浙江大学

《地理空间数据库》实习报告

LBS 空间数据库设计——ofo 小黄车

姓　　名　黄欣雨、林益鑫

学　　号　3160102577、3160102599

指导教师　陶　煜　波

专　　业　地理信息科学

所在学院　地球科学学院

提交日期　2018 年 6 月 10 日

目录

一、引言

1、编写目的

对庞大的信息随着共享单车的规模不断扩大，用户使用量急剧增加，有关的各种信息也成倍增长，有必要整合共享单车管理系统来提高管理工作的效率。通过这样的系统，可以做到信息的规范管理、科学统计和快速查询，从而减少管理方面的工作量，同时也可以方便用户对信息的获取。

本系统针对软件界面的人性化，生活化，做了突破性的工作，以及多项管理功能的集成上作了初步的拓展，目的在于使管理者和访问者易于甚至乐于接受。

2、项目背景

软件产品：共享单车管理系统 （ofo 小黄车）

软件用户：普通用户、修理人员以及管理员

本产品是主要针对用户和修理人员的需求设计的，可以完成用户注册、登录、修改信息、注销；寻车、用车；车辆相关信息查询、修改；使用、报修、维修记录的插入、修改。

二、需求分析

（一）任务

1、目标

（1）给出软件系统的数据流程图和数据结构

（2）提出详细的功能说明，确定设计限定条件，规定性能需求

（3）密切与用户的联系，使用户明确自己的任务，以便实现上述两项目标

（4）模拟以最低的成本开发可供共享单车应用的智能管理系统

2、用户特点

本系统主要面向的用户是共享单车的使用者、维修者和运营者，对用户学历知识方面的要求不高，使系统使用易于操作

（二）数据描述

1、静态数据

静态数据是系统内部有关的数据结构和操作规程。具体包括用户信息表、共享单车车辆信息表、使用记录表、使用状态表、报修记录表、维修站信息表、维修人员信息表、维修记录表

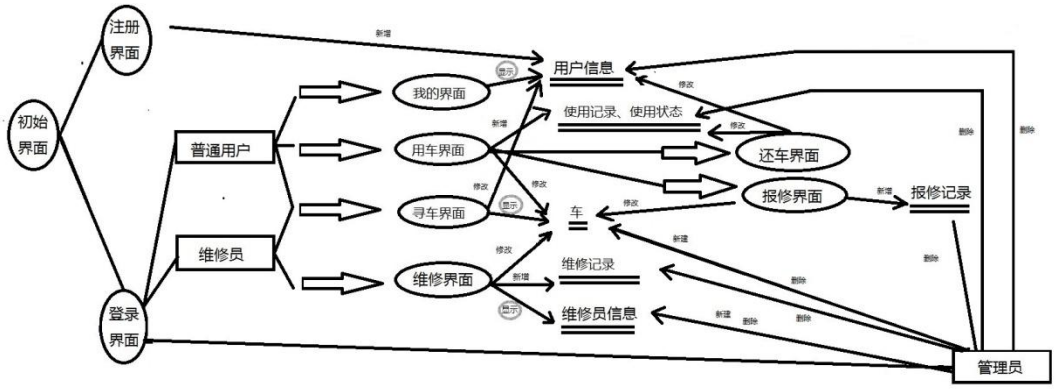
2、动态数据

动态数据指程序运行中会发生变化的数据，又可下分为输入动态数据和输出动态数据，具体于本项目主要指的就是 用户关系表中除了 ID 的所有数据以及其他关系中的 time、position、status 类时间、空间、状态属性的数据

3、数据描述

根据上述分析设计出各种数据实体，以及它们之间的关系，为后面的逻辑结构设计打下基础，这些实体包括各种具体信息，通过相互之间的作用形成数据的流动

4、数据流图



（三）需求规定

1、功能描述

（1）注册功能：有新的注册的时候验证该用户 id 是否存在，不存在则添加该新用户记录

（2）登录功能：验证登录用户是否为数据库中的合法用户，判断是否为普通用户还是维修人员还是内部管理员。普通用户能实现查看、搜索和对报修记录的修改功能，维修人员可以实现修改车辆的 repair_status 和查看、搜索功能，管理员可以实现对主要信息的修改、查看、添加、删除功能

（3）寻车功能：对车的 ID、position 具有浏览查看搜索功能，利用函数寻找位置附近 300 米的符合用户需求的车

（4）用车功能（普通用户）：在正确输入密码开锁瞬间修改车的 using_status，新增使用记录和使用状态，关锁瞬间对使用记录、using_status 进行对应修改。

（5）还车功能：用户用车结束后，关锁瞬间转入还车页面，并对用户账户余额进行修改

（6）报修功能：在用车时，用车后提供报修功能，插入报修记录，由用户输入详情

（7）维修功能（维修人员）：新增并修改维修记录，对应修改车的 repair_status

2、界面需求：

（1）初始界面：用于用户选择，已有用户进入登录界面，新用户进入注册界面

（2）注册界面：新用户用手机号注册后加该用户信息添加到用户表中

（3）登录界面：由已有用户用账号登录，根据用户类型可以分为普通用户、维修人员与管理员

（4）我的界面：普通用户登录后可以点击我的界面按钮进入我的界面，该界面用以显示用户的各类信息，如骑行的历史记录、押金、红包等

（5）寻车界面：普通用户与维修员登入后都将可以进入寻车界面。对普通用户显示附近 300 米左右范围内的可用小黄车位置，对维修人员显示附近的坏车位置

（6）用车界面：当普通用户选择扫码识别车辆时，将进入用车界面（本实习中默认小黄车车锁均为扫码后获取固态密码在车锁上输入密码后自动开锁），显示密码，用户在车锁上输入密码后锁自动打开，同时向系统发出信号修改使用记录表、使用状态表信息，对车的 using_status 更改为占用

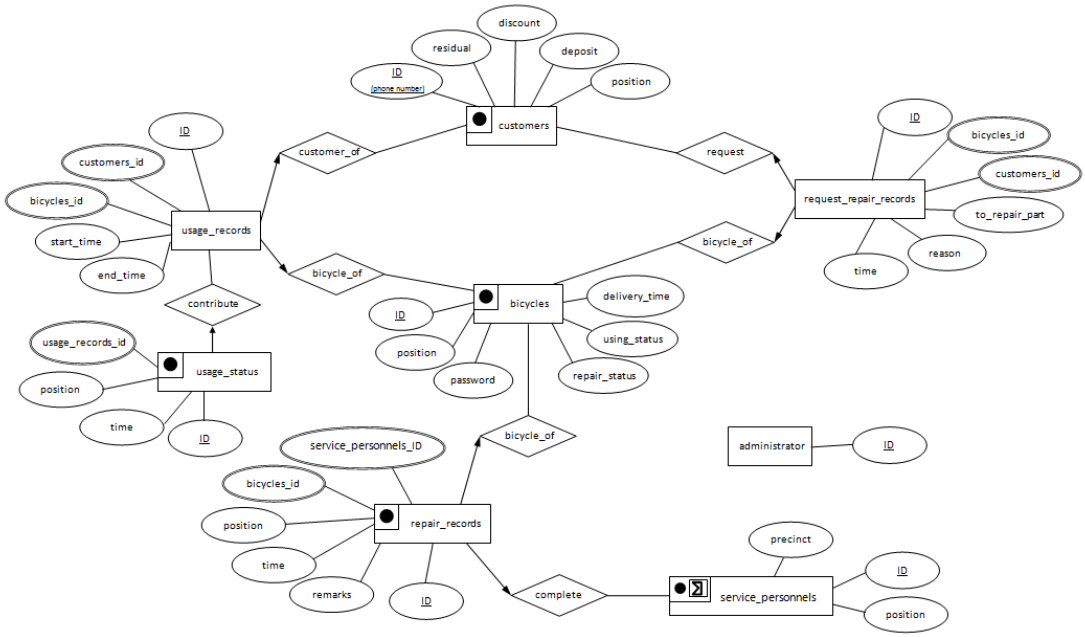
（7）维修界面：维修员到达指定车辆后，通过扫描该车二维码进入对应的维修界面，查看报修原因后开始维修。完成维修后填写备注点击维修完成系统将该车的 repair_status 改成已维修，新增一条维修记录

（8）还车界面：当用户到达目的地关锁时，系统接收到关锁信息，于是更新使用记录中的结束时间，并且结束使用状态的录入，显示时长并以此计算出车费（认定车费为 1 元/小时）。用户确认支付后根据折扣扣除余额，更新用户信息

（9）报修界面：用户也可以进入报修界面对当前车进行报修，提交后将对报修记录执行新增处理，对车的 repair_status 更新为故障

三、概念设计

在本系统中涉及到的实体有用户、车辆、使用记录、使用状态、报修记录、维修人员、维修记录、管理员。当用户每使用一次共享单车时就会添加一条使用记录，同时使用状态将开始每隔五秒对当前车辆位置状态进行记录。当用户对某一辆车报修时，报修记录新增。维修人员对某辆车完成修理后会对维修记录执行添加操作，并且更新车辆 repair_status 属性。管理员在后台对数据具有查看、更新、修改、定时清理等作用。



1、customers 关系中 ID（即用户的 phone number）是该关系的主键，residual 不能为负，position 为用户的位置点 geometry::point

2、bicycles 关系中 ID 是该关系的主键，passwords 统一为 4 位阿拉伯数字，position 为车的位置点 geometry::point

3、usage_records 关系中 ID 为该关系主键，customers_id 是外键依据 customers 关系中的 ID，bicycles_id 是外键依据 bicycles 关系中的 ID

4、usage_status 关系中 ID 为主键，bicycles_id 是外键依据 bicycles 关系中的 ID，position 为当前时间点下车的位置点 geometry::point

5、request_repair_records 中 ID 是主键，customers_id 是外键依据 customers 关系中的 ID，bicycles_id 是外键依据 bicycles 关系中的 ID

6、repair_records 中 ID 是主键，其中 service_personnels_id 也是外键依据 service_personnels 关系中的 ID，bicycles_id 是外键依据 bicycles 关系中的 ID，position 为当前位置点 geometry::point

7、service_personnels 关系中 ID 为主键，position 为修理人员位置点 geometry::point，precinct 为修理人员管辖区范围 geometry::polygon

8、administrator 具有属性 ID，ID 为该关系主键

四、逻辑设计

（一）customers

属性	类型	其他
ID	Varchar(15)	Primary key
Position	Geometry(point)	寻车时更新用户位置，搜索附近车辆
Residual	Double	每次用车后自动更新余额
Discount	Double	通过各种软件的活动获得的折扣
Deposit	Double	注册时所交押金

函数依赖：ID →Position，Residual，Discount，Deposit。符合 BC 范式。

（二）Bicycles

属性	类型	其他
ID	Varchar(15)	Primary key
Position	Geometry(point)	每次用车后更新位置
Password	Varchar(10)	新建时所设密码
Delivery_time	Timestamp	该车出厂时间
Using_status	Bool	是否正在使用
Repair_status	Bool	是否需要修理

函数依赖：ID →Position，password，Delivery_time，Using_status，Repair_status。符合 BC 范式。

（三）usage_records

属性	类型	其他
ID	Varchar(15)	Primary key
Customers_id	Varchar(15)	Foreign key – customers
Bicycles_id	Varchar(15)	Foreign key – bicycles
Start_time	Timestamp	用车时更新
End_time	Timestamp	还车时更新

函数依赖：ID → Customers_id，Bicycles_id，Start_time，End_time。符合 BC 范式。

（四）usage_status

属性	类型	其他
ID	Varchar(15)	Primary key
Usage_records_id	Varchar(15)	对应使用记录
Bicycle_id	Varchar(15)	对应发出位置信号的车号
Time	Timestamp	记录录入的当前时间
Position	Geometry(point)	用车时、还车前定时录入

函数依赖：ID →Usage_records_id，Bicycle_id，time，position。符合 BC 范式。

（五）request_repair_records

属性	类型	其他
ID	Varchar(15)	Primary key
Customers_id	Varchar(15)	Foreign key – customers
Bicycles_id	Varchar(15)	Foreign key – bicycles
Time	Timestamp	报修时间
To_repair_part	Text	损坏部位
Reason	Text	报修的具体描述

函数依赖：ID → customers_id，Bicycles_id，Time，To_repair_part，Reason。符合 BC 范式。

（六）repair_records

属性	类型	其他
ID	Varchar(15)	Primary key
Service_personnels_id	Varchar(15)	Foreign key - service_personnels
Bicycles_id	Varchar(15)	Foreign key – bicycles

Position	Geometry(point)	修车地点
Time	Timestamp	修车时间
Remarks	Text	修车过程与结果的具体描述

函数依赖：ID →Service_personnels_id，Bicycles_id，time，position，remarks。符合 BC 范式。

（七）Service_personnels

属性	类型	其他
ID	Varchar(15)	Primary key
Position	Geometry(point)	当前位置
precinct	Geometry(polygon)	该维修人员的责任区

函数依赖：ID →position，precinct。符合 BC 范式

（八）Adminstrator

属性	类型	其他
ID	Varchar(15)	Primary key

其他属性暂不考虑

五、物理设计

1、性能要求

（1）精度：查询时应保证查询率，所有在相应域中包含查询关键字的记录都应能查到，同时保证准确率。

（2）时间特性要求：一般操作的响应时间应在 1—2 秒内。

（3）适应性：允许操作系统之间的安全转换和与其它应用软件的独立运行要求。

（4）灵活性：在需求发生变化时，本系统的对这些变化的适应能力相 对而言是比较强的，

包括操作方式上的变化；运行环境的变化；同其他软件的接口的变化；精度和有效时限的变化。

高峰期认为主要集中在普通用户查询附近车辆中，所以可以对查询操作涉及到的属性建立索引

create unique index selb on bicycles(repair_status asc,using_status asc);

2、触发器举例

```
//在对维修记录进行插入时，修改当前维修完成车辆的 repair_status
create or replace function updaterepairstatus()
returns trigger as $$
begin
   update bicycles set repair_status=0    where bicycles.id=new.bicycle_id;
   return new;
end;
$$language plpgsql;
create trigger updaterepairstatus_tr
after insert on repair_records
for each row
execute procedure updaterepairstatus();

//用户报修时对车辆 repair_status 属性予以修改
create or replace function requestrepair()
returns trigger as $$
begin
   update bicycles set repair_status=1    where bicycles.id=new.bicycle_id;
   return new;
end;
$$language plpgsql;
create trigger requestrepair_tr
```

```
after insert on request_repair_records
for each row
execute procedure requestrepair();

//当用车完毕后对该用户的账户余额予以扣除
create or replace function accountmanage()
returns trigger as $$
declare
   dis float;
   res float;
   pay int;
begin
   dis:=(select discount from customers where customers.id=new.id);
   res:=(select residual from customers where customers.id=new.id);
   pay:=((select extract(day,end_time) from new)-(select extract(day,start_time) from new))*24
        +(select extract(hour,end_time) from new)-(select extract(hour,start_time) from new);
   if (select extract(minute,end_time) from new)>(select extract(minute,start_time) from new)
   then pay:=pay+1;
   end if;
   if dis>=pay
   then dis:=dis-pay;
          update customers set discount=dis where customers.id=new.id;
          return new;
   else
      update customers set discount=0 where customers.id=new.id;
      res:=res-(pay-dis);
      if res>=0 then
      update customers set residual=res where customers.id=new.id;
      return new;
      else raise notice"insufficient account,please refill it";
```

```
        return null;
    end if;
end;
$$language plpgsql;
create trigger accountmanage_tr
before update on usage_records
for each row
where new.end_time!=null
execute procedure accountmanage();
```

3、sql 语句举例

//id 为 13388866677 的用户查询附近 300 米范围内可用车辆

```
select b.position from bicycles b, customers c where b.using_status=0 and b.repair_status=0 and
ST_Dwithin(b.position::geography, c.position::georaphy, 300) and c.ID=”13388866677”;
```

//id 为 246 的用户在行程结束后报修当前车辆,选择了维修部位为 a，输入了维修原因 rea

```
insert into request_repair_records(id,bicycles_id,customers_id,to_repair_part,reason,time)
values
((select max(id) from request_repair_records)+1,
    (select bicycles_id from usage_records where customers_id=246 and end_time>=all(select
end_time from usage_records where customers_id=246)),246,a,rea,CURRENT_TIMESTAMP)
```

//id 为 27 的维修人员查询附近 200 米范围内待修理车辆

```
select b.position from bicycles b,customers c where
ST_Dwithin(b.position::geography,c.position::georaphy,200)
and b.repair_status=1 and c.ID=27
```

//管理员查询所有维修人员辖区内待修理车辆数量

select count(*) as count from bicycles b,service_personnels s where

ST_intersects(s.precinct,b.position) and b.repair_status=1 group by s.id

4、权限分配

（1）普通用户：

----Select：bicycle-ID，position；Customer-ALL

----Update：request_repair_records-to_repair_part，reason

（2）维修人员：

----Select：bicycle-ID-ALL，service_personnels-ALL，request_repair_records-ALL

----Update：repair_records-remarks

（3）管理员

---Delete：ALL

---Select：ALL

---Update：ALL, except for usage_status

---Insert：bicycle，service_personnels