/Users/tangjian/iWork/iWorkshop/iPic/mwlogo-38.jpg

数据库设计说明书

**2016/10/28**

文档修订记录表

（状态：C―创建，A—增加，M—修改，D—删除，R—发布）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 版本 | 状态 | 简要说明（内容和范围） | 更改人 | 更改日期 | 审批人 | 审批日期 |
| V0.1 | C |  | Whl |  |  |  |
| V0.2 | M |  | whl | 2016/11/15 |  |  |
| V0.3 | M | 修正表设计的错误，完善Tag表，重新改版文档，增加必要的文字说明等。 | Jet | 2016/11/16 |  |  |
| V0.4 | M | 调整文档结构，增加“推荐薪水”计算范围说明，完善2.4的说明。 | Jet | 2016/11/17 |  |  |
| V0.5 | M | 在User表中增加certificate和worktype字段，tag表增加工作内容标签，修改worktype表，恢复taglevel表。 | Jet | 2016/11/17 |  |  |
| V0.6 | M | 增加2.4.2中证书和特色标签的区别，更新目录结构，修改taglevel表。 | Jet | 2016/11/17 |  |  |
| V0.7 | M | 完善2.4.2 中“工作内容标签”，新增“照顾老人”，“照顾小孩”的子标签以及其说明。新增2.4.4发送至前端的tag信息的josn对象说明 | Whl | 2016/11/17 |  |  |
| V0.8 | M | 修改2.4.4，把证书类型与特色类型分开了。修改2.4，tag表中type字段中个标签的顺序 | whl | 2016/11/17 |  |  |
| V0.9 | M | 修改2.4.2增加每个标签的权重说明,增加个别标签的描述。修改了3.2权重算法 | whl | 2016/11/18 |  |  |
| V1.0 | M | 增加Redenvelope，Hint，Message三张表，增加Order表redenvId字段 | Jet | 2016/11/18 |  |  |
| V1.01 | M | 增加User表期望薪水ExpectSalary，调整几个字段默认初始化的值，成都对应编号510100，调整文档结构。 | Jet | 2016/11/19 |  |  |
| V1.02 | M | User表中,服务城市svericeCity类型改为int，password类型改为varchar（32） | Whl | 2016/11/19 |  |  |
| V1.03 | M | 照顾老人标签的精度改为1 | Jet | 2016/11/20 |  |  |
| V1.04 | M | 2.4.2修改“周六工作”标签编号17，调整文档结构。删除原有2.4.1需求条件，需求条件中文通过TagName来获取。 | Jet | 2016/11/20 |  |  |

目 录

[1 前言 1](#_Toc467424162)

[2 数据表 1](#_Toc467424163)

[2.1 User 1](#_Toc467424164)

[2.2 Certificate 3](#_Toc467424165)

[2.3 WorkType 4](#_Toc467424166)

[2.4 Tag 5](#_Toc467424167)

[2.4.1 标签价格 5](#_Toc467424168)

[2.4.2 工作内容计算字段 7](#_Toc467424169)

[~~2.5~~ ~~TagLevel~~ 9](#_Toc467424170)

[2.6 TagUserRe 9](#_Toc467424171)

[2.7 Order 9](#_Toc467424172)

[2.8 InsSpec 10](#_Toc467424173)

[2.9 InsOrder 11](#_Toc467424174)

[2.10 RedEnvelope 12](#_Toc467424175)

[2.11 Hint 12](#_Toc467424176)

[2.12 Message 12](#_Toc467424177)

[3 算法 13](#_Toc467424178)

[3.1 小元“推荐薪水”算法 13](#_Toc467424179)

[3.1.1 推荐薪水 13](#_Toc467424180)

[3.1.2 计算条件 13](#_Toc467424181)

[3.1.3 标签单价算法 13](#_Toc467424182)

[3.2 筛选排序的算法 14](#_Toc467424183)

# 前言

本文档是针对微元汇数据库结构设计的说明。包含[数据表](#_数据表)和[算法](#_算法)两个部分。

# 数据表

## User

用户表，用于存储用户的基本信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空ABLE** | **默认值** | **字段类型及长度** | **描述** |
| Id | 用户id | NO | 自增数字 | int |  |
| EduLevel | 教育程度 | NO | 1 | tinyint | 小学：1  中学：2  大学：3 |
| Role | 角色 | NO | 1 | tinyint | 雇主：1  服务员：2  保险经理：3  运营人员：4用户 0002 服务人员用户 |
| UserName | 姓名 | YES |  | varchar(10) | 经过小元认证的用户真实姓名 |
| NickName | 昵称 | YES |  | varchar(10) | 用户昵称，通过微信授权或者手动修改 |
| Password | 密码 | NO |  | varchar(32) | 加密之后的32为md5码 |
| CreateTimete | 创建时间 | YES |  | datetime | YYYY-MM-DD hh:mm:ss |
| Openid | 微信的openid | YES |  | varchar(45) |  |
| Phone | 电话号码 | YES |  | varchar(11) | 11位手机号 |
| IdCard | 身份证号 | YES |  | varchar(18) | 18位身份证号 |
| Gender | 性别 | YES |  | tinyint | 男：1  女：2 |
| Balance | 余额 | NO | 0.0 | float | 用户充值的金额初始化是0 |
| AvatarUrl | 头像 | YES |  | varchar(256) | 存储头像的URL |
| ExpValue | 经验值 | NO | 0 | int | 初始化0 |
| Birthday | 生日 | YES |  | date | YYYY-MM-DD |
| Status | 状态 | NO | 1 | tinyint | 启用：0  禁用：1启用 |
| Certificate | 证书 | YES |  | varchar(50) | 数据结构参见此表备注1  参见[2.2](#_Certificate) |
| WorkType | 工种 | NO | JSON数组对象 | varchar(50) | 数据结构参见此表备注2  参见[2.3](#_WorkType) |
| WorkStatusus | 在岗状态 | NO | 2 | tinyint | 在岗：0  待岗：1  未知：2  参见此表备注3于系统判断，目前还没有完成订单，暂时用这个字段表示状态 |
| Address | 地址 | YES |  | varchar(256) | 家庭住址 |
| MouthRest | 每月休息 | YES |  | tinyint | 每月休息天数 |
| WorkExp | 服务/工作经验 | NO | 1 | tinyint | 服务/工作经验，单位是月。 |
| SelfEvaluation | 自我评价 | YES |  | varchar(512) | self-evaluation  自我评价 |
| HomeTown | 籍贯 | YES |  | varchar(32) | 籍贯 |
| Marriage | 婚姻情况 | NO | 2 | tinyint | 是：0  否：1  保密：2 |
| Folk | 民族 | YES |  | varchar(10) | 民族：汉 |
| ServiceCity | 服务城市 | NO | 1 | int | 服务城市编号  成都：510100 |
| ProfitUserId | 利益分享人的用户编号 | YES |  | int | 平台和线下服务站站长的一个利益分享字段。该服务员成单后平台需要分享收益给该userid。 |
| ExpectSalary | 期望薪水 | NO | 0.0 | float | 服务员自己填写的期望薪水 |
|  |  |  |  |  |  |

备注1：Certificate字段数据结构：[{ "id":"表id字段1" , "name":"表name字段1” },{ "id":"表id字段2" , "name":"表name字段2” }]

备注2：WorkType字段数据结构：[{ "id":"表id字段1" , "name":"表name字段1” },{ "id":"表id字段2" , "name":"表name字段2” }]

备注3：WorkStat字段，在岗与否的判断，根据订单判断在岗情况，每天23点做一个自动运行的服务，只要当前有订单则认为该服务员在岗。若不清楚则填2（未知）。

## Certificate

证书表，用于存储用户证书的相关信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 自增编号 | NO | 自增数字 | int |  |
| UserId | 用户编号 | NO | 用户编号 | int | 可重复，一人有多个证。参见[2.1](#_User) |
| Name | 名称 | NO | 证书中文名称 | varchar(45) | 例如：育婴师，营养师 |
| CertificateIDateID | 证书编号 | NO | 证书编号 | varchar(45) |  |
| Grade | 等级 | NO | 5 | tinyint | 初级：5  中级：4  高级：3  参见此表备注 |
| Assessmentnt | 评定 | NO | 0 | tinyint | 合格：0  不合格：1 |
| AuthorityDateyDate | 颁发日期 | YES |  | date | YYYY-MM-DD |
| AuthorityUnityUnit | 发证单位 | YES |  | varchar(45) |  |
| SerialNumbermber | 发证流水码所属段号 | YES |  | varchar(45) |  |
|  |  |  |  |  |  |

备注：初级5级，中级4级，高级3级

## WorkType

工种，记录工作类型的中文和编号，是用户、证书表的附表。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 编号 | NO | 数字 | int | 自定义编号 |
| Name | 工作名称 | NO |  | varchar(30) | 参见当前表备注 |

备注

|  |  |
| --- | --- |
| **IdAME** | **Name** |
| 1 | 住家保姆 |
| 2 | 不住家保姆 |
| 3 | 全天钟点 |
| 4 | 半天钟点 |
| 5 | 月嫂 |

## Tag

标签表：记录用户的特性，查询排序的时候使用

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| TagId | 标签id | NO |  | int |  |
| TagName | 标签名称 | YES |  | varchar(16) | 标签中文名称 |
| Type | 类型 | NO | 1 | tinyint | 工种标签：0  工作内容标签：1  特色标签：2  证书标签：3  系统标签：4  参见此表备注 |
| Weight | 权重 | NO | 0 | smallint | 标签排序的参数 |
| Price | Tag基础价格 | NO | 0 | smallint | 价格计算参见2.1 |
| Coefficient | 系数 | No | 0 | float（6.2） | Tag价格计算公式所需的变量，每增加一个单位所增加的数字。 |
| MinValue | 最小数量 | No | 1 | smallint | 小于最小数量按照最小数量计算。例如100平米，3个人。 |
| Precision | 精度 | No | 1 | tinyint | 增加单位，例如10平米，1个人 |

### 标签价格

标签价格

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TagId** | **TagName** | **Weight** | **Type** | **Price** | **描述** |
| 0 | 不住家保姆 | 5 | 0 | 2200 |  |
| 1 | 住家保姆 | 5 | 0 | 2200 |  |
| 2 | 半天钟点 | 5 | 0 | 1000 |  |
| 3 | 全天钟点 | 5 | 0 | 2200 |  |
| 300 | 月嫂 | 0 | 0 | 0 | 联系线下工作人员 |
| 301 | 家电维修 | 0 | 0 | 0 | 联系线下工作人员 |
| 4 | 育婴师证 | 10 | 3 | 200 |  |
| 5 | 催乳师证 | 10 | 3 | 200 |  |
| 6 | 小儿推拿证 | 10 | 3 | 200 |  |
| 7 | 护士证 | 10 | 3 | 200 |  |
| 8 | 营养师证 | 10 | 3 | 200 |  |
| 9 | 茶艺师证 | 10 | 3 | 100 |  |
| 10 | 驾驶证 | 10 | 3 | 100 |  |
| 11 | 面食 | 5 | 2 | 100 |  |
| 12 | 照顾过外国人 | 5 | 2 | 100 |  |
| 13 | 英语 | 5 | 2 | 100 |  |
| 14 | 早教 | 5 | 2 | 100 |  |
| 15 | 插花 | 5 | 2 | 100 |  |
| 16 | 西餐 | 5 | 2 | 100 |  |
| 17 | 周六工作 | 5 | 1 | 200 | 是否能够周六工作 |
| 18 | 小元认证 | 20 | 4 | 0 | 包含实名、证书 |
| 19 | 小元体检 | 20 | 4 | 0 |  |
| 20 | 小元评价 | 20 | 4 | 0 |  |
| 21 | 卫生面积 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 22 | 服务人数 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) | 此标签对应[2.4.3](#_工作内容计算字段)**半天钟点人数** |
| 23 | 服务人数 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) | 此标签对应[2.4.3](#_工作内容计算字段)  **全天钟点人数** |
| 100 | 照顾小孩 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 101 | 小于1岁 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 102 | 大于1岁小于3岁 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 103 | 大于3岁 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 200 | 照顾老人 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 201 | 自理 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 202 | 半自理 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |
| 203 | 不能自理 | 5 | 1 | 参考[2.4.3](#_工作内容计算字段) |  |

备注1：’小于1岁’，’大于1岁小于3岁’，’大于3岁’。为’照顾小孩’的子选项

备注2：’自理’，’半自理’，'不能自理‘。为’照顾老人’的子选项

备注3：‘月嫂’，‘家电维修’目前联系线下工作人员。

### 工作内容计算字段

工作内容计算字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **说明AME** | **Price** | **MinValue** | **Coefficient** | **Precision** |
| 卫生面积（平米） | 200 | 100 | 0.04 | 10 |
| 半天钟点人数（做饭、买菜、洗衣） | 300 | 3 | 0.4 | 1 |
| 全天钟点人数（做饭、买菜、洗衣） | 400 | 3 | 0.4 | 1 |
| 照顾小孩 | 200 | 1 | 参见照顾小孩 | 1 |
| 照顾老人 | 200 | 1 | 参见照顾老人 | 1 |

照顾小孩

|  |  |  |
| --- | --- | --- |
| **界面显示** | **计算方式** | **Coefficient** |
| 小于1岁 | 岁数<1 | 3 |
| 大于1岁小于3岁 | <=1岁数<3 | 2 |
| 大于3岁 | 岁数>=3 | 1 |

照顾老人

|  |  |
| --- | --- |
| **界面显示AME** | **Coefficient** |
| 自理 | 1 |
| 半自理 | 1.5 |
| 不能自理 | 2 |

设定：如果照顾小孩和老人同时选择，服务员薪水翻倍。

#### 发送至前端的json对象说明

给前端的tag信息为一个**二维的**josn**对象数组**。每个对象包涵**tagID**与**tagName**

属性。对于有子选项的’照顾小孩’， ’照顾老人’。额外有sonTag属性。单个Josn对象示例如下：

{

"tagId": 100,

"tagName": "照顾小孩",

"sonTag": [

{

"tagId": 101,

"tagName": "小于1岁"

},

{

"tagId": 102,

"tagName": "大于1岁小于3岁"

},

{

"tagId": 103,

"tagName": "大于3岁"

}

]

}

整个二维的josn对象数组，示例如下：

[

[

*//工种的对象组*

{‘tagId’:000,’tagName’:’不住家保姆’},

{‘tagId’:001,‘tagName’:‘住家保姆’}

],

[

*//工作内容的对象组*

],

[

*//特色的对象组*

] ,

[

*//证书的对象组*

]

]

## ~~TagLevel~~

~~标签的层级表，用于查询筛选。~~

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **~~字段名AME~~** | **~~标题~~** | **~~允许空 BLE~~** | **~~默认值~~** | **~~数据类型和长度~~** | **~~描述~~** |
| ~~Id~~ | ~~自增编号~~ | ~~NO~~ |  | ~~int~~ |  |
| ~~TagId~~ | ~~标签编号~~ | ~~NO~~ |  | ~~int~~ |  |
| ~~SonId~~ | ~~子标签编号~~ | ~~NO~~ |  | ~~int~~ |  |
| ~~Level~~ | ~~筛选的层级~~ | ~~NO~~ |  | ~~int~~ |  |

~~备注：筛选分三层，第一层工种，例如住家保姆、月嫂等；第二层工作内容，例如卫生、照顾小孩，月嫂跳转页面发消息等。~~

## TagUserRe

标签和用户关系表，用于建立用户和标签的关系。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 自增编号 | NO | 0 | int |  |
| UserID | 用户编号 | NO | 0 | int |  |
| TagID | 标签编号 | NO | 0 | int |  |

## Order

交易订单表，用于存储交易数据信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空BLE** | **默认** | **数据类型和长度YPE** | **描述OMMENT** |
| Id | 自增编号 | NO | 自增数字 | Int |  |
| OrderId | 订单标号 | NO | 程序生成编号 | varchar(50)unsigned | YYYYMMDDhhmmss  +TransactionId（支付订单号）单号 |
| RedenvId | 红包编号 | YES |  | int | 红包编号 |
| UserId | 用户编号 | NO | 对应编号 | int unsigned | 雇主用户表编号 |
| ServantId | 服务员编号 | NO | 对应编号 | int unsigned | 服务员用户表编号 |
| Create TimetartDate | 订单创建时间 | NO | 字段创建时间 | datetime | YYYY-MM-DD hh:mm:ss |
| ValidTimendDate | 订单失效时间 | NO | 程序生成晚于订单创建时间30分钟 | datetime | YYYY-MM-DD hh:mm:ss  30分钟支付时间，过期订单作废 |
| PaidTime te | 支付完成时间时间 | YES |  | datetime | YYYY-MM-DD hh:mm:ss  支付完成时间时间 |
| ~~ServiceId~~ | ~~服务类型~~ | ~~YES~~ |  | ~~tinyint 3) unsigned~~ | ~~服务的类型 000 雇佣家族服务人员型 000 雇佣家族服务人员 001 存心消费~~ |
| ServicePrice | 服务费用 | NO | 服务费 | float(6,2) | 服务费用 |
| Salary | 服务员工资 | NO | 薪水 | float(6,2) | 服务员薪水 |
| Status | 状态 | NO | 1 | tinyint4) | 完成交易：0  等待交易：1  取消交易：2：1 完成付款：2 交易取消：3 |
| TransactionId ionID | 支付平台支付订单号 | YES |  | varchar(32) 32) | 支付平台订单号的支付订单号 |
| Platform | 支付平台 | YES |  | varchar(16) | 支付平台英文简称 |
| ResultCode de | 支付结果 | YES |  | varchar(16) | 支付平台返回的支付结果状态码返回的业务结果 |
| ErrCodeAndDes ndDes | 错误代码代码描述返回的错误代码代码描述 | YES |  | varchar(256) 256) | 支支付平台返回的错误代码代码描述返回的错误代码代码描述 |
|  |  |  |  |  |  |

备注：必须是在账单失效时间内支付的账单才是合法账单，状态可以变成完成交易状态。

## InsSpec

保险（Insurance）规格表，用于存储保险规格信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 编号 | NO | 自增数字 | int | 自增编号 |
| Name | 保险名称 | NO | 记录对应保险名称 | varchar(50) | 保险中文名称 |
| InsCom | 保险公司名称 | NO | 保险对应保险公司名称 | varchar(50) | 保险公司中文全称 |
| AgentId | 保险员编号 | NO | 保险经理用户编号 | int | 保险经理用户编号 |
| InsPrice | 保险价格 | NO | 0.00 | float(6,2) | 保险实际价格 |

## InsOrder

保险订单表，存储购买保险的信息。有订单不一定有保险，但有保单一定有订单。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 编号 | NO | 自增数字 | int | 自增编号 |
| InsureId | 保单编号 | NO | 对应保单编号 | varchar(30) | 保险公司保单编号 |
| OrderId | 订单编号 | NO | 对应订单编号 | varchar(50) | 保单对应的订单表中的订单编号 |
| InsSpecId | 规格编号 | NO | 对应规格编号 | int | 规格表中的自增编号 |
| Beneficiary Id | 用户编号 | NO | 对应用户编号 | int | 保险受益人编号，通常是雇主 |
| Status | 状态 | NO | 1 | tinyint | 已出单：0  已通知：1  出单失败：2  保单取消：3  参见此表备注 |
| Desc | 保单描述 | YES |  | varchar(50) | 保单的描述信息。例如：出单失败的原因。 |

备注：Status字段，生成记录默认1。生成记录同时通知对应的保险经理需要出单。保险经理收到通知后，需要保证保单第二天生效。并且需要操作保单状态反馈给平台。

## RedEnvelope

红包表，用于记录红包种类、规格。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 编号 | NO | 自增数字 | int | 自增编号 |
| Name | 名称 | NO |  | varchar(30) | 红包中文名字 |
| Type | 类型 | NO | 1 | tinyint | 注册红包：1 |
| Amount | 金额 | NO | 0 | int | 红包金额 |

## Hint

提示表，用于帮助、须知、提示等信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 编号 | NO | 自增数字 | int | 自增编号 |
| Name | 名称 | NO | 帮助 | varchar(30) | 提示名字 |
| Type | 类型 | NO | 1 | tinyint | 帮助：1  须知：2  提示：3 |
| Text | 内容 | NO | 对应的文字 | varchar(255) | 文字内容 |

## Message

消息表，各种消息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名AME** | **标题** | **允许空 BLE** | **默认值** | **数据类型和长度** | **描述** |
| Id | 编号 | NO | 自增数字 | int | 自增编号 |
| Type | 类型 | NO | 1 | tinyint | 系统消息：1  用户消息：2 |
| FromId | 发送用户 | YES | 用户编号 | int | 发送消息用户编号  系统发送：0 |
| ToId | 接收用户 | YES | 用户编号 | int | 接收消息用户编号 |
| Message | 消息 | NO | 消息 | varchar(255) |  |

# 算法

## 小元“推荐薪水”算法

### 推荐薪水

各个标签的价格之和为推荐价格。推荐薪水是一个区间范围，**推荐薪水下限**是筛选出来的标签价格之和，**推荐薪水上限**是下限价格加上服务员本身特长（特色），例如育婴师证，西餐等。

### 计算条件

参见[2.4.2](#_标签价格)，[2.4.3](#_工作内容计算字段)。

### 标签单价算法

需求包括内容，参见[2.4.1](#_需求条件)。

* 标签价格公式=price\* coefficient\* minValue
* 面积，人数计算公式=工种Tag基础价格+需求 Tag基础价格（（需求 -MinValue）/Precision\* Coefficient）
* 照顾小孩、老人计算公式=照顾人Tag基础价格\*人数\* Coefficient

公式计算示例如下表

|  |  |
| --- | --- |
| **公式名称** | **公式** |
| 标签单价 | = 工种Tag基础价格+***需求1 Tag基础价格***（（***需求1*** -MinValue）/Precision\* Coefficient）….+***需求n Tag基础价格***（（***需求n*** -MinValue）/Precision\* Coefficient）+***照顾人Tag基础价格\*人数\* Coefficient***  例如，半天钟点打扫166平米房间，5人  =1000+200（1+（166-100）/10\*0.04）+300（1+（5-3）/1\*0.4）  =1000+332+540  ≈1870（舍去个位数）  例如，全天钟点打扫166平米房间，5人，1个半岁小孩，1个2.5岁小孩，1个不能自理的老人（**注意同时有小孩和老人**）  =2\*（2200+200（1+（166-100）/10\*0.04）+400（1+（5-3）/1\*0.4）+200\*1\*3+200\*1\*2+200\*1\*2）  =2\*（2200+332+720+600+400+400）  ≈9300（舍去个位数） |

## 筛选排序的算法

思路概述：用户选择的自定义标签权重之和+服务员所有系统标签权重+服务员所有证书标签权重之和从高到低进行排序

例如用户选择tagID：1,2,3,4

SELECT a.userID,SUM(weight) as sumweight

FROM

(SELECT t.userID,t.tagID FROM `taguserre` t LEFT JOIN `tag` ta ON t.tagID=ta.tagID WHERE t.tagID=1 OR t.tagID =2 OR t.tagID=3 OR ta.type=3 OR ta.type=4)

AS a -- 在用户便签关系（taguserre）表中找出含有筛选标签和系统标签的服务员

LEFT JOIN `tag` ta ON a.tagID=ta.`tagID`

GROUP BY a.userID -- 相同的服务员合并

ORDER BY sumweight DESC -- 权重之和由高到低排序

即可完成排序