# 数据库设计/接口设计文档

姚正公

2016年10月9日

1. 关键标识释义（对应全文出现的特别词汇）：
2. 用户权限等级：分为3级，1级中心管理员（直接写入数据库），2级高级管理员（中心管理员设置），3级运维人员（中心管理员、高级管理员设置）
3. 业务员：每家企业用户，在验证成功建立了代表人账户以后，可以自行添加业务员账户，用来协同处理和分担事务，其中【所属企业Id】即为企业用户【用户Id】，

业务员页面显示的页面与企业用户相同，但是没有【增加删除修改业务员】页面

4、强制审核用户打分表：是否标记了【理由】，如果标记了则该条打分要经过审 核理由之后才能够算入平均评分的计算之中，并且才能够将该条评论内容加入到【产品 评论】表中，一旦审核之后，该条记录必须从该表中删除

1. 评论归属标识码：标记该评论的所属产品，即为对应的保险产品【产品Id】
2. 该条打分唯一Id：对应产品评论中的【打分Id】
3. 用户发帖：用户发帖中的帖子编号为该帖子的创建编号，当该帖子为原创，则帖子编号与原帖编号相同，都指向最初创建该帖子内容时的那个帖子编号，如果这个帖子为分享帖，则帖子编号与原帖编号不同，原帖编号指向该帖所分享内容的最原始帖子的编号，而帖子编号则是该分享帖的创建编号
4. Id生成策略公元纪年的后两位加上一年中的第几天加上时刻（秒数）一个静态全局同步累加器返回的数值（三位数值）
5. Model
6. 管理员

（1）用户Id（2）用户名（3）用户密码（4）用户权限等级（5）电子邮箱

（6）电话（7）其他联系人（8）电话（9）上次更改时间

1. 企业用户

（1）用户Id（2）用户名（3）用户密码（4）企业微信OpenId

（5）用户实名制注册公司名（6）证明材料存储路径（7）代表人姓名

（8）身份证号（9）电话（10）电子邮箱（9）信息更新时间

1. 业务员

（1）业务员Id（2）所属企业Id（3）用户名（4）用户密码（5）微信OpenId

（6）业务员姓名（7）身份证号（8）电话（9）电子邮箱

（10）信息更新时间

1. 一般用户

（1）用户Id（2）用户名（3）用户密码（4）微信OpenId（5）电子邮箱

（6）电话（7）信息更新时间

1. 保险企业

（1）序号（2）企业名（3）产品归属识别码（4）信息更新时间

（5）关联的企业用户Id（对应：企业用户【用户Id】）

1. 保险产品

（1）产品Id（2）产品名（3）所属企业Id（4）所属企业名

（5）产品发售时间（6）所属险种（数字编号）（7）产品简介

1. 保险产品评价（该表作废，参考开发人员须知295行的说明进行修改）

（1）产品Id（2）综合评分（3）服务评分（4）用途定位评分（5）理赔处理

1. 强制审核用户打分（该表作废，参考开发人员须知295行的说明进行修改）

（1）产品Id（2）打分用户（3）服务打分（4）用途定位打分（5）理赔处理打分

（6）该条打分唯一Id （7）建立日期（8）评论内容

1. 险种列表

（1）数字编号（2）险种名

1. 产品评论

（1）编号（2）评论归属标识码（即产品Id）（3）所属用户Id（4）评论建立时间

（5）评论内容 （6）分数（7）理由/吐槽（classify）

1. 法律条文

重新设计匹配搜索算法

1. 法律案例

基于lucene

1. 用户发帖

（1）帖子编号（2）发帖用户Id（3）原帖编号（4）是否为分享

（5）帖子创建时间（6）帖子缩略文字内容（7）缩略图片存储路径

（8）点赞数量（9）评论数量（10）所属险种编号

1. 帖子原始内容

（1）原帖编号（2）帖子内容（3）图片存储路径

1. 用户评论

（1）评论编号（2）所属帖子编号（3）发表评论用户Id（4）评论创建时间

（5）评论内容

16、用户档案信息

（1）用户年龄（2）性别（3）民族（4）文化程度（6）学业涉足领域

（7）职业涉足领域（8）是否已婚（9）孩子数量（10）老人数量

（11）是否在职（12）待遇水平（13）是否购买过保险（14）购买过的险种

1. 数据库

Create database dpinsurance;

Create table admin(admin\_id varchar（13） not null primary key , admin\_name varchar(32) not null , admin\_password varchar (32) not null , admin\_level smallint not null , admin\_email varchar(256) not null , admin\_phone varchar(32) not null , admin\_colleague varchar(32) not null , admin\_colleague\_phone bigint not null , update\_time datetime not null) engine = innodb default charset = gbk ; //管理员

Create table enterpriseuser(enterprise\_user\_id varchar（13） not null primary key , enterprise\_user\_name varchar(256) not null , enterprise\_user\_password varchar(32) not null , enterprise\_user\_openid varchar(256) , enterprise\_name varchar(256) not null , proof\_path varchar(512) not null , proof\_name varchar(512) not null , representative\_name varchar(128) not null , representative\_idcard varchar(64) not null , representative\_phone varchar(32) not null , representative\_email varchar(256) not null , update\_time datetime not null) engine = innodb default charset = gbk ; //企业用户

Create table clerk(clerk\_id varchar（13） not null primary key , enterprise\_id varchar（13） not null , clerk\_nickname varchar(256) not null , clerk\_password varchar(32) not null , clerk\_openid varchar(256) not null , clerk\_name varchar(128) not null , clerk\_idcard varchar(64) not null , clerk\_phone varchar(32) not null , clerk\_email varchar(256) not null , update\_time datetime not null) engine = innodb default charset = gbk ; //办事员

Create table user(user\_id varchar（13） not null primary key , user\_name varchar(256) , user\_password varchar(32) , user\_openid varchar(256) , user\_phone varchar(32) , user\_email varchar(256) l , update\_time datetime not null) engine = innodb default charset = gbk ; //一般用户

Create table insurance\_enterprise(enterpriseid varchar（13） not null primary key , enterprise\_name varchar(256) not null , relevant\_enterprise\_user\_id int not null , update\_time datetime not null) engine = innodb default charset = gbk ; //保险企业

Create table insurance\_product(product\_id varchar（13） not null primary key , product\_name varchar(256) not null , enterprise\_id int not null , enterprise\_name varchar(256) not null , category\_number int not null , product\_introduce varchar(40000) not null,update\_time datetime not null) engine = innodb default charset = gbk ; //保险产品

Create table insurance\_product\_evaluate(product\_id varchar（13） not null primary key , service\_score float(3,2) not null , positioning\_score float(3,2) not null , settlement\_score float(3,2) not null) engine = innodb default charset = gbk ; //保险产品评价

Create table temp\_insurance\_product\_evaluate(id varchar（13） not null primary key auto\_increment, product\_id bigint not null , owner\_user\_id bigint not null , overall\_score float(3,2) not null , service\_score float(3,2) not null , positioning\_score float(3,2) not null , settlement\_score float(3,2) not null , comment\_content varchar(512) not null , create\_time datetime not null) engine = innodb default charset = gbk ; //强制审核用户打分

Create table insurance\_category (category\_numeber varchar（13） not null primary key , category\_name varchar(256) not null ) engine = innodb default charset = gbk ;

//保险种类

Create table insurance\_product\_comment(comment\_product\_id int not null primary key, owner\_user\_id bigint not null , comment\_content varchar(512) not null , create\_time datetime not null) engine = innodb default charset = gbk ; //产品评论

Create table posting\_item(posting\_id varchar（13） not null primary key , posting\_user\_id bigint not null , posting\_original\_id varchar（13） not null , point\_like\_number int not null , comment\_number int not null , posting\_outline\_content varchar(256) not null , posting\_is\_copied int not null , thumbnails\_path varchar(128) not null , create\_time datetime not null , category\_number int not null ) engine = innodb default charset = gbk ; //用户发帖（简介）

Create table posting\_content(posting\_original\_id varchar（13） not null primary key , posting\_content varchar(40000) not null , pictures\_path varchar(128) not null ，create\_time datetime not null ) engine = innodb default charset = gbk ; //帖子内容

Create table posting\_comment(comment\_id int not null primary key auto\_increment , posting\_id bigint not null , owner\_user\_id bigint not null , comment\_content varchar(512) not null , create\_time datetime not null ) engine = innodb default charset = gbk ; //用户评论

Create table user\_information(user\_id varchar（13） not null primary key , user\_age tinyint not null , user\_gender varchar(4) not null , user\_nation\_code tinyint not null , education\_level\_code tinyint not null , study\_area tinyint not null , work\_area tinyint not null , is\_married tinyint not null , have\_children\_number tinyint not null , is\_having\_job varchar(4) not null , oldman\_number tinyint not null , salary\_level\_code tinyint not null , have\_insurance varchar(4) not null , categorys\_insurance int not null , create\_time datetime not null ) engine = innodb default charset = gbk ;

//用户档案信息

Create table tool\_nation(nation\_code tinyint not null , nation\_name varchar(64) not null ) engine = innodb default charset = gbk ; //民族编码表

Create table tool\_education(education\_level\_code tinyint not null , education\_level\_name varchar(64) not null ) engine = innodb default charset = gbk ;

//教育水平编码表

Create table tool\_area(area\_code tinyint not null , area\_name varchar(64) not null ) engine = innodb default charset = gbk ; //涉足领域编码表

1. 接口设计
2. 系统平台主页面（公共信息、新闻、企业产品排名等的展示，三种身份：一般用户、企业用户、游客（游客无需登陆，直接浏览网站））
   1. 用户登录页面
   2. 用户注册页面
   3. 用户账户信息管理页面
3. 管理员登录后台模块
   1. 管理员管理系统主页面
   2. 登录页面
   3. 角色管理页面
      1. 本账户信息修改页面
      2. 高级管理员、运维人员添加页面
      3. 高级管理员、运维人员删除、修改页面
   4. 平台性能管理
      1. Cpu性能分析
      2. 内存使用情况
      3. 磁盘使用情况
      4. 流量统计
   5. 平台数据管理
      1. 评分评论查看审核统计
      2. 产品发布删除申请审核页面
      3. 用户账户信息查看、删除与统计
      4. 用户档案信息的查看、删除与统计
      5. 其它实时数据统计
4. 企业用户登录模块
   1. 登录页面
   2. 企业用户管理页面
      1. 申请发布删除产品页面
      2. 评论查看和回复页面
      3. 添加删除修改业务员页面
5. 保险点评模块（主要构件：企业名搜索栏，吐槽（互助圈链接），企业用户快速入口链接，企业名列表）
   1. 企业搜索栏
   2. 某保险企业产品列表（主要构件：产品搜索栏，产品名列表）
      1. 产品搜索栏
      2. 某产品页面（营业网点去掉，若是通过企业快速入口登录，则通过该页面回复评论，显示的用户名为企业用户名（业务员微信号也必须登录才能将回复的用户名以企业名显示））
6. 互助圈模块（主要构件：保险类别列表，帖文列表）
   1. 单一帖文项目（构件包括：帖文，图片，点赞/评论/分享功能栏）
      1. 点击进入某帖文页面（页面主要构件：帖文（带先是全文按钮），图片原文，点赞及点赞数量显示按钮，评论输入框，用户评论显示列表）
      2. 发表评论时，如若未登录，会跳出登录页面要求登陆
7. 法律条文/案例模块（主要构件：页顶推荐文章，法律条文/案例选择列表，条文/案例搜索栏，条文/案例列表）
   1. 点击法律条文/案例列表项，进入原文页，显示文章或条文
8. 用户个人信息档案
9. 法律条文/法律案例模块
   1. 模块结构设计
   2. 匹配关键字算法分析
10. 外部数据抓取模块
    1. 模块结构设计
    2. 算法分析
11. 注意事项
    1. 系统采用REST风格
    2. 变量命名规范，意义明确，采用匈牙利命名法
    3. Spring框架使用xml与注释相结合的方式，javabean使用xml配置，其他配置使用注释
    4. 方法必须配有文字注释，包括：方法用途，方法参数列表释义，返回值释义，特殊局部变量说明等
    5. 注意高并发时的IO时延，尽量减少一次请求对应多次数据库IO的情形
    6. 注意代码及数据库同步问题