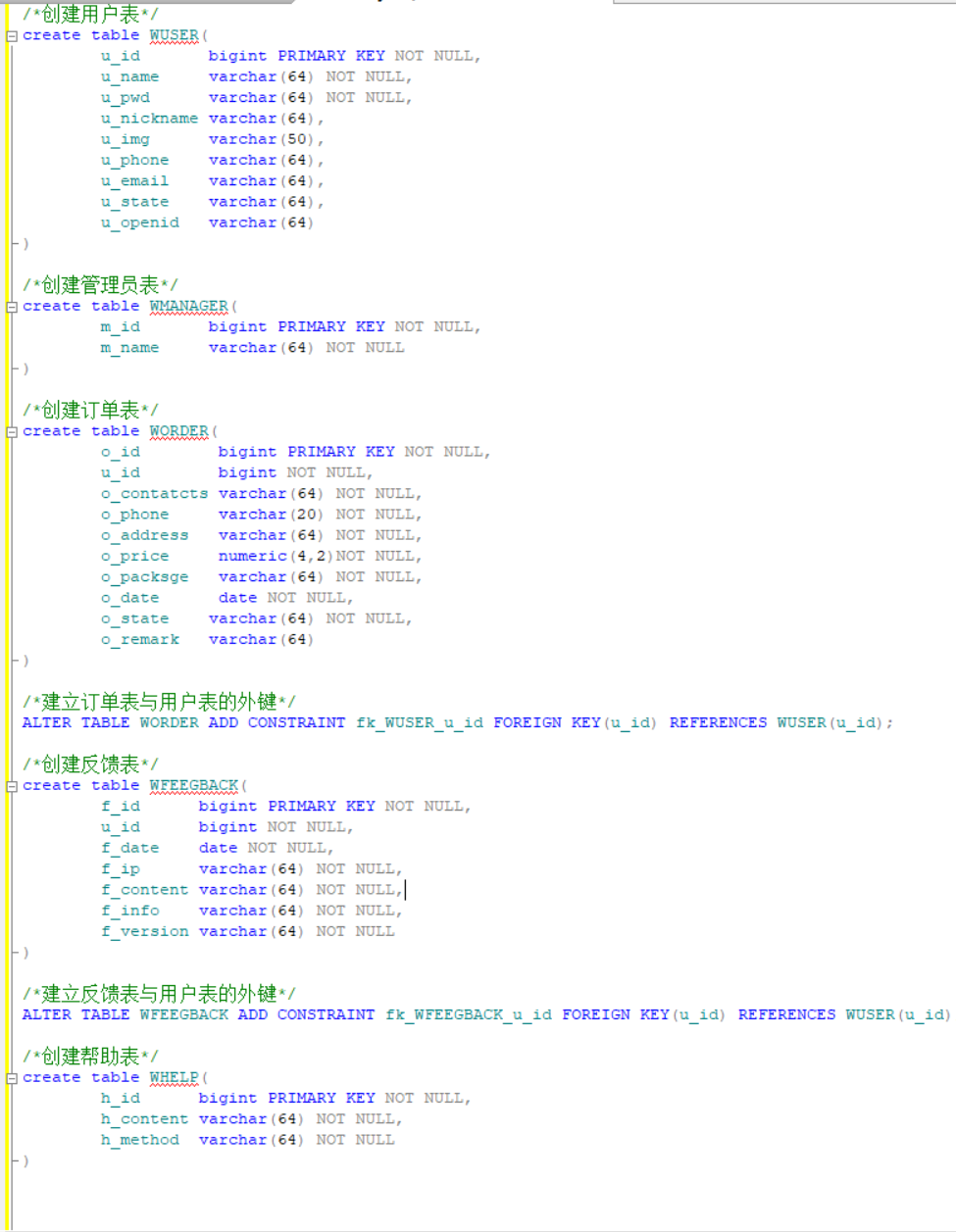
1. **运用设计**

**4.1安全保密设计**

数据库基本安全架构, 用户分类不同类型的用户授予不同的数据管理权限,分为两类，一类是数据库管理员权限类，还有时普通用户群体类。数据分类，同一类权限的用户对数据库中数据管理和使用的范围又可能是不同的。

**4.2数据库设计**

**4.2.1创建表**

****

1. **数据库验收标准**

**5.1数据库验收步骤**

