**<简时>数据库设计说明书**

**所属学院： 数学与计算机科学学院**

**团队名称： time masters**

**指导老师： 单红**

**项目成员： 陈伟杰、许俊鑫、赵梓峻**

**陈俊延、林羽希、谢一新**

**叶如茵、林煜、高雨欣**

**2020年4月**

**目录**

[1.引言](#_Toc6174482) 2

[1.1编写目的 2](#_Toc6174483)

[1.2背景 2](#_Toc6174484)

[1.3参考资料 2](#_Toc6174486)

[2.外部设计 3](#_Toc6174487)

[2.1标识符和状态 3](#_Toc6174488)

[2.2使用它的程序 3](#_Toc6174489)

[2.3约定 3](#_Toc6174490)

[3.结构设计 4](#_Toc6174492)

[3.1概念结构设计 4](#_Toc6174493)

[3.2逻辑结构设计 5](#_Toc6174499)

[4.运用设计 5](#_Toc6174505)

[4.1数据字典设计 5](#_Toc6174506)

[4.2安全保密设计 5](#_Toc6174507)

1. **引言**
   1. **编写目的**

《<简时>数据库设计说明书》目的是设计《简时》系统中各项功能与非功能性需求实现时所需要的的数据库，并可以此数据库为根据实现具体的数据存储，同时为详细设计人员提供设计依据。其他本项目组的开发人员也可参阅。

* 1. **背景**

开发软件名称：简时

数据库名称：time\_masters

项目任务提出者：2020春 S班 单老师

项目开发者：time\_masters团队 (第五组)

用户：备考学生

* 1. **参考资料**

《数据库设计说明书》-国家标准格式

# **2.外部设计**

外部设计是研究和考虑所要建立的数据库的信息环境，对数据库应用领域中各种信息要求和操作要求进行详细地分析，了解应用领域中数据项、数据项之间的关系和所有的数据操作的详细要求，了解哪些因素对响应时间、可用性和可靠性有较大的影响等各方面的因素。

## **2.1标识符和状态**

数据库前缀：TimeMaster\_

用户名：root

密码：123456

权限：全部

有效时间：开发阶段

说明：系统正式发布后，可能更改数据库用户或者密码，在统一位置编写数据库连接字符串，在发行前予以改正

## **2.2使用它的程序**

将要使用此数据库的应用程序名称为简时1.0，本应用程序主要利用Java作为应用开发工具，使用MySQL作为后台的数据库，Android作为系统平台。

## **2.3约定**

1. 所有命名具有描述性，杜绝拼音或者拼音英文混杂的命名方式
2. 字符集采用UTF-8
3. 所有数据表第一个字段都是系统内部使用主键列，自增字段，不可空，名称为id，确保不把此字段暴露给最终用户
4. 除特别说明外，所有字段默认都设置不允许为空，需要设置默认值
5. 所有普通索引的命名都是表名加设置索引的字段名组合，多重索引的命名采用表名\_字段1名\_字段2名的方式

## **2.4支持软件**

操作系统：Android5.0+

数据库系统：MySQL

数据库管理系统：Navicat for MySQL

用于装入、生成、修改、更新数据库的程序：Java,使用Android Studio作为开发工具

# **3.结构设计**

数据库的结构设计中有许许多多需要考虑的因素，如数据库的背景、应用环境，只有一个对所有这些因素都很了解的数据库设计专家，他设计的数据库才能易于使用和维护，并且具有高效和一致的特征。

## **3.1概念结构设计**

概念数据库的设计应该极易于转换为逻辑数据库模式，又容易被用户所理解。概念数据库设计中最主要的就是采用实体-关系数据模型来确定数据库的结构。数据是表达信息的一种重要的量化符号，是信息存在的一种重要形式。数据模型则是数据特征的一种抽象。它描述的是数据的共性，而不是描述个别的数据

### 3.1.1实体和属性的定义

◆ 属性

每个实体都有一组特征或性质，称为实体的属性。实体的属性值是数据库中存储的主要数据，一个属性实际上相当于表中的一个列。下面来看看用户实体。这个实体具有哪些属性呢？对于一个用户来说， 都具有id、用户名、密码、手机号、住址、信誉积分等。所以 关于用户实体的属性如下：

● ID

● 用户名

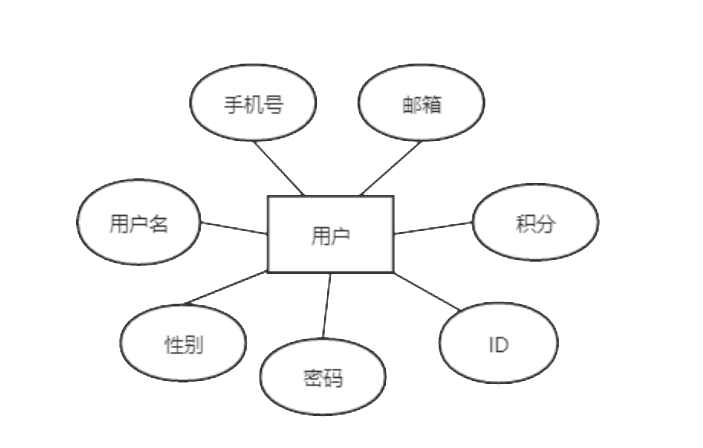
● 密码

● 手机号

● 积分

● 性别

● 邮箱



在绘制 E-R 图中，属性由椭圆包围，在属性和它所属的实体间使用直线进行连接，以实体用户为例进行示例，如上图所示。

由于篇幅有限这里就不列出所有实体的属性

### **3.1.2 E-R图的绘制**

实体-关系图是表现实体-关系模型的图形工具，简称E-R图。这节给出一个完整的数据库的 E-R 图设计示例。

还要为每个关系命名一个“关系名称”，实体与关系相连的直线旁 都根据关系的属性标注有“1”或“N”。E-R图为读者的数据库提供了一个不错的蓝图，可以分成三步进行：首先设计局部 E-R 图；然后合并各局部 E-R 图，并解决可能存在的冲突，得到初步 E-R 图；最后修改和重构初步 E-R 图，消除其中的冗余部分，得到最终的全局E-R图，即概念模式。设计全局 E-R 模式的目的不在于把若干局部E-R模式形式上合并为一个 E-R 模式，而在于消除冲突使之成为能够被全系统中所有用户共同理解和接受的统一的概念模型。使设计人员仅从用户角度看待数据及处理要求和约束，产生一个反映用户观点的概念模式。

### **3.1.3 设计局部E-R模式**

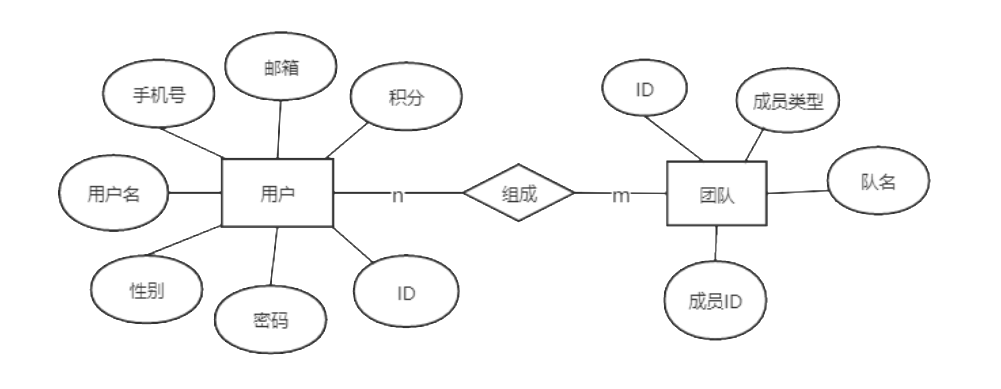
先设计局部E-R图，也称用户视图。在设计初步E-R图时，要尽量能充分地把组织中各部门对信息的要求集中起来，而不需要考虑数据的冗余问题。 局部概念模型设计是从用户的观点出发，设计符合用户需求的概念结构。局部概念模型设计的就是组织、分类收集到的数据项，确定哪些数据项作为实体，哪些数据项作为属性，哪些数据项是同一实体的属性等。确定实体与属性的原则：

■ 能作为属性的尽量作为属性而不要划为实体；

■ 作为属性的数据元素与所描述的实体之间的联系只能是 1:n 的联系；

■ 作为属性的数据项不能再用其他属性加以描述，也不能与其他实体或属性发生联系。

以下是部分局部E-R图的展示



### **3.1.4 设计全局E-R模式**

综合各局部E-R图，形成总的E-R图，即用户视图的集成。所有局部E-模式都设计好了后，接下来就是把他们综合成单一的全局概念结构。全局概念结构不仅要支持所有局部E-R模式，而且必须合理地表示一个完整、一致的数据库概念结构。

另外，在进入下一节之前，先回顾一下概念数据库的设计，其中主要是实体-关系模型的建立。简要总结一下实体-关系模型建立的步骤：

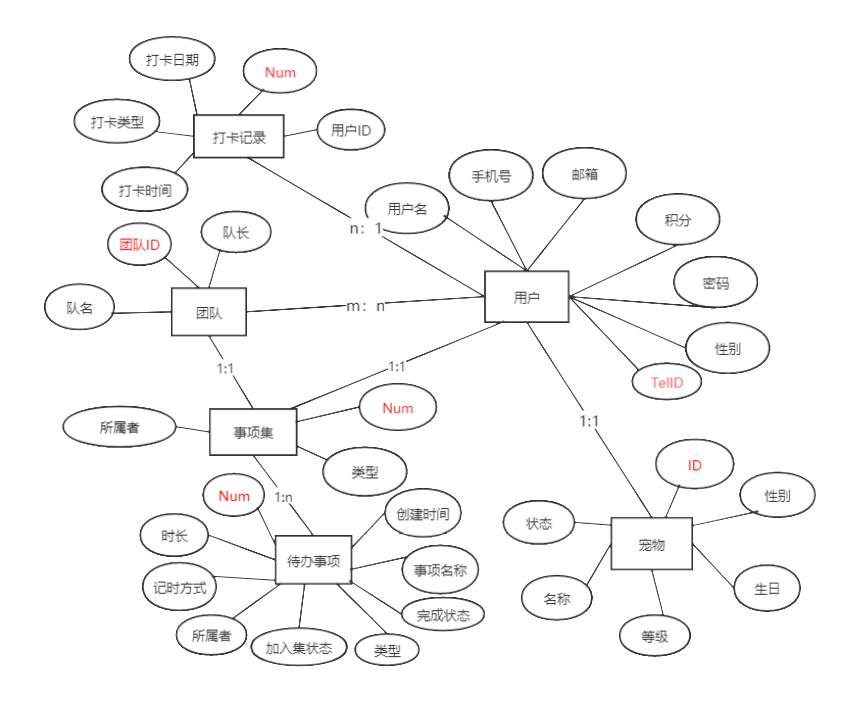
(1)对需求进行分析，从而确定系统中所包含的实体。

(2)分析得出每个实体所具有的属性。

(3) 保证每个实体有一个主属性，该主属性可以是实体的一个属性或多个属性的组合。主属性必须能唯一地描述每个记录。

(4) 确定实体之间的关系。 经过这些步骤后，读者就可以绘制出E-R图。之后可以再看看数据库的需要，判断是否获取了所需的信息，是否有遗漏信息等，读者可以再对E-R图进行修改，添加或删除实体与属性。

以下为全局E-R图的展示



## **3.2 逻辑结构设计**

逻辑结构设计的任务是把概念设计阶段建立的基本E-R图，按照选定的内容，管理系统软件支持的数据模型，转化成相应的逻辑设计模型。也就是可以将实体、实体间的关系等模型结构转变为关系模式，即生成数据库中的表，并确定表的列。下述讨论由实体-关系模型生成表的方法。

■任务：

将基本E-R图转换为与选用简时app所支持的数据模型相符合的逻辑结构。

■ 过程：

1) 将概念结构转换为现有简时app的关系模型。

2) 从功能和性能要求上对转换的模型进行评价，看它是否满足用户要求。

3) 对数据模型进行优化

# **3.3物理结构设计**

数据库设计的最后阶段是确定数据库在物理设备上的存储结构和存取方法，也就是设计数据库的物理数据模型，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。本次项目开发采用的是 MySQL建立数据库。

数据表结构如下：

User

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| ID | int | / | 0 | / | 1 | INC |
| TelID | char | 12 | 0 | / | 1 | / |
| Name | varchar | 30 | 0 | / | 1 | / |
| Email | varchar | 30 | 1 | / | 0 | / |
| Passward | varchar | 30 | 0 | 123456 | 0 | / |
| Sex | enum(‘man’,’woman’) | / | 0 | ‘man’ | 0 | / |

Pet

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| ID | int | / | 0 | / | 1 | INC |
| Name | varchar | 30 | 0 | / | 0 | / |
| Sex | enum(‘male’,’female’) | / | 0 | ‘male’ | 0 | / |
| Level | tinyint | / | 0 | 0 | 0 | / |
| Birthday | date | / | 0 | 1970-01-01 | 0 | / |
| Status | enum | / | 0 | / | 0 | / |
| User | int | / | 0 | / | 0 | User.ID |
| Skin | varchar | / | 0 | ‘init’ | 0 | / |

AccRecords

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| ID | int | / | 0 | / | 1 | INC |
| User | int | / | 0 | / | 0 | User.ID |
| Scount | int | / | 0 | 0 | 0 | / |
| Fcount | int | / | 0 | 0 | 0 | / |
| AccTime | int | / | 0 | 0 | 0 | / |
| DailyTime | int | / | 0 | 0 | 0 | / |

DailyRecords

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| ID | int | / | 0 | / | 1 | INC |
| User | int | / | 0 | / | 0 | User.ID |
| Scount | int | / | 0 | 0 | 0 | / |
| Fcount | int | / | 0 | 0 | 0 | / |
| AccTime | int | / | 0 | 0 | 0 | / |
| Date | date | / | 0 | / | 0 | / |

MonthRecords

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| ID | int | / | 0 | / | 1 | INC |
| User | int | / | 0 | / | 0 | User.ID |
| Scount | int | / | 0 | 0 | 0 | / |
| Fcount | int | / | 0 | 0 | 0 | / |
| AccTime | int | / | 0 | 0 | 0 | / |
| Date | date | / | 0 | / | 0 | / |

TypeRecords

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| ID | int | / | 0 | / | 1 | INC |
| User | int | / | 0 | / | 0 | User.ID |
| Scount | int | / | 0 | 0 | 0 | / |
| Fcount | int | / | 0 | 0 | 0 | / |
| AccTime | int | / | 0 | 0 | 0 | / |
| Type | varchar | / | 0 | / | 0 | / |

ToDoSet

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| Num | int | / | 0 | / | 1 | INC |
| Name | varchar | 30 | 0 | / | 0 | / |
| UserID | char | 12 | 0 | / | 0 | User.ID |
| Type | enum(‘user’,’team’) | / | 0 | user | 0 | / |

ToDo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| Num | int | / | 0 | / | 1 | INC |
| Name | varchar | 30 | 0 | / | 0 | / |
| ToDoSet | int | / | 1 | NULL | 0 | ToDoSet.Num |
| Owner | char | 12 | 0 | / | 0 | User.ID&Team.ID |
| Time | time | / | 0 | 00:00:00 | 0 | / |
| Date | date | / | 0 | 1970-01-01 | 0 | / |
| TimeLength | time | / | 0 | 00:00:00 | 0 | / |
| Type | enum | / | 0 | / | 0 | / |
| Status | enum(‘finish’,’fail’,unfinish’) | / | 0 | ‘unfinish’ | 0 | / |

ClockIn

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| Num | int | / | 0 | / | 1 | INC |
| Name | varchar | 30 | 0 | / | 0 | / |
| UserID | char | 12 | 0 | / | 0 | User.ID |
| Time | time | / | 0 | 00:00:00 | 0 | / |
| Date | date | / | 0 | 1970-01-01 | 0 | / |
| Type | enum | / | 0 | / | 0 | / |

Team

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| ID | char | 12 | 0 | / | 1 | INC |
| Name | varchar | 30 | 0 | / | 0 | / |
| Leader | char | 12 | 0 | / | 0 | User.ID |

Organize

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Length | Allow NULL | Default | Primary Key | Other |
| Num | int | / | 0 | / | 1 | INC |
| UserID | char | 12 | 0 | / | 0 | User.ID |
| TeamID | char | 12 | 0 | / | 0 | Team.ID |

# **3.结构设计**

**4.1 数据字典设计**

对数据库设计中涉及到的各种项目，如数据项、记录等一般要建立起数据字典，以说明它们的标识符、同义名及有关信息。在本节中说明对此数据字典设计的基本考虑。

**4.2 安全保密设计**

通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待

而获得的数据库安全保密设计考虑。对数据库设计中涉及到的各种项目，如数据

项、记录等一般要建立起数据保护措施，以说明它的 标识符、同义名及有关信息。数据库由专门数据库管理用员对数据库操作，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。

**4.3 数据库实施**

**4.3.1 创建数据库**

CREATE DATABASE 数据库名称

**4.3.2 创建表**

CREATE TABLE User (

ID int NOT NULL AUTO\_INCREMENT,

TelNum char(12) NOT NULL,

Name varchar(30) NOT NULL,

Email varchar(30) ,

Password varchar(30) NOT NULL DEFAULT ‘123456’,

Sex enum(‘man’,‘woman’) NOT NULL DEFAULT ‘man’,

Integral int NOT NULL DEFAULT ‘0’,

PRIMARY KEY (ID)

)

CREATE TABLE Pet (

ID int NOT NULL AUTO\_INCREMENT,

Name varchar(30) NOT NULL,

Sex enum(‘male’,‘female’) NOT NULL DEFAULT ‘male’,

Level tinyint NOT NULL DEFAULT ‘0’ ,

Birthday date NOT NULL DEFAULT ‘1970-01-01’,

Status enum NOT NULL,

PRIMARY KEY (ID)

)

CREATE TABLE KeepPets (

Num int NOT NULL AUTO\_INCREMENT,

UserNum char(12) NOT NULL,

FOREIGN KEY(TelNUm),

REFERENCE User(TelNum),

PetID int NOT NULL,

FOREIGN KEY(ID),

REFERENCE Pet(ID),

PRIMARY KEY (Num)

)

CREATE TABLE ToDo (

Num int NOT NULL AUTO\_INCREMENT,

Name varchar(30) NOT NULL,

UserNum char(12) NOT NULL,

FOREIGN KEY(TelNUm),

REFERENCE User(TelNum),

Time time NOT NULL DEFAULT ‘00:00:00’,

Date date NOT NULL DEFAULT ‘1970-01-01’,

TimeLength time NOT NULL DEFAULT ‘00:00:00’,

Type enum NOT NULL,

PRIMARY KEY (Num)

)

CREATE TABLE ClockIn (

Num int NOT NULL AUTO\_INCREMENT,

Name varchar(30) NOT NULL,

UserNum char(12) NOT NULL,

FOREIGN KEY(TelNUm),

REFERENCE User(TelNum),

Time time NOT NULL DEFAULT ‘00:00:00’,

Date date NOT NULL DEFAULT ‘1970-01-01’,

Type enum NOT NULL,

PRIMARY KEY (Num)

)

CREATE TABLE Team (

ID int NOT NULL AUTO\_INCREMENT,

Name varchar(30) NOT NULL,

Leader char(12) NOT NULL,

FOREIGN KEY(TelNUm),

REFERENCE User(TelNum),

PRIMARY KEY (ID)

)

CREATE TABLE Organize (

Num int NOT NULL AUTO\_INCREMENT,

UserNum char(12) NOT NULL,

FOREIGN KEY(TelNUm),

REFERENCE User(TelNum),

TeamID int NOT NULL,

FOREIGN KEY(ID),

REFERENCE Team(ID),

PRIMARY KEY (Num)

)