**ECP**

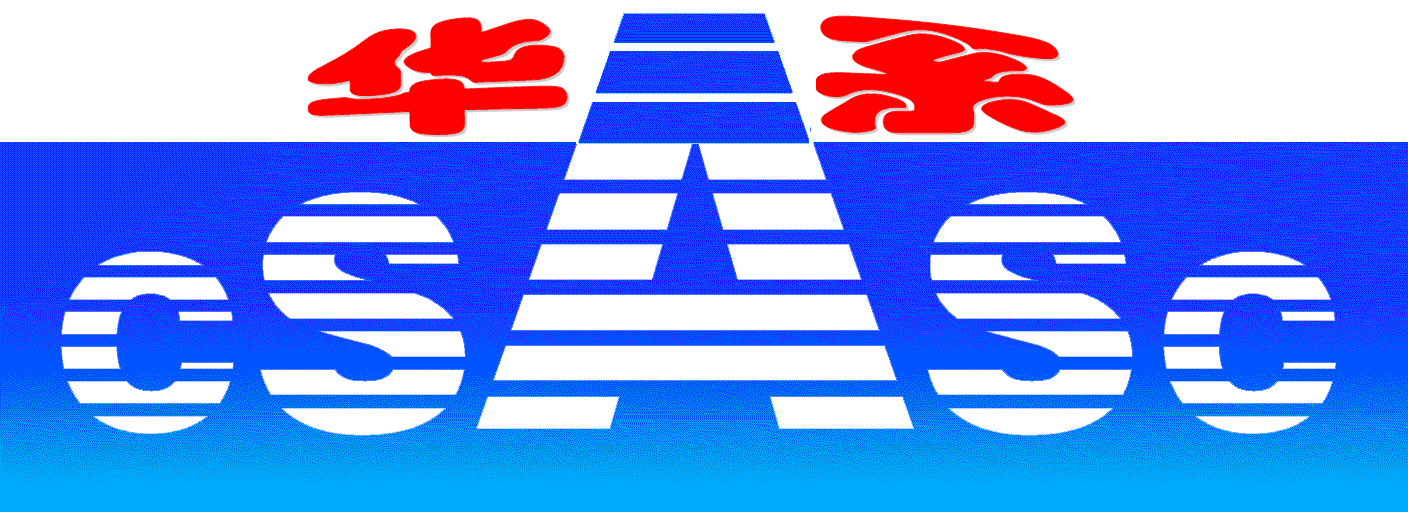
系统实现详细设计

**数据库设计**

**作者：王文浩 郭靖 林梓越**

**华南理工大学计算机系统研究所**

**Research Institute of Computer Systems at South China University of Technology**





2015-02-02 广州

**地址**:①510641 广州市 天河区五山路 华南理工大学 半山东路 四号楼东北门

②510006 广州市 广州大学城外环东路 华南理工大学 B3大楼501,334-336

**Email**: [autosys.adm@gmail.com](mailto:autosys.adm@gmail.com) Tel: 020-88230838,13602713116

ECP实现的技术文档系列：

* ECP系统界面设计

系统功能与模块划分；系统的用户交互界面设计

* ECP系统实现的详细设计

数据库设计；数据访问接口定义；其他类与接口定义

目 录

目录

[第 1 章 数据库设计介绍 4](#_Toc6757)

[第 2 章 数据库表清单 5](#_Toc16425)

[2.1 双创平台集市功能表清单 5](#_Toc25524)

[2.2 双创平台店铺模块表清单 5](#_Toc6139)

[用户查看消息主页查看相应的消息信息，属于用户的消息 5](#_Toc17955)

[订单支付完成之后，用户发货，形成商品的物流信息 6](#_Toc9123)

[2.3就业平台数据库表清单 6](#_Toc14443)

[2.4用户模块表清单 7](#_Toc681)

[第 3 章 主要表关联关系 8](#_Toc6371)

[第 4 章 数据库对应表关系 9](#_Toc2886)

[4.1 市场模块数据库设计 9](#_Toc6419)

[4.1.1供应表 9](#_Toc29755)

[4.1.2产品类别表 9](#_Toc31434)

[4.1.3供应详情表 9](#_Toc28152)

[4.1.4收藏表 10](#_Toc21261)

[4.1.5需求表 10](#_Toc11874)

[4.1.6需求服务商关联表 11](#_Toc26969)

[4.1.7订单表 11](#_Toc19331)

[4.1.8支付记录 11](#_Toc15881)

[4.1.9售后记录 12](#_Toc25820)

[4.2 店铺模块数据库设计 12](#_Toc22756)

[4.2.1店铺表 12](#_Toc22586)

[4.2.2消息表 13](#_Toc18715)

[4.2.3店面消息表 13](#_Toc26170)

[4.2.4审批表 14](#_Toc19013)

[4.2.5物流信息表 14](#_Toc3039)

[4.3 就业平台数据库设计 14](#_Toc24082)

[4.3.1 招聘岗位信息表 15](#_Toc16871)

[4.3.2 用户求职申请表 16](#_Toc26833)

[4.3.3 合同表 16](#_Toc6084)

[4.3.4 消息表 17](#_Toc3845)

[4.3.5合同模板表 19](#_Toc3942)

[4.3.6招聘会表 20](#_Toc17547)

[4.3.7招聘会审核表 20](#_Toc2097)

[4.3.8 招聘会企业表 21](#_Toc16942)

[4.3.9 岗位分类维度表 21](#_Toc15067)

[4.3.10 消息类别维度表 22](#_Toc31896)

[4.3.10求职状态维度表 22](#_Toc5382)

[4.4 用户模块 23](#_Toc20475)

[4.4.1用户注册信息表 23](#_Toc31437)

[4.4.2用户角色表 24](#_Toc28147)

[4.4.3角色表 24](#_Toc26017)

[4.4.4角色权限表 24](#_Toc18275)

[4.4.5个人基本信息表 25](#_Toc21290)

[4.4.6机构基本信息表 25](#_Toc14946)

[4.4.7证件表 26](#_Toc19458)

[4.4.8企业状况表 26](#_Toc19575)

[4.4.9个人经历表 27](#_Toc18827)

[4.4.10学历维度表 28](#_Toc15273)

[4.4.11职称维度表 28](#_Toc17721)

[4.4.12行业类型维度表 28](#_Toc11962)

[4.4.13证件类型维度表 29](#_Toc14077)

[4.4.14岗位类型维度表 29](#_Toc31952)

[第 5 章 详细字段信息 29](#_Toc18608)

[5.1就业平台模块数据库 30](#_Toc8562)

[5.2用户数据库 31](#_Toc25787)

# **数据库设计介绍**

数据库的设计包括了创新创业平台店面功能、集市功能的数据库设计。

各功能模块的数据组成统一的数据库系统，数据库的设计包含了数据库的表结构设计，字段详细信息，各表的关联信息等。

用户模块主要包含系统涉及的各种用户主体，主要分为两类，个人和机构。机构又可以进一步分为公司，企业，事业单位，政府机构，非政府机构等类型。每个主体都有一张状况表（或经历表），来跟踪相关对象随时间的变化。同时证件表用作对用户资质和经历的认证，单独放在一张表中。涉及到的所有维度（类型）都整合在一张编码表中。

数据字段的类型按照MYSQL数据库的存储方式给出。

# **数据库表清单**

2.1 双创平台集市功能表清单

|  |  |  |
| --- | --- | --- |
| **表名称** | **表中文名称** | **表说明(相关功能模块)** |
| product | 产品表 | 产品表是指集市模块功能中服务商提供的产品/服务信息的展示 |
| product\_category | 产品类别表 | 服务商提供的产品/服务所属哪一个类别 |
| supply\_detail | 供应详情表 | 服务商提供的服务或者产品详细的交易成功之后的信息 |
| collect | 收藏表 | 用来保存用户对于店铺或者产品(服务)收藏的信息 |
| demand | 需求表 | 需求表是用户发布的需求/服务的信息 |
| demand\_supplier | 需求服务商关联表 | 用户发布的需求和服务商之间的关系是一对多，用户的需求可以被多个服务商投标 |
| order | 订单表 | 用户提交需求后，可以让服务商看到。服务商可以选择投标。同时会生成一条订单信息 |
| payment\_record | 支付记录 | 订单完成之后，用户的支付记录 |
| after\_sale： | 售后记录 | 对订单的售后记录的维护和保存 |

2.2 双创平台店铺模块表清单

|  |  |  |
| --- | --- | --- |
| **表名称** | **表中文名称** | **表说明(相关功能模块)** |
| shop | 店铺表 | 示店铺信息 |
| message | 消息表 | 用户查看消息主页查看相应的消息信息，属于用户的消息 |
| shop\_info: | 店面消息表 | 店面消息表，支持用户查看收到有关商店交易交流的信息 |
| audit： | 审批表 | 删除/撤销申请需要系统审批，审批流程表 |
| Logistics: | 物流信息表 | 订单支付完成之后，用户发货，形成商品的物流信息 |

2.3 就业平台数据库表清单

|  |  |  |
| --- | --- | --- |
| **表名称** | **表中文名称** | **表说明(相关功能模块)** |
| job\_info | 招聘职位信息表 | 招聘单位发布的招聘职位信息 |
| job\_apply | 用户求职申请表 | 用户申请企业发布的职位信息生成的求职信息表 |
| contract | 合同表 | 用户求职成功后与用人单位签订的合同表 |
| message | 消息表 | 用户查相关的消息，招聘消息，订阅招聘消息以及站内通知信息等 |
| contract\_template | 合同模板表 | 用来显示合同模板相关的信息 显示在页面的固定信息 |
| job\_fair | 招聘会表 | 用来介绍一个招聘会的相关信息 |
| job\_fair\_audit | 招聘会审核表 | 招聘会审核表 一个招聘会对应多加企业，用来表示参加企业与招聘会之间的对应关系 |
| job\_fair\_company | 招聘会企业表 | 用来说明参与招聘会的企业与招聘会之间的对应关系。 |
| job\_type | 简历表 | 用户申请的岗位的类别。 |
| message\_type: | 消息类别维度表 | 用户发送的消息的类别 ; |
| apply\_status: | 求职状态维度表 | 求职状态类别表 |
| theme\_type | 招聘会主题维度表 | :求招聘会主题的具体类型。 |

2.4 用户模块表清单

|  |  |  |
| --- | --- | --- |
| **表名称** | **表中文名称** | **表说明(相关功能模块)** |
| **t\_user** | **用户表** | 描述用户相关属性。 |
| **t\_user\_role** | **用户角色表** | 描述用户角色的对应关系。 |
| **t\_role** | **角色表** | 描述角色相关属性。 |
| **t\_role\_permission** | **角色权限表** | 描述角色对应的权限设置。 |
| **t\_person\_info** | **个人基本信息表** | 描述个人相关的基本信息 |
| **t\_organization\_info** | **机构基本信息表** | 描述机构的基本信息 |
| **t\_certificate\_info** | **证件表** | 描述各种证件相关信息 |
| **t\_org\_state** | **企业状况表** | 描述企业的运营状况 |
| **t\_per\_experience** | **个人经历表** | 描述个人的历史经历信息 |
| **~~t\_academic\_dim~~** | **~~学历维度表~~** | ~~描述各种学历类别~~ |
| **~~t\_professional\_dim~~** | **~~职称维度表~~** | ~~描述各种职称类别~~ |
| **~~t\_sector\_dim~~** | **~~行业类型维度表~~** | ~~描述各种行业类型~~ |
| **~~t\_certificate\_dim~~** | **~~证件类型维度表~~** | ~~描述各种证件类型~~ |
| **~~t\_position\_dim~~** | **~~岗位类型维度表~~** | ~~描述各种岗位类型~~ |
| **t\_encode\_dim** | **编码表** | 描述各种编码类型及编码方式 |

# **主要表关联关系**

(待完善)

# **数据库对应表关系**

4.1 市场模块数据库设计

4.1.1供应表

product：供应表是指集市模块功能中服务商提供的产品/服务信息的展示

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 产品ID | product\_id | bigint(20) | 自增，数据库主键 |
| 产品类别 | category\_id | bigint | 产品所属哪一个类别 |
| 关键字 |  |  | 描述产品的特征或类型的词语 |
| 产品介绍 | product\_desc | varchar(100) | 产品介绍 |
| 服务商id | supplier\_id | bigint | 供应产品的服务商 |
| 产品价格 | product\_price | varchar(20) | 产品价格 |
| 产品成交量 | product\_sale | varchar(20) | 产品销量 |
| 产品图片 | product\_pic | varchar(20) | 产品图片 |
| 产品评分 | product\_evaluate | varchar(20) | 产品评价/评分 |
| 店铺id | Shop\_id | bigint | 产品所属的店铺 |
| 是否删除 | lgc\_del\_flag | bit | 该产品是否删除或者下架 |
| 是否推荐商品 | Promt\_flag | Bit | 该产品是否为展位主推商品 |
|  |  |  |  |

4.1.2产品类别表

product\_category服务商提供的产品/服务所属哪一个类别

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 产品类别ID | category\_id | bigint(20) | 自增 |
| 类别名称 | category\_name | varchar(20) | 产品类别的名称 |
| 类别介绍 | category\_desc | varchar(100) | 类别介绍 |
| 是否删除 | lgc\_del\_flag | bit | 该产品类别是否删除或者下架 |

产品表

product服务商/店铺提供的产品/服务基本信息.产品信息被用户和店家共同修改，几个状态变量：成交量/库存：在用户下单支付成功后，成交量加1.库存减1；在用户退货后，成交量减1，库存加1. 商品上架时，成交量为零。库存由店铺设置。

产品降价/升价时，现在的价格变为“商品原价”，修改的价格变成商品的“价格”。

如果商品价格下降，触发消息推送。查找商品收藏表，过滤那些收藏价格高于商品现在价格的用户，向他们推送消息。

商品库存从零变为非零时，要查找购物车中有这件商品的用户，向他们推送消息。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 商品id | product\_id | int（11） | 主键，自增 |
| 店铺ID | store\_id | int(11) | 关联店铺表 |
| 商品照片 | goods\_Photo | varchar(128) |  |
| 商品分类 | goods\_Category | varchar(128) | 本来应该关联分类表或编码表，暂时直接存储 |
| 商品名称 | goods\_Name | varchar(128) |  |
| 商品描述 | description | varchar(254) | 这里可能要指向另外一个页面，或者可以图文存储的文件 |
| 价格 | goods\_Price | int(10) | 商品现在的价格，或售价 |
| 商品原价 | org\_Price | int（11） | 商品标价 |
| 产地 | production | varchar(128) | 这里可能需要一个地区表 |
| 成交量 | amount | int（11） | 这个值可能会有多种形式，如月销量，年销量，累积销量等。可能需要另外一张统计表 |
| 包装盒价格 | package\_Price | int(10) |  |
| 库存 | goods\_Stock | int(10) |  |
| 供应商id | supplier\_id | int(11) | 商品的来源供应商 |

分类表

classify\_tree ：分类表，产品、服务、店铺的分类信息。显示三级的分类信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 编号 | id | int |  |
| 第一级 | root | varchar(32) | 所属模块 |
| 第二级 | col2 | varchar(32) | 大类 |
| 第三级 | col3 | varchar(32) | 细类 |
| 分类类型 |  |  |  |

分类属性表

classify\_attribute ：分类表中分类对应的不同属性信息。现在根据第一级的分类来列举属性。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 属性编号 | id | int |  |
| 分类编号 | class\_id | int | 属性所对应的分类编号 |
| 属性名称 | attriName | varchar(32) |  |
|  |  |  |  |

分类属性取值详情表

classify\_attribute\_data ：商品在分类表中每个分类对应的不同属性的具体取值信息。以列式存储。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 商品编号 | id | int | 带有附加属性的商品或店铺的编号 |
| 分类编号 | class\_id | int |  |
| 属性编号 | attri\_id | int |  |
| 属性取值 | attri\_value | varchar（128） |  |
| 类型 | typeName | varchar（32） | 所属的模块：商品，店铺，服务，就业 |

开户行信息表

bank店铺或公司关联的交易信息机构。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺ID | store\_id | int(11) | 关联店铺表/或机构 |
| 账户类型 | account\_Type | tinyint（4） |  |
| 开户名 | account\_Name | varchar(128) |  |
| 账号 | account\_Num | varchar(32) |  |
| 开户行名称 | account\_Bank | varchar(128) | 银行算是机构，要关联机构表中的信息 |
| 银行id | org\_id | int（11） |  |
| 开户支行名称 | account\_SubBank | varchar(128) |  |
| 支行id | suborg\_id | int(11) |  |

产品评论表

product\_comment 服务商/店铺提供的产品/服务评价信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 评论id | cmt\_id | int(11) |  |
| 商品id | product\_id | int（11） | 主键，自增 |
| 店铺ID | store\_id | int(11) | 关联店铺表 |
| 用户id | user\_id | int(11) |  |
| 用户昵称 | nick\_name | varchar(128) |  |
| 评论内容 | cmt\_content | varchar(254) |  |
| 评论图片 | cmt\_photo | varchar(128) |  |
| 评论时间 | cmt\_time | int（11） | 时间字符串，精确到天 |
| 评论类型 | cmt\_type | tinyint（4） | 1最差，5最好 |

4.1.3供应详情表

supply\_detail：服务商提供的服务或者产品详细的交易**成功**之后的信息 每次成功提供一次服务就会产生一次供应信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 详情id | detail\_id | bigint(20) | 自增 |
| 店铺id | shop\_id | bigint |  |
| 产品/服务id | product\_id | bigint | 可以查看产品详情 |
| 雇主评价 | evluate | varchar(50) | 雇主对该次服务的评价 |
| 成交金额 | money | varchar(20) | 该次服务/产品的成交金额 |
| 成交数量 | num | int | 该次服务/产品的成交数量 |

4.1.4收藏表

collect：用来保存用户对于店铺或者产品(服务)收藏的信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 收藏id | collect\_id | bigint(20) | 自增 |
| 收藏时间 | collect\_date | datetime |  |
| 收藏店铺 | collect\_shop | bigint |  |
| 收藏产品 | collect\_product | bigint |  |
| 是否删除 | lgc\_del\_flag | bit | 已删除，未删除 |
| 用户ID | user\_id | bigint | 与基本表中用户ID一致 |

## 4.1.5需求表

demand：需求表是用户发布的需求/服务的信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 需求id | demand\_id | bigint(20) | 自增 |
| 需求主题 | demand\_title | varchar(20) | 需求的主题 |
| 需求关键字 |  |  | 需求涉及的核心技术描述 |
| 需求类别 |  |  | 需求所属领域或范围的描述 |
| 需求内容 | demand\_content | varchar(100) | 发布需求的内容 |
| 需求发布时间 | demand\_publish\_time | datetime | 需求的发布时间 |
| 需求发布人 | user\_id | bigint | 需求的发布者 |
| 需求预算 | demand\_budget | Int | 需求的预期价格 |
| 是否删除 | lgc\_del\_flag | bit | 该需求是否删除 |

## 4.1.6需求服务商关联表

demand\_supplier:用户发布的需求和服务商之间的关系是一对多，用户的需求可以被多个服务商投标

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 需求id | demand\_id | bigint | 需求的主键 |
| 服务商id | supplier\_id | bigint | 投标改需求的服务商 |
| 投标时间 | bid\_time | datetime | 投标时间 |
| 投标备注 | bid\_content | varchar(50) | 投标描述 |
| 投标金额 | Bid\_money | varchar | 服务商对需求的报价金额 |

## 4.1.7订单表

order:用户提交需求后，可以让服务商看到。服务商可以选择投标。同时会生成一条订单信息在用户-“我的管理”模块 (店铺是否归服务商所有？)。用户可以删除已完成的订单记录，设置一个标志位。店铺不可以删除订单记录，作为报税等信息凭证。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 订单id | order\_id | bigint | 自增 |
| 用户id | user\_id | bigint | 提交需求的用户 |
| 服务商id | service\_id | bigint | 竞标需求的服务商 |
| 店铺id | shop\_id | bigint | 提供服务的店铺id |
| 下单时间 | prder\_time | datetime | 下单的时间 |
| 产品id | product\_id | bigint | 订单关联的产品？多个商品呢 |
| 需求id | demand\_id |  | 与该订单关联的需求 |
| 进行状态 | order\_status | int | 订单进行的状态（进行中，取消等） |
| 支付状态 | payment\_status | int | 订单的支付状态 成功、失败 |
| 成交数量 | num | int |  |
| 托管金额 | money | varchar | 订单的托管金额 |
| 取消原因 | fail\_reason | varchar |  |
| 取消时间 | fail\_time | datetime |  |
| 订单是否取消或者删除 | lgc\_del\_flag | bit |  |

## 集市订单表

market\_order：集市模块的订单信息。暂时用系统自动编号的ID充当订单ID，另外设置一个订单编号，作为未来的订单id。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 订单id | orderNum | int |  |
| 用户id | userID | int |  |
| 收货人 | addressName | varchar(128) |  |
| 收货人联系店电话 | addressPhone | varchar(32) |  |
| 收货地址 | addressDetail | varchar(128) |  |
| 商品所在商店id | storeId | int |  |
| 商品商店名字 | storeName | varchar(128) |  |
| 商品id | goodId | int |  |
| 商品描述 | goodDesc | varchar(254) |  |
| 商品单价 | price | float |  |
| 商品原单价 | originPrice | float |  |
| 购买的商品数量 | goodCount | int |  |
| 订单总价 | totalMoney | float |  |
| 快递方式 | deliveryType | tinyint |  |
| 快递价格 | deliveryPrice | int |  |
| 买家留言 | extraMessage | varchar(254) |  |
| 订单状态 | orderState | tinyint | 订单状态，类似于淘宝，有五种,用枚举类型表示：0-待付款，1-待发货，2-待收货，3-交易成功（待评价）4-交易成功（已评价/追加评价）对于用户付款提交订单付款成功的，订单进入到待发货状态，故一般提交订单，订单状态为1-待发货同时，客户端提交的订单数据只有这一些，后台还需要随机生成一个订 //单编号orderNum，后面获取订单数据需要这一个数据段 |
|  |  |  |  |

## 4.1.8支付记录

payment\_record：订单完成之后，用户的支付记录

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 支付id | payment\_id | bigint |  |
| 支付时间 | payment\_time | datetime |  |
| 订单id | order\_id | bigint |  |
| 支付金额 | payment\_sum | varchar(20) |  |
| 支付方式 | payment\_way | int | 支付宝 微信 银联 等 |
| 支付状态 | payment\_status | int | 成功、失败 |
| 支付记录是否删除 | lgc\_del\_flag | bit | 已删除，未删除 |

4.1.9售后记录

after\_sale:对订单的售后记录的维护和保存

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 售后id | id | bigint | 自增 |
| 订单id | orderId | bigint |  |
| 店铺Id | storeId | int |  |
| 店铺名称 | storeName | varchar |  |
| 商品id | goodId | int |  |
| 商品描述 | goodDesc | varchar |  |
| 申请退款总金额 | totalMoney | float |  |
| 运费 | deliveryPrice | float |  |
| 售后描述 | drawbackReason | varchar(100) |  |
| 售后时间 | submitTime | datetime | 提交售后申请的时间 |
| 申请售后类型 | drawbackType | tinyint | //有两种类型，枚举表示 0-仅退款， // 1-退款（含退货） |
| 申请类型状态" | drawbackState | tinyint | //两种，枚举表示 0-正在受理，1-退款成功 //用户刚刚提交申请时，处于0-正在受理状态 |
| 申请售后说明 | drawbackExtraMsg | varchar |  |
| 完成时间 |  | timestamp |  |
| 处理人员 |  | int |  |

4.2 店铺模块数据库设计

4.2.1店铺表

shop:表示店铺信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 | 编号 |
| 店铺id | shop\_id | int(11) | 自增 | 1 |
| 店铺联系人 | shop\_contact\_name | varchar(128) |  | 2 |
| 店铺联系人电话 | shop\_contact\_phone | varchar(32) |  | 3 |
| 店铺联系人Email | shop\_contact\_email | varchar(128) |  | 4 |
| 店铺联系人住址 | shop\_contact\_address | varchar(254) |  | 5 |
| 店铺名称 | shop\_name | varchar(254) |  | 6 |
| 店铺经营范围 | shop\_busi\_category | varchar(254) |  | 7 |
| 店铺有效期开始 | shop\_start\_date | int（11） |  | 8 |
| 店铺有效期截止 | shop\_expire\_date | int(11) |  | 9 |
| 店面关键字 | shop\_keyword | varchar(254) |  | 10 |
| 店面介绍 | shop\_introduction | varchar(254) |  | 11 |
| 店面logo | shop\_logo | varchar(128) | logo图片路径 | 12 |
| 店铺拥有者 | shop\_owner | varchar(32) | 店铺属于哪一个用户 | 13 |
| 店铺状态 | shop\_status | tinyint(4) | 开，关，暂停营业 | 14 |
| 店铺编号 | shop\_code | varchar(32) | 按照一定的规则对所有的店铺进行编号 | 15 |
| 是否删除 | del\_flag | tinyint(4) | 已删除，未删除 | 16 |
| 拥有者身份证 | owner\_attachment | Varchar(32) | 默认值为0，表示没有上传身份证；如已上传，值为身份证附件地址 | 17 |
| 是否有实体店 | Shop\_flag | tinyint(4) | 有实体店，无实体店 | 18 |
| 是否上传执照 | shop\_attachment | Varchar(128) | 默认值为0，表示没有实体店或未上传执照；如已上传实体店执照，值为营业执照附件地址 | 19 |
| 保证金 | Insure\_money | varchar(32) | 默认值为0，表示未交店铺保证金；如已交，存储所交金额，转为字符串格式 | 20 |
| 营业执照编号 | license\_num | varchar（32） |  | 21 |
| 许可证号 | permit\_num | varchar(32) |  | 22 |
| 个人身份证编号 | ID\_cardNo | varchar(32) | 身份证编号 |  |

配送方式表

delivery:商铺支持的物流配送方式。现在设想配送方式为4种（普通，自提，当日达，限时达），然后另建一张表保存和配送对应的订单的进度信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺id | shop\_id | int（11） | 自增 |
| 配送方式 | delivery\_type | tinyint（4） | 如果有记录则支持该种配送方式，1普通，2自提，3当日达，4限时达 |
| 配送方式编号 |  |  |  |
|  |  |  |  |

配送地址表

deliveryAddress:用户的送货地址。 一个用户可以有多个送货地址。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 用户id | user\_id | varchar(32) | 用户id |
| 配送地址 | delivery\_address | varchar(254) | 配送地址 |

配送信息表

delivery\_info:商铺物流的进度信息。可能存在的问题为，一个店铺所有的配送进度信息会混杂在一起，不方便检索。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺id | shop\_id | int（11） | 自增 |
| 配送方式 | delivery\_type | tinyint（4） | 1普通，2自提，3当日达，4限时达 |
| 配送进度信息 | message | varchar(254) |  |
| 订单id | order\_id | varchar（32） | 订单可能要区分不同类型如活动之类 |
|  |  |  |  |

店铺关注表

store\_attention记录用户关注的店铺id。这里尽量保留用户的历史记录。用于用户行为分析。主要是各种时间的记录。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺id | shop\_id | int（11） | 用户关注的店铺 |
| 用户id | user\_id | int(11) |  |
| 关注状态 | state | tinyint | 1，关注，2，取消关注 |
| 关注日期 | submitTime | datetime |  |

购物车表

shop\_cart记录用户在购物车中添加的商品和店铺

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺id | shop\_id | int（11） |  |
| 用户id | user\_id | int（11） |  |
| 商品id | product\_id | int(11) |  |
| 添加日期 | addTime | datetime |  |
| 支付状态 | payState | tinyint |  |
| 支付时间 | payTime | datetime |  |
| 订单编号 | orderID | bigint |  |

商品收藏表

goods\_collection 记录用户在收藏夹中收藏的商品

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺id | shop\_id | int（11） |  |
| 用户id | user\_id | int(11) |  |
| 商品id | product\_id | int(11) |  |

店铺商品表

store\_goods 记录店铺中的商品，及上架信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺id | shop\_id | int（11） |  |
| 商品id | product\_id | int(11) |  |
| 库存 | stock | int(11) |  |
| 销量 | sales | int(11) |  |
| 上架日期 | addTime | datetime |  |

推荐商品列表

goods\_recmd在主页上推荐的商品列表。属于平台的活动。区别于店铺的促销活动。这里的推荐用于在主页上显示（不同位置），检索时优先显示，

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店铺id | shop\_id | int（11） |  |
| 商品id | product\_id | int(11) |  |
| 推荐位置 | recmd\_loc | varchar(128) |  |
| 推荐级别 | recmd\_level | tinyint(2) |  |
| 推荐开始日期 | recmd\_start | datetime |  |
| 推荐结束日期 | recmd\_end | datetime |  |

4.2.2消息表

message:用户查看消息主页查看相应的消息信息，属于用户的消息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 消息id | message\_id | bigint | 自增 |
| 用户id | user\_id | bigint | 该消息是属于哪一个用户 |
| 消息内容 | message\_content | varchar(150) |  |
| 消息发送时间 | message\_send\_time | datetime |  |
| 消息类型 | message\_type | int | 供应类消息、需求类消息、对话消息 |
| 是否回复 | message\_isReplied | bit | 已回复、未回复 |
| 回复内容 | message\_reply\_content | varchar（150） | 消息的回复内容 |
| 是否阅读 | message\_isRead | bit | 已读、未读 |
| 发送者 | user\_id | bigint | 谁发送的消息 |
| 消息回复时间 | message\_reply\_time | datetime |  |
| 是否删除 | lgc\_del\_flag | bit | 已删除，未删除 |

4.2.3店面消息表

shop\_info:店面消息表，支持用户查看收到有关商店交易交流的信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 店面消息id | Shop\_info\_id | bigint | 自增 |
| 店面id | Shop\_id | bigint |  |
| 用户id | User\_id | bigint |  |
| 消息内容 | Shop\_info\_content | varchar |  |
| 消息类型 | Shop\_info\_type | int | 订单消息、招标消息、售后消息、邮件消息 |
| 消息时间 | Shop\_info\_time | datetime | 消息发送时间 |
| 是否已读 | Shop\_info\_isRead | bit |  |
| 消息发送者 | Shop\_info\_sender | bigint |  |
| 是否删除 | lgc\_del\_flag | bit | 已删除，未删除 |

4.2.4审批表

audit：删除/撤销申请需要系统审批，审批流程表。对于合同签订可能需要机构或单位中的多级部门审批

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 审批id | audit\_id | bigint | 自增 |
| 审批时间 | audit\_time | datetime |  |
| 审批人员 | audit\_person | bigint | 此次申请由谁审批 |
| 审批结果 | audit\_result | int | 批准、不批准 |
| 审批理由 | audit\_reason | varchar |  |
| 用户id | user\_id | bigint | 该店铺的审批与哪个用户有关，用户申请 |
| 店铺id | shop\_id | bigint |  |
| 是否删除 | lgc\_del\_flag | bit | 已删除，未删除 |

4.2.5物流信息表

logistics:订单支付完成之后，用户发货，形成商品的物流信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 物流表id | logistics\_id | bigint | 自增 |
| 物流时间 | logistics\_time | datetime |  |
| 物流介绍 | logistics\_desc | varchar |  |
| 物流名称 | logistics\_name | varchar |  |
| 物流单号 | logistics\_num | varchar |  |
| 物流价格 | logistics\_price | double |  |
| 物流状态 | logistics\_status | int | 确认收货，完成，取消(转售后) |
| 订单id | order\_id | bigint | 该物流关联的订单 |
| 是否删除 | lgc\_del\_flag | bit | 已删除，未删除 |

4.3 就业平台数据库设计

4.3.1 招聘岗位信息表

job\_info：招聘单位发布的招聘岗位信息

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 岗位id | job\_info\_id | varchar(32) | 职位的主键，自增字段，数据库存储的标记 |
| 岗位编码 | job\_code | varchar(32) | 岗位的编码信息，每个求职岗位都有一个，按照一定格式编码，方便企业的统计 |
| 营业执照号 | license\_no | varchar(32) | 与企业营业执照编号一致 |
| 岗位名称 | job\_name | varchar(128) | 岗位的具体名称 |
| 岗位父类编码 | job \_type\_pcode | varchar(32) | 与岗位分类表的编码一致 |
| 岗位子类编码 | job \_type\_scode | varchar(32) | 岗位的二级分类信息 |
| 发布时间 | publish\_time | int(11) | 岗位的发布时间 |
| 截止时间 | end\_time | int(11) | 岗位的截止时间 |
| 岗位状态 | job\_status | tinyint(4) | 停止/进行 |
| 需求人数 | number | int(8) | 岗位需要的人员 |
| 岗位关键字 | job\_keyword | varchar(128) | 该岗位的关键字，用于描述岗位的关键信息 |
| 专业类别父类 | major\_type\_pid | varchar(32) | 按照国家标准专业分类父类专业 |
| 专业类别子类 | major\_type\_sid | varchar(32) | 按照国家标准专业分类子类专业 |
| 出勤形式 | attendance\_type | varchar(32) | 全职/兼职/学生实习 |
| 待遇 | payment | varchar(32) | 岗位的薪资待遇 元/天 ,元/月 |
| 岗位职责 | job\_duty | varchar(254) | 岗位的职责描述 |
| 工作省份 | Job\_province | Varchar(20) |  |
| 工作城市 | Job\_city | Varchar(20) |  |
| 工作地点 | job\_location | varcha(128)r | 岗位的工作具地点介绍 |
| 岗位要求 | job\_require | varchar(254) | 应聘改岗位的基本要求。身高/性别/实习时间/学历/职称 |

4.3.2 用户求职申请表

job\_apply:用户申请企业发布的岗位信息生成的求职申请表。用户申请一个岗位，产生一条申请信息，对应的申请存储在申请表中。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 求职id | job\_apply\_id | bigint(20) | 数据库主键，自增，数据库存储 |
| 岗位编码 | job\_code | varchar(32) | 与job\_info表中，职位编码一致，对应唯一的岗位信息 |
| 申请编号 | job\_apply\_code | varchar(32) | 求职编号，用户申请职位后，对于此次申请有一个对应的编码，该编码是按照一定规律生成的唯一字段 |
| 身份证号 | id\_cardNo | varchar(32) | 申请该职位的用户的身份证编号 |
| 求职状态 | apply\_status\_code | varchar(32) | 已批准的但未完成合同的申请/已批准的且完成合同的申请/被拒绝的申请/被拒绝的 |
| 求职时间 | apply\_time | int(11) | 申请该岗位的时间 |
| 求职介绍 | apply\_desc | varchar(254) | 对于自己申请该职位的原因的介绍 |
| 求职附件 | apply\_attachment | varchar(128) | 求职的证书等相关的证书附件作为求职的证明 |
| 求职要求 | apply\_demand | varchar(254) | 完善申请表，自己对于该岗位关于薪资，上班时间，以及其他条件的一些要求，属于用户自己填写的，使用分号隔开进行存储 |
| 营业执照号 | license\_no | varchar(32) | 与企业的主键一致，提供该职位的企业 |
| 用户id | user\_id | varchar(32) | 登录系统的用户的登录账号，用户利用该账号申请相应的岗位 |
| 是否删除 | is\_delete | tinyint(1) | 该条求职申请是否删除，删除/未删除 |

4.3.3 合同表

contract:用户求职成功后与用人单位签订的合同信息对应的数据库表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 合同id | contract\_id | bigint(20) | 数据的主键，自增 |
| 合同编号 | contract\_code | varchar(32) | 每一份用户和企业签订的合同对应的编号 |
| 营业执照号 | license\_no | varchar(32) | 对应签订合同的企业的信息 |
| 身份证号码 | Id\_cardno | varchar(32) | 签订用户的身份证的编号 |
| 岗位编码 | job\_code | varhchar(32) | 该求职者申请成功的职位信息 |
| 薪资水平 | payment | varchar(32) | 签订合同达成的薪资待遇水平 |
| 签订时间 | contract\_time | int(11) | 签订合同的时间 |
| 合同生效时间 | contract\_start\_time | int(11) | 合同中约定的服务时间范围，即是用户的入职时间。 |
| 合同结束时间 | contract\_end\_time | int(11) | 合同的到期时间 |
| 合同更新次数 | update\_time | int(4) | 合同续签后会更新合同的结束时间，并对更新次数加一 |
| 试用期开始时间 | probation\_start | int(11) | 签订合同的入职时间，也是试用期的时间 |
| 试用期结束时间 | probation\_end | int(11) | 试用期的结束时间 |
| 合同状态 | contract\_status | tinyint(4) | 待签订/签订中/已完成 |
| 合同附件 | contract\_attachment | varchar(128) | 合同相关的附件信息，证书，身份证等证件合成一个文件,对应一个存储路径 |
| 模板编号 | contract\_modelno | varchar(32) | 对应的合同的模板编号，将用户填写的合同信息，填写到合同模板中去。 |
| 实习开始时间 | Practice\_begin | date |  |
| 实习结束时间 | Practice\_end | date |  |

4.3.4 消息表

message:用户查看相关的消息，实现对消息的存储。

用户发送和接收的消息分为不同的类别（常见的有系统消息，订阅消息，认证消息，好友消息），用户收到消息后，并不一定立即存储，如果是一般的信息，此时的消息是以xml的格式临时存储的，在达到一定的天数后，该消息会被自动清除；如果用户需要对消息进行保存，可以将临时消息，存储到数据库中，后面可以方便用户的查看。消息的临时xml的格式设计如下:



|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 消息id | message\_id | bigint | 自增 |
| 用户id | user\_id | varchar | 该消息是属于哪一个用户，接收消息的用户。求职者用身份证号或者企业营业执照号 |
| 消息内容 | message\_content | varchar(150) | 发送的消息内容，消息的内容有多种。如果传送的是文字类别的消息，则直接存储到数据库，如果对应的是图片/音频/视频或者其他媒体消息，不能简单用字符消息存储的，则存储对应的路径，需要的时候，根据该路径去获取即可。将发送的消息内容获取，存储到本地路径下。 |
| 消息发送时间 | message\_send\_time | datetime | 消息的发送时间 |
| 消息类别 | message\_type | int | 消息类别父类分为订阅消息，企业消息认证消息，系统消息  消息子类要更加具体化，订阅消息的类别又分为哪些消息，详细可见消息类别纬度表 |
| 是否回复 | message\_isreplied | bit | 已回复、未回复 |
| 回复内容 | message\_reply\_content | varchar | 消息的回复内容 |
| 是否阅读 | message\_isread | bit | 已读、未读 |
| 消息发送者 | user\_id | bigint | 谁发送的消息一般是企业或者用户 |
| 消息回复时间 | message\_reply\_time | datetime | 对话消息，是否回复 |
| 是否删除 | lgc\_del\_flag | bit | 已删除，未删除 |

4.3.5合同模板表

contract\_template:用来显示合同模板相关的信息 显示在页面的固定的合同信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 模板编号 | template\_no | varchar(32) | 可能存在多个合同(劳动合同，试用期劳动合同等)，每个不同的合同都有一个合同的模板编号对应不同的模板 |
| 公司名称/甲方 | org\_id | varchar(32) | 甲方联系人，甲方类受托方和委托方 经济关系 甲乙要介绍清楚型 |
| 甲方联系人 | legal\_rep | varchar(32) | 甲方联系人姓名 |
| 公司营业执照号 | license\_no | varchar(32) |  |
| 公司地址 | address | varchar(128) |  |
| 签订时间 | contract\_time | int(11) | 时间字段都采用int类型，unix时间戳来进行存储。 |
| 试用期开始时间 | probation\_start | int(11) |  |
| 试用期结束时间 | probation\_end | int(11) |  |
| 合同有效期 | expire\_date | int(11) |  |
| 5险一金约定比例 | insure\_ratio | varchar(32) |  |
| 应聘者/乙方 | user\_id | varchar(32) |  |
| 联系电话 | phone | varchar(16) |  |
| 姓名 | name | varchar(32) |  |
| 身份证号 | id\_cardno | varchar(32) |  |
| 性别 | sex | tinyint(4) |  |
| 通讯地址 | address | varchar(128) |  |
| 合同条款 | contract\_term | longtext | 合同条款，固定的一些合同的条款的信息 |
| 合同正文 | contract\_content | longtext | 合同的正文，可能不同的合同的正文不一样。 |
| 合同类型 | contract\_type | varchar(32) | 合同的具体类型 |
| 合同名 | contract\_name | varchar(128) | 合同名称 |
| 甲方签名 | partA\_sig | varchar(254) |  |
| 甲方签订日期 | partA\_date | int(11) |  |
| 乙方签名 | partB\_sig | varchar(254) |  |
| 乙方签订日期 | partB\_date | int(11) |  |
| 薪资水平 | Payment | Varchar(10) |  |
| 实习开始时间 | Practice\_begin | Varchar(5) |  |
| 实习结束时间 | Practice\_end | Varchar(5) |  |

4.3.6招聘会表

job\_fair:用来描述一次招聘会的相关信息。每次招聘会都对应着一张数据库表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 招聘会id | job\_fair\_id | bigint(20) | 主键，自增 |
| 招聘会编码 | job\_fair\_code | varchar(32) | 每次招聘会都有一个编码信息，编码是唯一的，按照一定的规则生成。 |
| 招聘会时间 | job\_fair\_time | int(11) | 招聘会举办的时间 |
| 招聘会介绍 | job\_fair\_desc | varchar(254) | 招聘会具体信息的介绍，简单的介绍 |
| 招聘会关键字 | job\_fair\_keyword | varchar(128) | 招聘会的关键字信息，方便进行查找和搜索 |
| 招聘会主题编码 | job\_fair\_theme | varchar(128) | 用来说明招聘会的主题是综合/机械/IT/等具体类型的 |
| 招聘会文件 | job\_fair\_file | varchar(128) | 对应一个招聘会的详细文件，统一采用pdf格式的形式，具体描述此次招聘会的信息,包括参与招聘的具体企业的名单，企业招聘岗位的列表，以及招聘具体的人数以及对于企业信息的简单介绍。 |
| 招聘会主办方 | job\_fair\_host | varchar(32) | 招聘会的主办方 |
| 招聘会地点 | job\_fair\_place | varchar(128) | 招聘会举办地点 |
| 招聘企业数 | job\_fair\_companys | int(4) | 招聘会参与总的的企业数 |
| 提供岗位数 | job\_fair\_posts | int(8) | 招聘会提供的总的岗位数 |
| 招聘会状态 | job\_fair\_status | tinyint(4) | 已举办/正在进行 |
| 招聘会名字 | job\_fair\_name | varchar(128) |  |
| 招聘会所在城市 | Job\_fair\_city | varchar(32) |  |

4.3.7招聘会审核表

job\_fair\_audit:招聘会审核表 。主办方举办招聘会，需要提交申请，提交相应的信息让系统管理员进行审核，审核成功之后；招聘会可以在规定的时间举办。如果不成功给出理由。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 主键 | audit\_id | bigint(20) | 主键，数据库自增字段 |
| 招聘会编码 | job\_fair\_code | varchar(32) | 参与审核的招聘会的信息 |
| 申请说明 | apply\_info | varchar(254) | 申请的相关说明 |
| 申请时间 | apply\_time | int(11) | 申请招聘会的时间 |
| 申请人 | apply\_person | varchar(32) | 举办此次招聘会的单位或着主板方 |
| 审核状态 | audit\_status | tinyint(4) | 待审核/审核成功/审核成功 |
| 审核时间 | audit\_time | int(11) | 管理员审核招聘会的时间 |
| 审核原因 | audit\_reason | varchar(254) | 审核成功或者失败都需要给出相应的说明 |
| 招聘会文件 | job\_fair\_file | varchar(128) | 上传申请举办此次招聘会的计划书，提交给管理员审核。对应着一个文件的存储url. |

4.3.8 招聘会企业表

job\_fair\_company:用来说明参与招聘会的企业与招聘会之间的对应关系。

每次招聘会对应多家参展的企业，是一个一对多的关系，因此将企业招聘会的编码(每次招聘会对应一个唯一编码)，放入招聘会企业表中作为外键，建立一对多的关系。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 主键 | job\_fair\_companyId | bigint(20) | 主键，数据库自增字段 |
| 展位号 | stand\_number | varchar(32) | 参与企业的招聘会的编号 |
| 营业执照号 | license\_no | varchar(32) | 招聘会企业的营业执照号,可以拿到企业的相关信息 |
| 岗位编码 | job\_code | varchar(32) | 招聘岗位的编码信息，对应着该岗位的信息。 |
| 招聘人数 | recuit\_num | int(8) | 对应岗位招聘的人数 |
| 招聘会编码 | job\_fair\_code | varchar(32) | 此次参展企业属于哪一次招聘会 |
| 备注 | remark | varchar(254) | 对于招聘会企业的简单描述或者备注 |

4.3.9 岗位分类维度表

job\_type:用户申请的岗位的类别。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 主键 | message\_type\_id | bigint | 消息类别的主键自增 |
| 岗位编码父类 | job\_type\_pcode | varchar | 对于每个类型的岗位进行编码，可以按照默认的指定格式来进行编码 |
| 岗位编码子类 | job \_type\_scode | varchar | 如果岗位没有父级类别，岗位父类可以置为空。 |
| 备注 | remark | varchar | 简单介绍该类别信息 |
| 岗位类别名称 | job \_type\_name | varchar | 该类岗位的名字 |

4.3.10 消息类别维度表

message\_type:用户发送的消息的类别 ;

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 主键 | message\_type\_id | bigint | 消息类别的主键自增 |
| 消息编码父类 | message\_type\_pcode | varchar | 对于每个类型的消息进行编码，可以按照默认的格式来进行编码 |
| 消息编码子类 | message\_type\_scode | varchar | 如果消息没有父级类别，消息可以置为空。 |
| 备注 | remark | varchar | 简单介绍该类别 |
| 消息类别名称 | message\_type\_name | varchar | 该类消息的名字 |

4.3.11求职状态维度表

apply\_status:求职状态类别表。用户申请岗位后，用户的求职状态，会经过待处理，已处理，待签合同，签订合同等状态的变化，此表用来记录每一种状态的变化

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 求职状态id | apply\_status\_id | bigint | 求职状态的主键，自增 |
| 求职状态编码 | apply\_status\_code | varchar | 对求职状态进行编码 |
| 求职状态名称 | apply\_status\_name | varchar | 求职状态编码对应的求职状态的描述。  未受理/处理中/已经批准/历史 |
| 求职状态备注 | apply\_status\_remark | varchar | 关于该状态的解释和备注 |

4.3.12招聘会主题维度表

theme\_type:求招聘会主题的具体类型。用来说明招聘会是哪一种类型的。比如综合，IT，机械电子，化工等

|  |  |  |  |
| --- | --- | --- | --- |
| 字段中文名 | 字段英文名 | 字段类型 | 备注 |
| 招聘主题id | theme\_type | bigint | 主键，自增 |
| 主题编码 | theme\_code | varchar | 招聘会主题类型按照一定的规则来进行编码 |
| 说明 | theme\_desc | varchar | 关于该主题的说明与介绍 |

4.4 用户模块

介绍：用户模块主要的功能是用户登录及权限管理。权限管理使用 用户-角色-授权-权限 模型。一个用户可以有多个身份，用户可以将自己不同身份的账号授权给另外一个用户使用。同时一个用户可以有多个角色，其中管理员角色可以向下继续创建角色。用户通过具有的角色获取相应的权限。

4.4.1系统注册信息表

**表名：user\_info 系统注册信息表**

**添加账号根据一定的规则自动生成账号 、用户类别 普通客户**

**用途：** 描述系统用户信息，一个用户可以有多个角色，用户与角色的对应关系在用户角色表中定义。本表用于用户登录信息及交易时各种相关信息验证。用于判断用户的合法性。用户密码使用散列存储。这个系统一个，描述登录系统的用户的注册信息，记录结构为（用户账号，昵称，创建时间，有效期天数，电话，EMAIL，…，）  
注意，下面的用户ID，如果没有特色指出，是指用户账户名，即用户账号。

身份：帐户共用，即一个账户对应多种不同身份，例如，账户a可以是店主，也可以是顾客，也可以是专家，也可以是学生。为了支持该功能，需要增设身份表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **用户id** | uid | varchar(32) | 主键，可以自增，系统内唯一的用户id |
| **用户名** | user\_name | varchar(60) | 唯一的用户名，可以使用手机号/邮箱号作为用户名，用作登录 |
| **昵称** | nick\_name | varchar(60) | 用户的别名，可以自己自由设置 |
| **用户类型** | user\_type | varchar(32) | 用户类型目前分为个人和机构两种 |
| **用户密码** | pass | varchar(128) | 用户密码（哈希值） |
| **有效期天数** | valid\_day | int(11) | 用户账户的有效时间 |
| **授权权利** | auth\_right | tinyint(1) | 逻辑值，1表示该用户可以将自己的权限选择授予其他用户 |
| **身份表** | identity | varchar(32) | 用户可以具有多种身份，该段字段链接描述用户所就要有身份的表 |
| **用户邮箱** | mail | varchar(254) | 用户联系邮箱（站内） |
| **界面模板** | theme | varchar(255) | 用户使用的控制界面的模板，可以改变用户界面的色系等参数 |
| **创建时间** | created | int(11) | 用户创建日期时间戳 |
| **上次登录时间** | access | int(11) | 用户上次登录时间戳 |
| **登录时间** | login | int(11) | 用户的登录时间戳 |
| **状态** | status | tinyint(4) | 用户状态（1，可用；0，禁用）） |
| **时区** | timezone | varchar(32) | 用户所在的时区 |
| **语言** | language | varchar(32) | 用户所使用的语言 |
| **照片** | picture | int(11) | 外键，关联到文件表中用户照片文件 |
| **密保问题** | pass\_ques | blob | 用于用户找回密码时设置的问题 |
| **备用邮箱** | backup\_mail | varchar(254) | 用户创建账户时创建的邮箱 |

用户钱包表

**表名：user\_wallet** 记录用户在系统中的钱包金额。暂时假定里面只有一种币种。

|  |  |  |  |
| --- | --- | --- | --- |
| **文名** | **字段英文名** | **字段类型** | **备注** |
| **用户id** | uid | varchar(32) | 主键，可以自增，系统内唯一的用户id |
| **钱包余额** | walletMoney | float |  |

4.4.2用户角色表

**表名：user\_role 用途：**描述用户和角色的对应关系。一个用户可以对应多个角色。

整个系统一个，定义用户所属的角色，记录结构为（用户ID，角色ID，时限）。对于刚长注册的账户，系统按照默认的角色生成一条用户角色记录

注意：同一用户帐户，不同的身份可以对应不同的角色

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **用户ID** | uid | varchar(32) | 主键，用户id |
| **用户身份ID** | identity | varchar(32) | 支持指出是授予该用户的哪一个身份的角色 |
| **角色ID** | rid | varchar(32) | 主键，角色id |
| **时限** | expire\_date | int(11) | 用户所具有角色的有效期 |

4.4.3角色表

**表名：role 用途：**整个系统一个，描述系统权限组合。一个角色可以包含多种权限。一个角色可以赋予多个不同的用户。角色表的记录结构为（子系统ID，角色ID，角色名称,），例子：  
S01 A01 普通用户  
S01 A02 管理员 （这个角色可以向下继续创建角色）

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **角色ID** | rid | varchar(32) | 主键，角色id |
| **子系统id** | subsystemid | varchar | 角色相关的子系统编号 |
| **角色名称** | role\_name | varchar(60) | 角色名称，唯一 |
| **角色类型** | role\_type | role\_type | 管理员，可以创建子管理员；团队队长，可以分配任务 |

4.4.4 授权表

**表名：auth 用途：**一个账号，可以将自己的部分访问控制权限授予另外一个已经注册的账号，使该账户可以代替授权账号操作授权账号的模块，比如，a授权给b, 则b 可以用b 的账号登录a的账号（a 的系统模块）。整个系统一个表。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **授权者id** | author | varchar(32) | 授权者的账号id |
| **被授权者id** | acceptor | varchar(32) | 被授权者的账号id |
| **授予角色** | role\_id | varchar(32) | 授权者a可以将自己某一身份对应的角色授权给被授权者b |
| **授予时限** | expire\_date | int(11) | 授权的有效时间 |
| **授权生效时间** | effective\_date | int(11) | 授权开始的时间 |

4.4.5身份表

**表名：identity 用途：**整个系统一个，也就是系统中，唯一确定账号的是账号加身份。身份值是枚举值，由编码表定义，对双创系统，值为店主，顾客，专家，学生，公司，…

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **身份ID** | identity\_id | varchar(32) | 用户的身份id |
| **身份名称** | identity\_name | varchar(60) |  |
|  |  |  |  |
|  |  |  |  |

例子：

U01 店主

U01 客户

U02 教务管理员

该表说明，账户U01具有两个身份，而U02只有一个身份

4.4.7角色权限表

**表名：role\_permission 用途：**描述角色和权限的对应关系，定义角色的权限。。权限字段定义了对模块的操作，一般为增，删，改，查看。存储内容为（1，查看就业平台消息，就业平台模块，NULL）表示角色id为1的用户具有查看就业平台消息的权限，该权限没有附加参数。（1，发布招聘消息，就业平台模块，{num：5}）表示角色id的用户可以在就业平台模块下发布5条消息。

整个系统一个，定义角色的权限，记录结构为（子系统ID,模块ID，角色ID,操作）。这里，模块ID是系统中可以有代码识别的代码模块的名称，例如是页面文件名。操作是枚举值，值包括只读、读写、删除、全权、无权等。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **角色ID** | rid | varchar(32) | 主键，角色id |
| **子系统id** | subsystem | varchar(32) | 权限所属的子系统 |
| **模块id** | module | varchar(32) | 权限所属的模块 |
| **操作** | permission | tinyint(4) | 角色可进行的操作名称 |

例子：

A01 S01 M02 只读

A01 S01 M01 读写

A02 S01 M01 查看

A02 S02 M01 读写

这里，定义了两个角色A01和A 02的权限。注意，一个角色的权限定义，一般需要多行。表以角色为关键字排序

4.4.8个人基本信息表

**表名：person\_info 用途：**描述个人相关的基本信息。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **用户ID** | user\_id | varchar(32) | 用户id用于唯一的识别一个用户，同时作为其它和用户相关信息表关联的主键。 |
| **身份证号码** | ID\_cardNo | varchar(32) |  |
| **姓名** | name | varchar(128) |  |
| **手机** | phone | varchar(32) |  |
| **email** | email | varchar(128) |  |
| **性别** | sex | tinyint(1) |  |
| **出生日期** | birth | int(11) |  |
| **家庭地址** | address | varchar(254) | 用户的家庭地址，地址可能有多个 |
| **最高学历** | degree | varchar(32) | 用户可能有多个学历 |
| **最高职称** | prof\_title | varchar(32) | 可能有多个值 |
| **照片** | photo | int(11) |  |
| **照片更新时间** | photo\_updateTime | int(11) |  |
| **项目经验** | projectExp | varchar（254） | 可以看做简单的自我介绍 |
| **护照号** | passport | varchar(32) | 可能有多个值 |

4.4.9机构基本信息表

**表名：organization\_info**

**用途：**描述机构的工商注册等基本信息。机构进一步可以分为公司，企业，事业单位，政府机构，以及社会组织，个体工商户等。这些统一保存在一张机构表中。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **机构代码证号** | org\_id | varchar(32) |  |
| **机构类型** | org\_type | varchar(32) | 结构类型为政府机构、公司（合资/独资）这些 |
| **名称** | org\_name | varchar(128) |  |
| **法人代表** | legal\_rep | varchar(32) |  |
| **注册地** | reg\_address | varchar(254) |  |
| **所在地** | site | varchar(254) | 公司总部所在地址 |
| **机构层级id** | hierarchy\_id | varchar(32) | 用于处理总公司和分公司等情况，同一层级上的机构为统一编号 |
| **营业范围** | business\_scope | varchar(254) |  |
| **注册资金** | reg\_capital | varchar(128) |  |
| **公司资质称号** | Title | Varchar(32) |  |
| **招聘联系人** | Contact | Varchar(32) |  |
| **联系人电话** | Contact\_phone | Varchar(32) |  |
| **联系人邮件** | Contact\_email | Varchar(32) |  |
| **营业执照号** |  |  |  |
| **税务登记证号** |  |  |  |
| **社会保险登记证号** |  |  |  |
| **统计登记证号** |  |  |  |
| **登记机关** | Reg\_auth | Varchar(64) |  |
| **宗旨** | Aim | Varchar(255) |  |
| **举办单位** | Holder | Varchar(64) |  |
| **经费来源** | Fund\_src | Varchar(64) |  |
|  |  |  |  |

机构部门表

**表名：org\_department**

**用途：**描述机构中的部门划分。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **机构编码** | org\_id | varchar(32) |  |
| **部门编号** | dprtID | int（11） |  |
| **部门名称** | dprtName | varchar(32) |  |
| **负责人** | principal | varchar(32) |  |
| **负责人编号** | uid | int（11） |  |
| **联系方式** | contact | varchar(32) |  |
| **部门职责** | duty | varchar(128) |  |

4.4.10证件表

**表名：certificate\_info**

**用途：**描述各种证件相关信息。系统中涉及的所有证件类的认证资料都在这张表中。现在企业推行5证合一。行业的分类见《国家行业分类标准》。五证是：营业执照，税务登记证，组织机构代码证，统计登记证，社会保险登记证。和食品相关的还有 食品流通许可证，食品卫生许可证。危险行业相关的有安全局颁发的各种证。娱乐行业要有文化局颁发的文化经营许可证。个人信息中有一类很大的证件信息要不要录入？准考证，只有临时性的用途，但量可能很大。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **证件编号** | cert\_id | varchar(32) | 证件由发布机构签发的编号，可以用作真伪查询等用途 |
| **证件名称** | cert\_name | varchar(128) |  |
| **认证机构编号** | auth\_org | varchar(32) | 认证机构为机构表中的机构 |
| **证件有效期** | cert\_expire | int(11) |  |
| **发证时间** | cert\_date | int(11) |  |
| **证件状态** | cert\_status | tinyint(4) | 证件状态为有效，无效，申请中，等有限值。 |
| **证件扫描件路径** | cert\_path | varchar(128) | 用于指向证件在本地保存的源文件 |
| **证件注册地址** | cert\_address | varchar(128) | 证件上的住所字段 |
| **证件类型** | cert\_type | tinyint(2) | 证件类型，表明证件的来源或用途，1，学习，2，工作，3，英语水平，4，计算机水平… |
| **用户编号** | user\_id | varchar(32) | 和证件相关的用户  或店铺或机构编号 |
| **企业执照号** | License\_no | varchar(32) |  |

4.4.11企业状况表

**表名：org\_state**

**用途：**描述企业的运营状况。企业状况用于评估企业的可信度及资质相关审核。涉及到变化的信息一般为按年度记录。不同类型的机构，可能无法获取所有的字段。比如事业单位可能没有营业额。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **企业id** | org\_id | varchar(32) | 对应于机构id |
| **营业执照号** | License\_no | varchar(32) |  |
| **年度** | year | int(11) | 信息统计的年度 |
| **员工数** | employee\_num | int(11) |  |
| **技术人员数量** | techStaff\_num | int(11) |  |
| **营业额** | revenue | varchar(128) |  |
| **资产总额** | total\_assets | varchar(128) |  |
| **负债总额** | total\_indebt | varchar(128) |  |
| **净销售额** | net\_sales | varchar(128) |  |
| **流动资产** | current\_assets | varchar(128) |  |
| **流动负债** | current\_indebt | varchar(128) |  |
| **营业费用** | operate\_expense | varchar(128) |  |
| **利润** | profit | varchar(128) |  |
| **贷款余额** | loan\_balance | varchar(128) |  |
| **纳税** | taxation | varchar(128) |  |
| **研发经费金额** | RD\_budget | varchar(128)v |  |
|  |  |  |  |

4.4.14个人经历表

**表名：per\_experience**

**用途：**描述个人的历史经历信息。主要从学生阶段开始到工作阶段。可以覆盖无业人员。

经历主要有学习经历，工作经历，实习经历，待业/无业也算作经历。一个用户可能会有多条经历记录。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段中文名** | **字段英文名** | **字段类型** | **备注** |
| **用户id** | uid | varchar(32) | 用户在系统中的标识字符串 |
| **证明材料编号** | cert\_id | varchar(32) | 理论上每段经历有1~2个证明材料（如学位证书，学历证书）。工作的证明材料为合同或纳税明细 |
| **身份证** | ID\_cardNo | varchar(32) |  |
| **年度** | year | int(11) |  |
| **活动类型** | experience\_type | tinyint（2） | 这里活动类型为经历类型：1学习/2工作/3实习 |
| **机构** | org\_id | varchar(32) | 对应于学习经历中的学校 |
| **部门** | depart | varchar(32) | 对应于学院 |
| **岗位级别** | position\_level | varchar(32) | 在公司中如果有从事过多个岗位，这里的岗位为最后一个所从事的岗位 |
| **岗位名称** | position\_name | varchar(128) | 对应于专业 |
| **准考证号** | admissionTicketNo | bigint(11) | 准考证号 |
| **学号** | SchoolNo | int（11） | 学号，教工的教工号也是这个字段 |
| **学制** | duration | tinyint（2） | 学制，最长128年 |
| **证明人id** | refer\_id | varchar(32) |  |
| **证明人电话** | refer\_phone | varchar(16) |  |
| **证明人email** | refer\_email | varchar(128) |  |
| **起始时间** | start\_date | int(11) | 两个时间戳 |
| **结束时间** | end\_date | int(11) |  |
| **是否在网上报名** | isNetReport | tinyint（2） |  |
| **是否确认过信息** | isCheck | tinyint（2） |  |
| **最后修改时间** | last\_modify | int（11） |  |

4.5 会话模块数据库设计

4.5.1会话表

conversation:会话表。

会话指的是在系统中参与及时通信的用户产生的信息，整个系统只有一份会话表，用于记录所有用户的会话信息，每个会话用户对应会话表里的一条记录，记录了该用户相关的所有会话信息，包括用户的会话记录、朋友关系（指向一个朋友关系表）、所在群（指向一个群组表）、私人收藏文件（指向一个私人收藏文件）、博报文件（指向一个博报文件）、微报文件（指向一个微报文件）、活跃度、诚信度等。

用户登录后，在参与系统中及时通信时，关于好友用户关系、所属群关系、好友会话记录、用户收藏、状态等都有可能发生变化，用户可以关注其他用户，也可能被其他用户关注，或者相互关注；用户可以创建群，也可以加入群；用户对感兴趣的文件可以进行收藏，作为私人收藏文件；用户个人在自己的会话朋友内或者网络空间内可以发表多媒体信息，称为“报示”，其中可以被有网络空间的任意用户搜索到的称为“博报”，只能由指定用户集阅读的称为“微报”；这些属性都会被记录在会话表中，用来描述每个用户的会话相关属性。

每个用户对应一个独立的朋友关系表、群组表、会话/微报/博报/收藏文件。会话表中对应的字段都是指向该文件或者表的名称，而这些表/文件都用表名/文件名指称。命名模式为：名称\_附加信息。其中，“名称”是指可以反映表/文件内容的英文关键词或者缩写；“\_附加信息”可以省略。附加信息用于区分同结构的表，例如用户对应的朋友关系表，每个用户对应一个，所以为了区分不同用户的朋友关系表，需要加上附加信息以却别，此时，附加信息就是该用户的标识（账户）。

一个用户对应一个独立的关系表、群表、会话/微报/博报/收藏文件。会话记录中的文件或者表的字段，都是指该文件或者表的名称。这些表都用表名（或者文件名）指称。

会话表记录结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户id | userID | varchar(32) | 标志唯一的用户，用户的账号，10位数字表示 |
| 朋友关系表 | relation\_table | varchar(32) | 指向与其他用户关联的关系描述表，表示好友用户关系。用朋友关系表的表名来表示 |
| 群组表 | group\_table | varchar(32) | 每个用户记录一张群表，记录用户所在群的编号、种类和群里的成员、数目。用群组表的表名来表示 |
| 会话文件 | conversation\_file | varchar(32) | 每个用户一个会话文件，记录与其他用户的会话记录，格式为XML，可以支持样式文件定义形式格式，可以设置是否漫游以及漫游时间。用该会话文件的文件名来表示 |
| 私人收藏 | private\_collect | varchar(32) | 每个用户一个私人收藏文件，记录用户收藏的信息，格式为XML，记录收藏的内容。用该私人收藏文件的文件名来表示 |
| 博报文件 | blog\_passage | varchar(32) | 每个用户一个博报文件，记录用户在会话朋友内或者网络空间内发表的多媒体信息，可以被有网络空间的任意用户搜索到，格式为XML，记录博报的内容和包括的资源，用该博报文件的文件名来表示 |
| 微报文件 | micro\_passage | varchar(32) | 每个用户一个微报文件，记录用户发表的多媒体信息，只能被指定用户集阅读，用该微报文件的文件名来表示 |
| 状态 | state | tinyint(4) | 描述用户当前是否在线以及以什么形式在线：  0-未登录，1-登录但离线，2-蜂窝通信在线，3-WIFI在线，4-有线宽带在线，5-其他形式在线（可以不采用编码，直接用文字表示） |
| 活跃度 | active | int(8) | 表示用户在该系统中的活跃度（具体计算机制未定） |
| 诚信度 | reliability | int(8) | 表示用户在该系统中的信誉（具体计算机制未定） |

4.5.2朋友关系表

friend:朋友会话表。(命名规则：friend\_userID)记录用户与其他所有用户的好友关系，比如关注、被关注、相互关注；和其他用户的联系热度，数值越高表示联系越频繁。 一个用户对应一个独立的关系表，用表名表示（指称）。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户id | userID | varchar(32) | 标志唯一的好友用户，好友用户的账号，10位数字表示 |
| 好友id | userID | int | 用户的好友ID |
| 关系 | relation | varchar(32) | 表示用户之间的好友关系，编码可采用中文，包括关注、被关注、相互关注 |
| 联系度 | intimacy | int(11) | 数值，表示用户与该用户的会话联系热度 |

4.5.3群组表

group:群组表。(命名规则：group\_userID)记录属于该用户的所有的群的信息，包括群的编号、群的种类、群里成员、群成员数目，每个用户有一张群表。每条记录描述一个群。每个群都有一个唯一的账户，与普通用户的账户类似，但属于特殊区段：“9”开头的账户。

一个群隶属一个管理者（创建者），作为该用户的一个特殊用户（朋友）。

群成员记录在群成员表中，指向一个群成员表。每个群一个群成员表，列出其各个成员。群成员表用群账号标识，形式为“member\_群账号”。

每个用户可以拥有（作为群主）多个群，同时也可以属于多个群的成员。所以群的数目可能大于用户数目。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 群编号 | groupID | varchar(32) | 标志唯一的群编号，表示特定的一个群。群编号与普通用户的编号方法相同，但规定群号的码段为9XXXXXXXXXX |
| 群种类 | type | varchar(32) | 表示该群所属的种类，可表示为临时群、500人永久群、1000人永久群、2000人永久群 |
| 成员数目 | member\_num | int(11) | 群的当前成员数目 |
| 群成员表 | member\_table | varchar(32) | 关联到表示该群成员ID的群成员表，记录属于该群的所有群成员ID。 |

4.5.4群成员表

Group\_member\_table:群成员表。(客户端命名规则：member\_groupID)记录属于该群的所有成员的用户ID、每个用户在群里的昵称、每个用户在群里的身份。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户id | user\_id | varchar(32) | 表示该群里的每个成员的用户ID |
| 群中昵称 | nick\_name | varchar(128) | 表示用户在该群中的昵称 |
| 身份 | identity | varchar(32) | 表示用户在该群中的身份，可表示为：群主（创建者）、管理员、成员 |

离线消息表

offline\_message:离线消息表，用于存储离线消息。在用户登录，推送消息后删除。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 发送者 | **senderID** | int（11） |  |
| 接收者 | **receiverID** | int（11） |  |
| 时间 | **time** | datetime |  |
| 消息名称 | **msgName** | varchar(128) |  |
| 类型 | **msgType** | tinyint（2） |  |
| 内容 | **msgContent** | text |  |
| 是否已读 | **isRead** | tinyint |  |
|  |  |  |  |

4.5.5订阅消息表

subscribe:订阅消息表（命名规则：subsrcibe）

订阅消息表整个系统一张，记录了每个用户的订阅消息。订阅消息是指用户自己通过设置消息的接受条件，满足该条件时消息推送给用户，这样的消息称为订阅消息。

其中订阅消息可分为系统默认的订阅消息，无需用户设置订阅条件的消息；和用户自定义的订阅消息，既用户自己设置订阅条件的消息。

订阅消息是由系统生成的，系统按照某种策略（默认或者用户自己配置）自动检查是否满足用户的订阅条件，满足则生成该消息。

用订阅消息表则记录了用户的这些消息。该表内容是临时的，到期的消息则删除。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 订阅者 | userID | varchar(32) | 订阅者的账号。账号为9999999999时表示群发。 |
| 消息种类1 | subscribe\_type1 | varchar（32） | 表示订阅消息是属于哪个分类，可表示为：系统消息、用户消息 |
| 消息种类2 | subscribe\_type2 | varchar（32） | 表示订阅消息所属的应用种类，可表示为：就业、购物、招标、组团、云制造、投资… |
| 消息关键字 | key\_word | varchar(254) | 可以表示该消息关键信息的关键字，支持用符号分割的多个关键字 |
| 消息发送者 | sendID | varchar(32) | 表示订阅消息的发送者，可以是系统、用户ID，系统用9999900000表示，意味着，99999XXXXX普通用户不可占用 |
| 消息发送时间 | send\_time | int(11) | 表示订阅消息发送的时间 |
| 建议保留时长 | keep\_time | int(11) | 表示订阅消息保留的时间，默认以分钟为单位（默认单位待讨论） |
| 消息内容 | content | longtext | 表示订阅消息的详细内容 |
| 订阅条件 | order\_condition | varchar(254) | 描述该消息的订阅条件，用逻辑表达式描述 |
| 发送状态 | send\_state | tinyint(4) | 描述该消息是否已发送 |

4.5.6订阅消息详情表

subscribe\_detail:订阅消息表（命名规则：subsrcibe\_detail）

订阅消息详情表整个系统一张，记录了向每个用户推送的订阅消息。用于在用户登录后显示订阅的内容。信息保留时间是3个月。超出时间的信息进行删除。记录的消息类型有：用户通过关注商品、店铺、岗位得到的推送消息；用户通过设置订阅条件得到的推送消息；用户好友申请的消息；

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 发送者 | sender | varchar（32） |  |
| 接收者 | receiver | varchar（32） |  |
| 消息内容 | content | varchar(254) |  |
| 消息类型1 | msgType1 | varchar（32） | 推送消息，好友申请，订阅消息 |
| 消息类型2 | msgType2 |  | 信息所属模块：集市，好友，店铺，就业，云制造 |
| 保存时间 | keepTime | timestamp |  |
| 发送时间 | sendTime | timestamp |  |

4.6 校务通数据库设计

4.6.1缴纳费用表

transaction\_fee记录学生在学校缴纳的费用及日期。这个可以进一步扩展用于整个平台的支付记录。如果交易的是某类物品要怎么记录（另起一张交易表吧）？还要确定缴费单号的生成规则。（这里只记录已经缴纳的费用，要缴纳多少，这个应该单独取张表。每个项目下面还要细分，比如学费下每个学院，每个年级可能都是不同的，这些都要反应在编码表中），或者把每个人要缴纳的费用用负数存在数据库中，缴纳了之后就加上一个缴纳金额。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 缴费项目 | item\_id | int(2) | 缴费项目有：1,学费，2,住宿费，3,空调费，4,一卡通费用等 |
| 缴费项目名称 | item\_name | varchar(128) |  |
| 缴费金额 | item\_sum | int(10) | 单位为元 |
| 缴费状态 | item\_state | tiny(2) | 已缴，未缴，未缴清 |
| 缴费日期 | item\_date | int(11) |  |
| 用户id | user\_id | int(11) | 缴费用户id |
| 收费用户id | acceptID | int（11） | 收费方id，这个可能为用户或机构，默认为学校对应的机构id |
| 缴费单号 | fee\_code | varchar(128) | 缴费流程的单号，应该是一个编码，用于追踪支付 |

学生教师基本信息表

stuff\_info 记录学生在学校的基本信息。本来想记录在个人经历表中，但是经历表中很可能还会有带包含关系的记录，如在校期间的获奖记录，会有多条记录；所以并不适合在校务功能中做关联查询。

老师的信息也可以记录在这张表里。（还是另起一张表吧）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户编号 | uid | int（11） |  |
| 姓名 | sName | varchar(128) |  |
| 学号 | SchoolNo | int（11） |  |
| 学校名称 | orgName | varchar(128) |  |
| 学校编号 | orgID | int（11） |  |
| 学院名称 | dprtName | varchar(128) |  |
| 学院编号 | dprtID | int（11） |  |
| 专业名称 | majorName | varchar(128) |  |
| 专业编号 | majorID | int（11） |  |
| 班级名称 | className | varchar(128) |  |
| 班级编号 | classID | int（11） |  |
| 宿舍名称 | dormitoryName | varchar(128) |  |
| 宿舍编号 | dormitoryID | int（11） |  |
| 学制 | duration | tinyint（2） |  |
| 入学时间 | startTime | datetime |  |
| 毕业时间 | endTime | datetime |  |
| 准考证号 | admissionNo | bigint（11） |  |
| 是否报名 | isNetReport | tinyint（2） |  |
| 是否确认信息 | isCheck | tinyint（2） |  |
| 手机号 | phone | bigint（11） |  |

课程申报表

course\_plan 课程部分的逻辑暂定如下。老师申报自己要开设的课程，教务处审批，学生选课，然后教务处排课，将结果保存在课程安排表和学生选课表中。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 课程名称 | courseName | varchar(128) |  |
| 课程代码 | courseCode | int(11) | 用来和选课表关联 |
| 课程性质 | courseNature | varchar(32) | 公选或必修 |
| 开课学院 | college | varchar(32) |  |
| 学院编号 | collegeID | int（11） |  |
| 教师工号 | teacherID | int（11） |  |
| 教师名称 | teacherName | varchar(32) |  |
| 周学时 | weeklyHours | tinyint（2） |  |
| 学分 | credit | tinyint（32） |  |
| 审批结果 | state | tinyint（2） | 1通过或0不通过 |
| 提交时间 | submitTime | datetime |  |

作息时间表

school\_hours 用来查询课程的上课时间。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 年份 | year | int |  |
| 开始时间 | sectionTimeBegin | time |  |
| 结束时间 | sectionTimeEnd | time |  |
| 节 | section | tinyint |  |

校历表

school\_calendar 用来查询假期和休息时间。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
|  |  |  |  |

4.6.2课程安排表

class\_schedule 记录学生的排课信息，课程基本单位为两个小节记一次课。每节40分钟，每次课相当于1个学分，3个学分就每周排三次课。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 课程名称 | courseName | varchar(128) |  |
| 课程代码 | courseCode | int(11) | 用来和选课表关联 |
| 课程性质 | courseNature | varchar(32) | 公选课，专业必修，专业选修， |
| 开课学院 | coursCollege | varchar(128) |  |
| 课程教师姓名 | teacherName | varchar(128) |  |
| 教师编号 | teacherID | int（11） | 老师的编号 |
| 上课地点 | coursePlace | varchar(64) | 上课地点字符串 |
| 上课时间 | courseTime | tinyint(2) | 第几节，1-6. |
| 起始结束周 | courseStartend | varchar(32) | 第几周，3-18 |
| 学分 | courseCredit | tinyint(2) |  |
| 周学时 | weeklyHours | tinyint(2) |  |
| 课程学生容量 | courseSize | int(3) |  |
| 课程学生剩余量 | courseAllowance | int(3) |  |

4.6.3选课表

class\_select 记录学生的选课信息，课程基本单位为两个小节记一次课。每节40分钟，每次课相当于1个学分，3个学分就每周排三次课。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户编号 | uid | int（11） |  |
| 学号 | Sno | int（11） |  |
| 课程编号 | courseId | int（11） |  |
| 课程名称 | courseName | varchar(128) |  |
|  |  |  |  |

4.6.4签到/请假表

class\_attendance 学生的课程签到和请假信息。 签到方式两种：一种是上传地理位置，一种是老师随堂点名；请假需要审批，默认为课程教师，或辅导员。

这里的上课时间可能要经过一些转化。在排课表中记录的是课程的上课节次。需要通过作息时间表把节次信息转换成时间信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户编号 | uid | int（11） | 用于整个平台关联信息，对比学号在学校内关联信息用 |
| 学号 | Sno | int（11） | 学号用于关联 学生在学校内的信息 |
| 课程编号 | courseNo | int（11） |  |
| 上课时间 | classTime | datetime | 一门课会对应多个上课时间和地点，所以在签到和请假的时候都需要传入这两个字段。 |
| 上课地点 | classPlace | varchar（32） |  |
| 签到时间 | signTime | int（11） |  |
| 签到地点 | signSite | varchar（32） |  |
| 请假类型 | leaveType | tinyint（2） | 事假，病假， |
| 请假开始时间 | leaveStart | int（11） |  |
| 请假结束时间 | leaveEnd | int（11） |  |
| 紧急联系人姓名 | contactName | varchar(32) |  |
| 联系人电话 | contact | varchar(32) | 电话有点长，没有好的存储方式，先存成字符串了 |
| 请假事由 | reason | varchar(254) |  |
|  |  |  |  |
| 点名时间 | callTime | int（11） | 由老师手动点名时记录，其余情况为空 |
| 状态 | status | tinyint（2） | 状态为1，签到,2，缺勤,3，请假 |
| 请假审批人 | auditor | varchar(32) | 任课老师或辅导员 |
| 教师编号 | teacherID | int（11） |  |

4.6.5课程作业表

class\_paper学生的课程作业，暂时认为是文件格式。代码文件或手写作业的扫描版

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户编号 | uid | int（11） |  |
| 学号 | Sno | int（11） |  |
| 课程编号 | courseNo | int（11） |  |
| 班级编号 | classNo | int（11） |  |
| 作业名称 | paperName | varchar(128) |  |
| 作业内容 | paperContent | longtext |  |
| 作业提交状态 | submitState | tinyint（2） | 这个貌似不用，有记录的时候就是提交了 |
| 批阅状态 | readState | tinyint(2) |  |
| 分数 | grade | tinyint(3) |  |
| 作业路径1 | paperPath1 | varchar(128) | 最多支持三个附件 |
| 作业路径2 | paperPath2 | varchar(128) |  |
| 作业路径3 | paperPath3 | varchar(128) |  |
| 提交时间 | submitTime | datetime |  |
| 备注 | remark | varchar（254） |  |

作业发布表

publish\_paper老师发布的作业，记录作业的名称，要求，截止时间等内容。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户编号 | uid | int（11） |  |
| 教工编号 | teacherID | int（11） | 老师也应该有一个学号 |
| 课程编号 | courseNo | int（11） |  |
| 课程名称 | className | varchar(32) |  |
| 作业名称 | paperName | varchar(32) |  |
| 作业要求 | requirement | varchar(254) |  |
| 附件1 | appendix1 | varchar(128) | 最多可以提交三个附件，作为补充内容 |
| 附件2 | appendix2 | varchar(128) |  |
| 附件3 | appendix3 | varchar(128) |  |
| 作业开始时间 | startTime | dateTime |  |
| 作业截止时间 | endTime | datetime |  |
| 作业提交人数 | submitNum | int(2) |  |

4.6.6考试安排表

class\_exams 考试安排，记录考试的排程信息，考场，监考等。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 课程编号 | courseNo | int（11） | 4,6级和其它考试使用特殊编号 |
| 考场 | examSite | varchar(32) |  |
| 监考 | invigilator | varchar(128) |  |
| 考试时间 | examTime | varchar（32） |  |
| 考试类型 | examType | tinyint（2） | 随堂考，期中，期末 |
| 考生人数 | perNum | int（3） |  |

4.6.7考试成绩表

class\_examsGrade 学生的考试成绩信息。和考试安排表对应。 貌似目前的设计考试成绩很难批量录入，要假设老师知道课程编号。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户编号 | uid | int（11） |  |
| 学号 | Sno | int（11） |  |
| 考试编号 | ExamID | int（11） |  |
| 课程编号 | courseNo | int（11） |  |
| 期中成绩 | midGrades | float（2） | 默认为浮点数 |
| 期末成绩 | endGrades | float（2） |  |
| 平时成绩 | commGrades | float（2） |  |
| 总评成绩 | totalGrades | float（2） |  |
| 备注 | comment | varchar（254） |  |
| 录入时间 | submitTime | dateTime | 数据提交的时间 |

4.6.8离校手续表

graduation\_formalities 记录离校的流程和对接人。 用机构编号区分学校。部门间还可以有顺序，表示办理时的前后次序。除了离职流程，别的流程也可以这样来处理。但如果数据库只限于一个学校使用的话，就可以不用机构ID。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 部门编号 | dprtID | int（11） | 这里的部门貌似要另建一张表 |
| 机构编号 | orgID | int（11） | 每个学校可能有多个离职流程，用类型区分 |
| 项目名 | itemName | varchar（32） | 离校手续的项目名称 |
| 办理内容 | content | varchar(254) |  |
| 办理地点 | handlePlace | varchar(32) |  |
| 联系人 | contactPerson | varchar(128) |  |
| 联系方式 | contact | varchar(32) | 联系电话的编码 |
| 前置要求 | requirement | varchar(254) |  |
| 类型 | type | tinyint（2） | 1，学生，2，老师，3， |
| 顺序 | order | tinyint（2） | 同一机构同一类型下流程的办理顺序 |

4.6.9离校进度表

graduation\_procedure 记录学生离校手续的办理情况，和手续表对应。 离校进度表需要学校给出离校手续后，给每个人生成一条离校记录，然后去办理，才能查询。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 用户编号 | uid | int（11） |  |
| 学号 | Sno | int（11） |  |
| 机构编号 | orgID | int（11） |  |
| 部门编号 | departmentID | int（11） |  |
| 进度 | status | tinyint（2） | 未办理，已完成 |

4.7 团队和孵化器模块数据库设计

4.7.1团队表

这里还没有厘清团队，成员，项目，孵化器之间的关系。在实际环境中，孵化器中的团队就是公司，所以一个公司不可能入驻两个孵化器。而孵化器的项目是为了培育公司，和团队中的项目（小型多次的任务）不是一个级别的。既然是公司，那么就可以招募和辞退员工。而在孵化器级别的项目中是不允许兼职的。而在公司内部，团队的任务是可以存在一部分共享的，也就是在小型多次的任务中，可以让一个人参与到多个项目中。

这样，孵化器级别的项目记录在孵化器项目表中。这里的项目都是大项目。成员的话默认为团队中所有人，具体任务可以灵活分配。

团队的项目记录在团队项目表中。团队中的项目可以共享开发人员。这里的项目都是小型的项目。

团队的几个含义：1，预备的公司，准备进驻孵化器。2，临时完成某一任务组建的小组（一般这种对应一个具体的项目）。

项目的含义：1，可以提供某类服务的产业，比如核电站项目（需要多方大型机构配合，具体单位实施）。2，某一具体的任务，比如网站开发项目。（孵化器的项目一般指前者，团队的项目一般指后者）

team 记录团队信息 一个用户只能加入一个正式团队，可以加入多个临时团队。团队内的小团队使用项目来区分。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 团队编号 | teamId | int(11) |  |
| 成员名称 | creatorName | varchar(32) | 这个暂时没用 |
| 成员编号 | creatorID | int(11) |  |
| 成员类型 | memberType | tinyint(2) | 1，创建者，0，参与者 |
| 团队名称 | name | varchar(128) |  |
| 专业方向 | field | varcahr(128) |  |
| 创建日期 | createDate | int(11) |  |
| 地区 | address | varchar(128) |  |
| 团队LOGO | logo | varchar(128) |  |
| 团队类型 | teamType | tinyint（2） | 正式，临时 |

4.7.2团队项目表

team\_project 记录团队完成和正在进行的项目信息。 一个用户可以参看自己创建的项目或参与的项目。如果允许一个团队有多个项目的话，其实就是在允许一个人可以参加多个项目。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 团队编号 | teamId | int（11） |  |
| 项目id | projectId | int（11） |  |
| 项目负责人ID | leaderID | int（11） | 创建团队的人默认为团队项目的负责人 |
| 项目名称 | projectName | varchar(128) |  |
| 项目评级 | grade | tinyint(2) |  |
| 项目状态 | projectStatus | tinyint（2） | 1完成，0进行中 |
| 开始时间 | startDate | int（11） |  |
| 结束时间 | endDate | int（11） |  |
| 入驻状态 | incubateStatus | tinyint（2） | 是否申请孵化器，是否通过审核。 |

4.7.3孵化器申请表

incubator\_apply 孵化器的申请信息。孵化器自己有项目，让团队来申请；或者团队带着项目来申请孵化器的项目。孵化器提供的支持有：场地，工商注册。团队入驻孵化器需要经过审核，在孵的团队项目需要接受监管。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 团队编号 | teamId | int（11） |  |
| 孵化器编号 | incubatorID | int（11） |  |
| 项目编号 | projectID | int（11） |  |
| 审批状态 | releaseStatus | tinyint（2） |  |
| 审批意见 | remark | varchar(128) |  |
| 申请内容 | applyContent | varchar（254） |  |
| 协议文档 | protocolDoc | varchar（128） |  |
| 任务分配文档 | taskDoc | varchar(128) |  |
| 附件 | appendix | varchar(128) |  |
| 备注信息 | comment | varchar（254） |  |

4.7.4团队项目日志表

team\_log 记录团队完成和正在进行的项目日志信息。 团队如果入驻了孵化器后，可以选择将日志提交孵化器管理员查阅。内容设置三个字段，依据类型变化\*\*完成内容，\*\*工作计划，\*\*工作总结。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 团队编号 | teamId | int（11） |  |
| 团队成员 | memberID | int（11） |  |
| 日志编号 | logID | int（11） | 主键 |
| 项目编号 | projectID | int（11） |  |
| 日志内容1 | logContent1 | text | 完成内容 |
| 日志内容2 | logContent2 | text | 工作总结 |
| 日志内容3 | logContent3 | text | 工作计划 |
| 日志类型 | logType | tinyint（2） | 1,日报；2，周报；3，月报 |
| 工作地点 | workPlace | varchar(32) |  |
| 进度 | progress | tinyint（2） | 单位为百分比 |
| 备注 | comment | varchar |  |
| 提交日期 | submitTime | datetime |  |

团队项目会议表

team\_meeting ：记录团队的会议内容。会议规模可大可小，大的会议有多方参与。目前这种方式的缺点是无法统计每个人参加的会议。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 团队编号 | teamId | int（11） |  |
| 会议记录员 | memberID | int（11） | 提交记录的用户默认为会议记录员 |
| 参加会议人员 | attend | text | 参会人员可能有很多 |
| 缺席会议人员 | absent | text |  |
| 会议时间 | time | timestamp |  |
| 会议地点 | place | varchar |  |
| 会议主题 | theme | varchar |  |
| 会议内容 | content | text |  |
| 会议总结 | conclusion | text |  |
| 备注 | comm | varchar |  |
| 提交时间 |  |  |  |

4.7.5项目详情表

project\_detail 记录项目的详细信息。 包括分工，进度，子任务等。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 项目编号 | projectId | int（11） |  |
| 项目名称 | projectName | varchar(128) |  |
| 项目类型 | projectType | tinyint（2） |  |
| 项目内容 | projectContent | text |  |
| 公告信息 | bulletin | text |  |
| 发布时间 | publishTime | datetime |  |
| 协议文档 | protocolDoc | varchar(128) |  |
| 任务分配文档 | taskDoc | varchar(128) |  |
| 项目介绍图片 | picture | varchar(128) |  |

4.7.6孵化器项目表

incubator\_project 记录孵化器中的项目信息。团队可以创建项目，孵化器也可以创建项目，所以需要记录团队和项目的关系，及孵化器和项目的关系。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 孵化器编号 | incubatorID | int（11） |  |
| 孵化器名称 | incubatorName | varchar(128) |  |
| 项目编号 | projectID | int(11) |  |

4.7.7项目成员表

project\_member 记录项目信息,项目成员关系，孵化器关系等。孵化器创建的项目是没有成员的，只有团队中有成员。而一个团队中的所有成员不一定都在一个项目中。所以要从项目成员表中查询成员参与的项目。从团队成员表中查询团队中的成员信息。

项目编号和孵化器编号可以为空，其为空的时候表示单纯的团队成员信息。项目和孵化器编号为空时，成员的职责描述可以认为是成员的个人简历（简介）。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 项目编号 | projectID | int(11) |  |
| 团队编号 | teamID | int（11） |  |
| 成员编号 | memberID | int(11) |  |
| 成员角色 | memberRole | varchar(32) |  |
| 职责描述 | responsibility | varchar(254) |  |
| 孵化器编号 | incubatorID | int(11) |  |
| 加入项目时间 | inTime | datetime |  |
| 退出项目时间 | offTime | datetime |  |
| 成员照片 | memberPic | varchar(128) |  |
| 项目状态 | projectState | tinyint（2） | 项目是否还在进行 |
| 参与状态 | memState | tinyint（2） | 该成员是否还在项目中 |

4.7.8孵化器表

incubator\_detail 记录孵化器中的基本信息。 孵化器属于某一个机构。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 机构编号 | orgID | int(11) |  |
| 孵化器编号 | incubatorID | int（11） |  |
| 管理团队编号 | adminTeamID | int(11) |  |
| 孵化器简介 | description | text |  |
| 孵化器地址 | address | varchar(128) |  |
| 介绍图片 | desPic | varchar(128) |  |
| 创建时间 | createTime | datetime |  |
| 联系方式 | contact | varchar(32) |  |
| 在孵项目数 | incubateNum | tinyint(2) |  |
| 管理员编号 | adminID | int(11) |  |
| 管理方案文档 | adminDoc | varchar(128) |  |

4.7.9团队招募表

team\_recruit 记录团队招募成员的信息。这个定位有点尴尬，对于大的团队这个信息应该是正式的，要和公司招聘走差不多的流程。对于小的团队，这个和论坛消息差不多，相当于一个小广告。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段别名 | 字段英文名 | 字段类型 | 备注 |
| 团队编号 | teamID | int(11) |  |
| 简介 | intro | text |  |
| 招募要求 | requirement | text |  |
| 地区 | region | varchar(128) |  |
| 招募人数 | recruitNum | int（3） |  |
| logo |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **详细字段信息**

5.1就业平台模块数据库

就业平台数据库设计是系统中的用户使用系统时，通过对系统的功能过程进行抽象，将业务过程抽象成与之对应的实体对象，方便用户在使用系统的过程中对于相关信息的存储，修改和使用。

**基本表:**

**招聘岗位基本信息**:岗位编码，营业执照号，岗位名称，岗位父类类别，岗位子类类别（目前暂时对岗位类别的设定采用二级分类，例如金融—>证券），发布时间，截止时间，岗位状态（截止/进行），需求人数，岗位关键字（以分号分隔的多个词语），专业类别父类，专业类别子类（也是采用二级类别，参考国家对于大学专业的分类），出勤形式（全职/实习/兼职），待遇，岗位职责，工作地点，岗位要求（申请该岗位的基本要求, 身高/性别/实习时间/学历/职称/工作经验等）。

**用户求职申请表:**申请编码，岗位编码，身份证号码，求职状态，申请时间，申请介绍，求职附件（用户上的自己的学历，获奖证书，照片等相关的证件合成一个pdf作为附件上传），营业执照号。

**合同表:**合同编号，营业执照号，身份证号码，岗位编码，薪资水平，签订时间，签订时间，合同生效时间，合同结束时间，合同更新次数，试用期开始时间，试用期结束时间，合同附件。

**消息表:**身份证号/营业执照号，消息内容，消息发送时间，消息类别，是否回复，回复内容，是否阅读，消息发送者，消息回复时间，是否删除。

**合同模板表**: 公司名称/甲方，甲方联系人，公司营业执照号，公司地址，签订时间，试用期开始时间，试用期结束时间，合同有效期，5险一金约定比例，应聘者/乙方，联系电话，姓名，身份证号，性别，通讯地址，合同条款，合同正文，合同类型，合同名，甲方签名，甲方签订日期，乙方签名，乙方签订日期。

**招聘会表**: 招聘会编码，招聘会时间，招聘会介绍， 招聘会关键字， 招聘会文件 ， 招聘会主办方 ， 招聘会地点 招聘企业数，提供岗位数, 招聘会状态。

**招聘会审核表:** 招聘会编码, 申请说明, 申请时间, 申请人, 审核状态, 审核时间, 审核原因, 招聘会文件。

**招聘会企业表:** 展位号，营业执照号，岗位编码，招聘人数，招聘会编码 备注

**维度表：**

**岗位分类维度表:**

父类:IT互联网 市场商务 电子电气人事行政 外语外贸财经法律 教育咨询 媒体设计

子类:数据库 硬件 软件 研发 销售 推广 电子 电子工程 HR 行政 前台 翻译 英语 日语 金融 投资 银行 法律 法务 律师 教师 幼教 创意 广告 平面设计

**消息类别维度表：**系统消息，订阅消息，认证消息，好友消息

**求职状态维度表:** 等待（招聘单位尚未完全确定是否批准）、批准（已被招聘单位批准） ，处理申请包括被拒的、本人拒绝的、已签订合同的、过期未处理的。

**招聘会主题维度表**：综合类，IT类，医药类，机械类，电子类，通信类，建筑类等

**消息xml格式设计:**

<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<messages name="userId">  
 *<!-- 使用发送者的用户编号来唯一区别一个消息文件，  
 企业是营业执照号，用户是身份证号等-->* <message id="消息发送时间">  
 <time>消息的发送时间</time>  
 <sendUser>消息发送者</sendUser>  
 <receiveUser>消息接收者</receiveUser>  
 <messageContent>消息的内容</messageContent>  
 <messageType>消息类别(存储消息类别的编码)</messageType>  
 </message>  
</messages>

5.2用户数据库

用户是指本系统中需要注册并进行操作使用的对象，包括个人和机构两类。其中个人包括学生、社会人士，机构有企业、学校、政府与社团部门

**基本表：**

* **个人基本信息**：身份证号码，姓名，手机， Email，性别，出生日期，家庭地址，级别/学历，职称，毕业院校，照片，照片更新时间。
* **机构基本信息**：机构代码证号，名称，法人代表，注册地，营业范围，注册资金，职工人数；
* **系统注册账户信息**：身份证号码，账户名，注册手机，注册Email，身份，注册时间，账户状态，认证标志，账户口令，密码找回问题，备用邮箱
* **证件表**：编号，证件编号，证件名称（用分号分隔的证件名），认证机构编号，证件有效期，发证时间，更新时间，证件状态。
* **企业状况**：营业执照号，年度，员工数，技术人员比例，营业额，资产总额，负债总额，净销售额，流动资产，流动负债，营业费用，利润，贷款余额，利息费用，所得税，纳税，研发经费比例，营业额出口比例。
* **个人经历**：编号，用户编号，证明材料编号（多个值以分号分割），身份证，年度，活动类型（大专/大学/研究生/博士生/工作/实习/培训），所在机构，岗位级别（学生/高级岗位/中级岗位/初级岗位/其他），岗位名称，证明人，证明人电话，证明人Email，起始时间，结束时间。

**权限控制部分的设计主要处理用户，角色，任务，权限之间的关系。整个用户和权限之间涉及两个中间结构，一个是角色，一个是任务。角色之间有层级关系，任务之间也有层级关系。最基本的情形用户通过角色来获取权限。然后用户也可以通过分配任务来获得权限。对于隶属于机构的角色，通过加入机构id来处理。**

* **角色表**：机构id（可以为空，表示角色不隶属于任何机构），角色id（对于普通用户来说没有机构），上级角色id（用来描述角色的层次关系），角色名称，角色类别（管理员：可以创建子管理员；团队队长）
* **任务表**：任务id，父任务，任务类型（私有，监管，工作流，审核），机构id（如果有的话，说明任务属于某个机构，没有则属于根任务所分配的用户）
* **任务实例表**：任务id，实例id，用户id（实例id为空或零时，用户为任务的管理者和分解者），开始时间，有效期
* **权限表**：权限id，对象（或资源id），访问类型（增删查改）
* **用户角色表**：用户id，角色id
* **角色权限表**：角色id，权限id
* **实例权限表**：任务实例id，权限id

**维度表（编码表）**

学历：博士、硕士、本科、大专、中专、高中、初中、小学

职称：

行业类型：农、林、渔、牧、工业、建筑业、批发业、零售业、交通运输业、仓储业、邮政业、住宿业、餐饮业、信息传输业、软件和信息技术服务业、房地产开发经营、物业管理、租赁、商业服务业

证件名称：身份证，成绩单，学历证，毕业证，职称证，其他证件名称，法人代表证，

职业领域：

岗位级别：工程师，高级工程师，初级工程师，中级工程师，医师，主任医师，副主任医师，

岗位类型：

企业类型：

学校：

专业：

学院：

学校名称变更表：

学院名称变更表：

专业名称变更表：

单位名称变更表：

t\_person\_info 个人基本信息表 描述个人相关的基本信息

t\_organization\_info 机构基本信息表 描述机构的基本信息

t\_certificate\_info 证件表 描述各种证件相关信息

t\_org\_state 企业状况表 描述企业的运营状况

t\_per\_experience 个人经历表 描述个人的历史经历信息

t\_academic\_dim 学历维度表 描述各种学历类别

t\_professional\_dim 职称维度表 描述各种职称类别

t\_sector\_dim 行业类型维度表 描述各种行业类型

t\_certificate\_dim 证件类型维度表 描述各种证件类型

t\_position\_dim 岗位类型维度表 描述各种岗位类型

5.3会话数据库设计

会话表数据库设计

会话表设计