# 数据库设计

## 1. 概述

本系统采用MySQL5.6 数据库，轻量、快速，且功能强大，移植性较好。

系统数据库设计遵循如下原则：

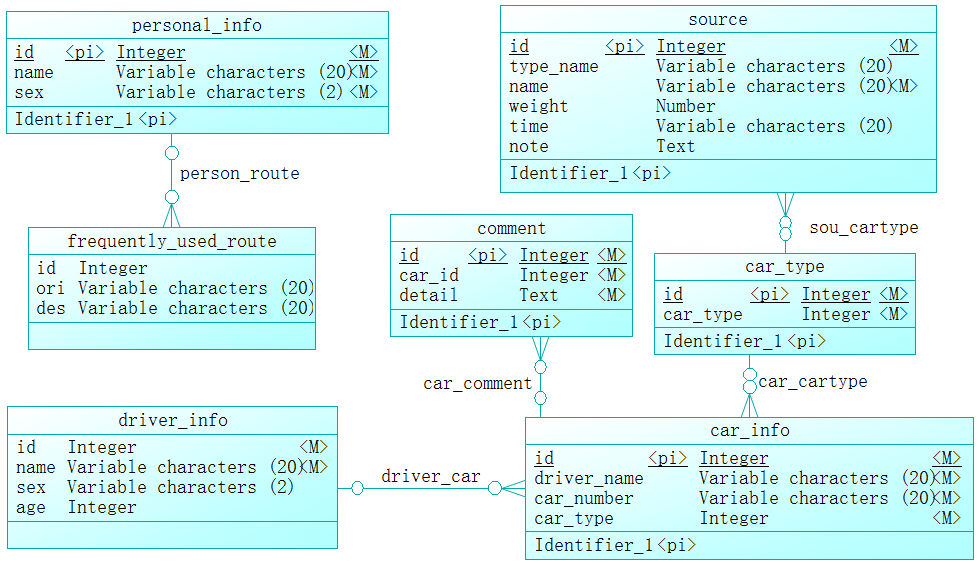
#### 1.1 命名规范

|  |  |  |
| --- | --- | --- |
| 命名项 | 命名规范 | 备注 |
| 表名 | 数据表名称必须以有特征含义的单词或缩写组成，中间以用大写分割词语。表名称不能用双引号包含 | 关系表的表名为 “表名 1 + to+ 表名 2”的形式， 或表名1/2 采用缩写的形式 |
| 字段名 | 主键都设为 ID， 其他字段名称必须用字母开头，采用有特征含义的单词或缩写，不能用双引号包含 | 外键名为被引用的键所在表名/缩写+Id |

#### 1.2 数据类型

|  |  |  |
| --- | --- | --- |
| 数据类型 | 选择准则 | 备注 |
| 字符型 | 固定长度的字串类型采用 CHAR，长度不固定的字串类型采用 VARCHAR。避免在长度不固定的情况下采用 CHAR 类型 | 如果在数据迁移时出现以上情况，则必须使用 trim()函数截去字串后的空格 |
| 数字型 | 数字型字段尽量采用 INT 类型 | 默认长度为 11 |
| 日期和时间 | 由数据导入或外部应用程序产生的日期时间类型采用 DATETIME 类型 | 无 |

## 2. 概念模型设计

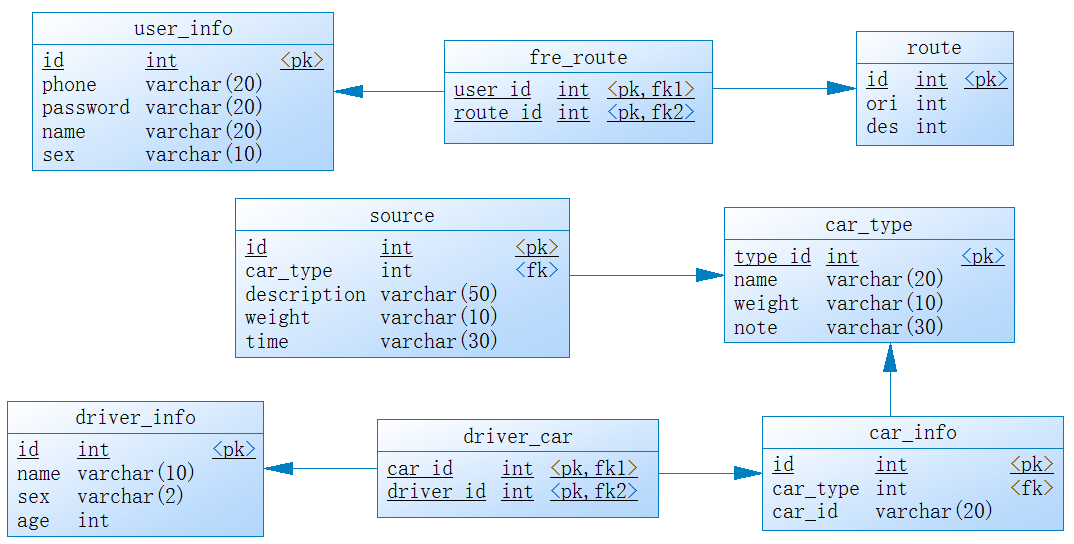


图一 系统概念模型

#### 表信息

|  |  |
| --- | --- |
| Personal\_info | 用户个人信息，包括姓名、性别等 |
| Frequently\_used\_route | 常用路线信息，包括起点、终点 |
| Source | 货源信息，包括需要的车辆类型、重量、时间、备注信息 |
| Car\_type | 车辆的类型信息，比如小型车、中型车 |
| Car\_info | 车辆信息，包括司机姓名、车牌号等 |
| Comment | 评论信息 |
| Driver\_info | 司机信息 |

## 3. 物理模型设计



图二 系统物理模型

#### 表信息

|  |  |  |  |
| --- | --- | --- | --- |
| user\_info 用户基本信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | Primary key | 用户唯一标识符 |
| phone | Varchar(20) | Not null | 手机号码，用作用户登录名 |
| password | Varchar(20) | Not null | 登录密码 |
| name | Varchar(20) |  | 用户姓名 |
| sex | Varchar(10) |  | 用户性别 |

|  |  |  |  |
| --- | --- | --- | --- |
| fre\_route 用户常用路线表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| user\_id | int | primary key,  foreign key | 用户标识符 |
| route\_id | int | primary key,  foreign key | 路线标识符 |

|  |  |  |  |
| --- | --- | --- | --- |
| route 路线表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 路线标识符 |
| ori | varchar(30) | not null | 路线起点 |
| des | varchar(30) | not null | 路线终点 |

|  |  |  |  |
| --- | --- | --- | --- |
| car\_type 车辆类型信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| type\_id | int | primary key | 类型标识符 |
| name | varchar(20) | not null | 类型名 |
| weight | varchar(10) |  | 最大载重 |
| note | varchar(30) |  | 备注信息 |

|  |  |  |  |
| --- | --- | --- | --- |
| driver\_info 司机信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 司机标识符 |
| name | varchar(10) | not null | 司机姓名 |
| sex | varchar(2) |  | 司机性别 |
| age | int |  | 司机年龄 |

|  |  |  |  |
| --- | --- | --- | --- |
| car\_info 车辆信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 车辆标识符 |
| car\_type | int | foreign key | 车辆类型 |
| car\_id | varchar(20) | not null | 车牌号 |

|  |  |  |  |
| --- | --- | --- | --- |
| driver\_car 司机车辆对应表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| driver\_id | int | primary key,  foreign key | 司机id |
| car\_id | int | primary key,  foreign key | 车辆id |

|  |  |  |  |
| --- | --- | --- | --- |
| source 货源信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 货源标识符 |
| car\_type | int | foreign key | 所需车辆类型 |
| description | varchar(50) |  | 货源描述 |
| weight | varchar(10) |  | 重量 |
| time | varchar(30) |  | 时间 |

## 4. 数据结构与程序的关系

服务器程序在对数据进行展示的时候，需对数据库数据结构，也就是数据表进行查询：在数据展示的过程中都需要对数据库中的所有表，进行联合查询。

物理数据结构主要用于各模块之间函数的信息传递。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。出错信息将送入显示模块中。