## 水电表（业务管理-水表管理/电表管理）

相关表：

t\_reading\_schedule——抄表计划

t\_reading\_task——抄表任务（用schedule\_id字段与t\_reading\_schedule的id字段关联）

t\_meter\_data——抄表数据（用task\_id字段与t\_reading\_task的id字段关联）

t\_billing\_meter\_data——计费中间表（house\_code(房号，关联tc\_building的house\_code字段)、meter\_type(水0电1)和billing\_time(抄表年月，t\_meter\_data也有该字段)组合成唯一索引）

tc\_electricity\_meter——电表基础数据（elect\_code关联tc\_building的house\_code字段，code关联t\_meter\_data的meter\_code字段）

tc\_water\_meter——水表基础数据（water\_code关联tc\_building的house\_code字段，code关联t\_meter\_data的meter\_code字段）

t\_meter\_abnormal\_rule——抄表异常规则，字段意义很复杂，尽量搭配注释看吧，基本没什么问题，出问题或者有修改再去细究也来得及

t\_meter\_param\_setup——抄表参数设置，字段意义很复杂，尽量搭配注释看吧，基本没什么问题，出问题或者有修改再去细究也来得及

1. 抄表数据定时任务

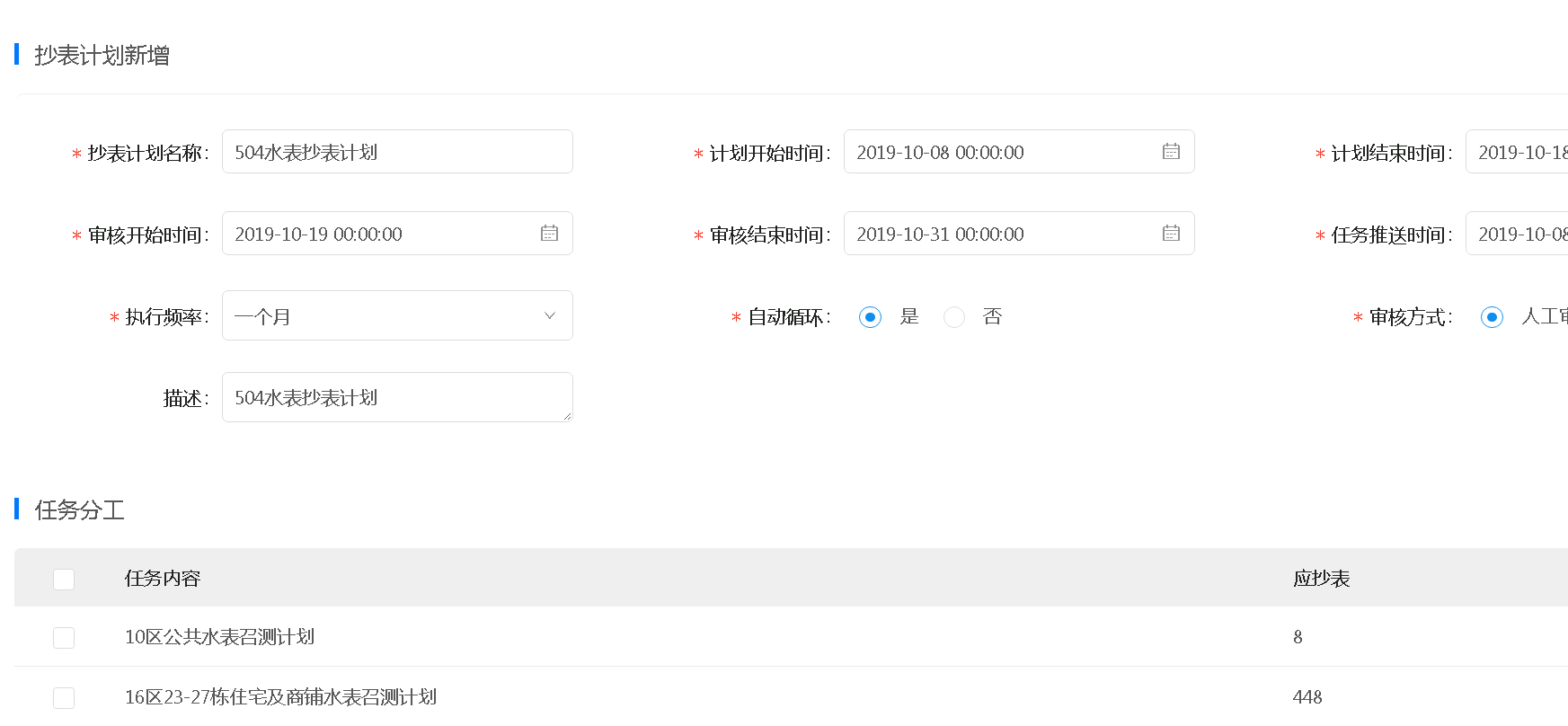


图 1

* 1. meterTaskAutoGenerateJobHandler 抄表数据生成

当前时间超过“任务推送时间”,进入该定时任务核心业务——sortAllMeterData。抄表状态(t\_meter\_data中的status字段)为1代表抄表未完成，将上次读数(last\_total\_reading)填充给本次读数(total\_reading)字段,修改状态为2进行下一步的操作。抄表状态为2的数据代表抄表完成，会依照上次的抄表数据新建本次抄表数据。抄表状态为3的代表下次推送，会修改这些抄表数据为本次抄表数据。

* 1. meterTaskAutoPushJobHandler 抄表数据推送

到达“计划开始时间”，由定时任务将t\_meter\_data中reading\_status为0的数据修改成1。

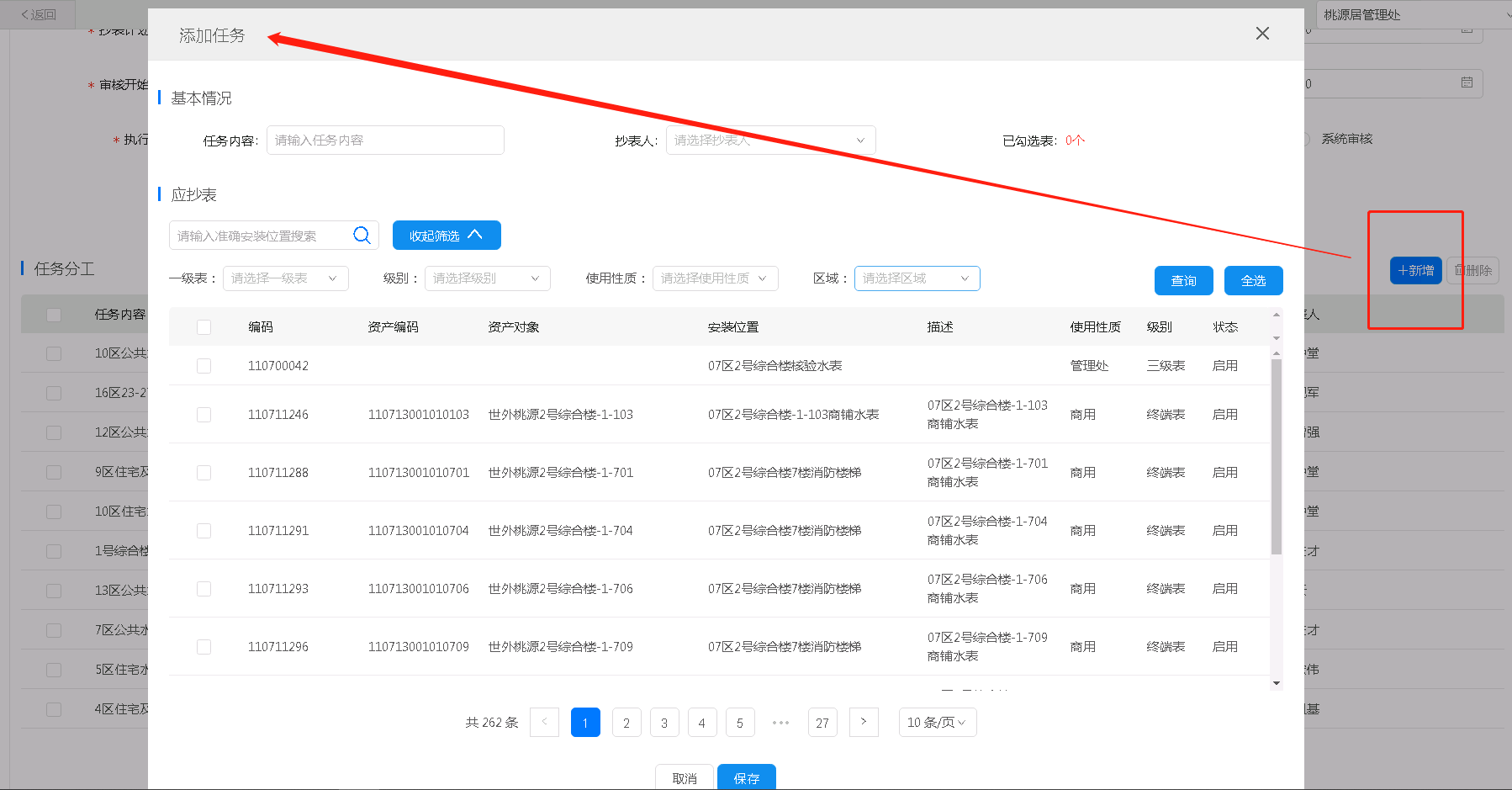
* 1. billingMeterDataGenerateJobHandler 生成计费中间表（正常情况下不使用该定时任务）

不带任何参数的情况下，会删除t\_billing\_meter\_data表数据，将抄表数据整理完插入t\_billing\_meter\_data表中

1. 水电表设置
   1. 抄表计划
      1. 抄表计划新增

新增修改t\_reading\_schedule

* + 1. 任务分工



新增修改t\_reading\_task，但是如果勾选了下面的水电表，就会在t\_meter\_data中新增task\_id等于该抄表任务id且状态为3的抄表数据

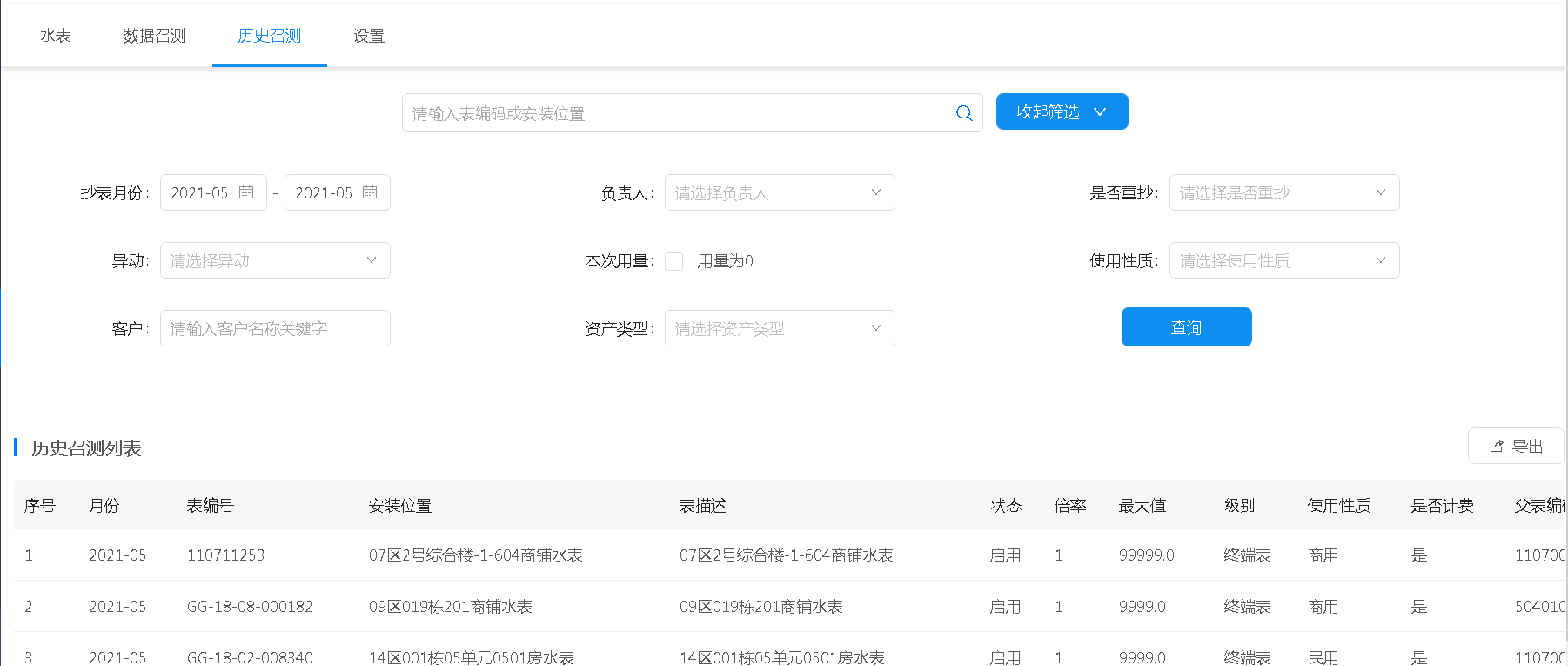
* 1. 数据警报

t\_meter\_abnormal\_rule

* 1. 参数设置

t\_meter\_param\_setup

1. 历史召测



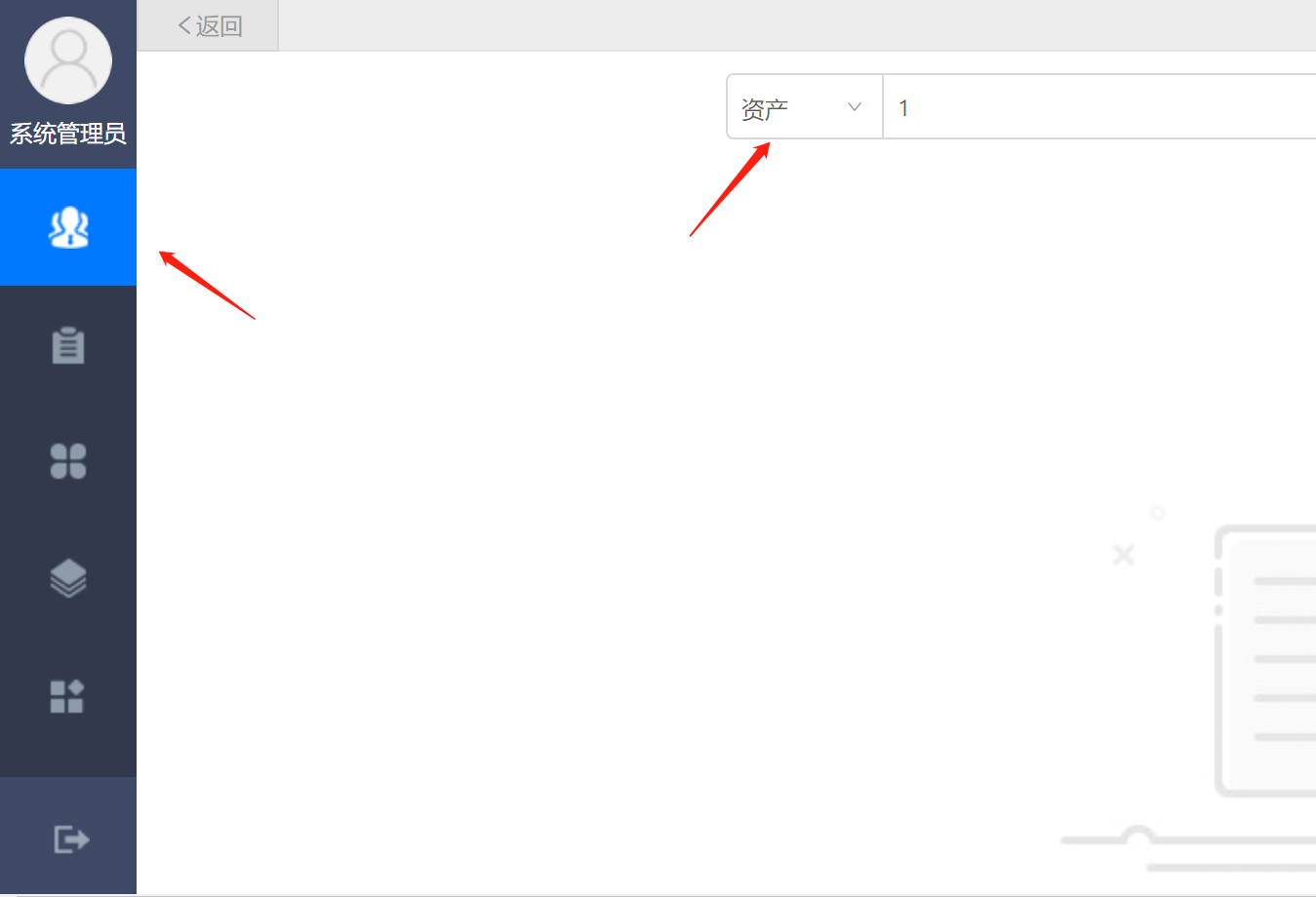
为了避免回表，查询结果的sql和查询分页的sql是分开的，和别的页面不一样，故sql查询很快，但相应的出现修改也要改三部分，导出、查询和分页

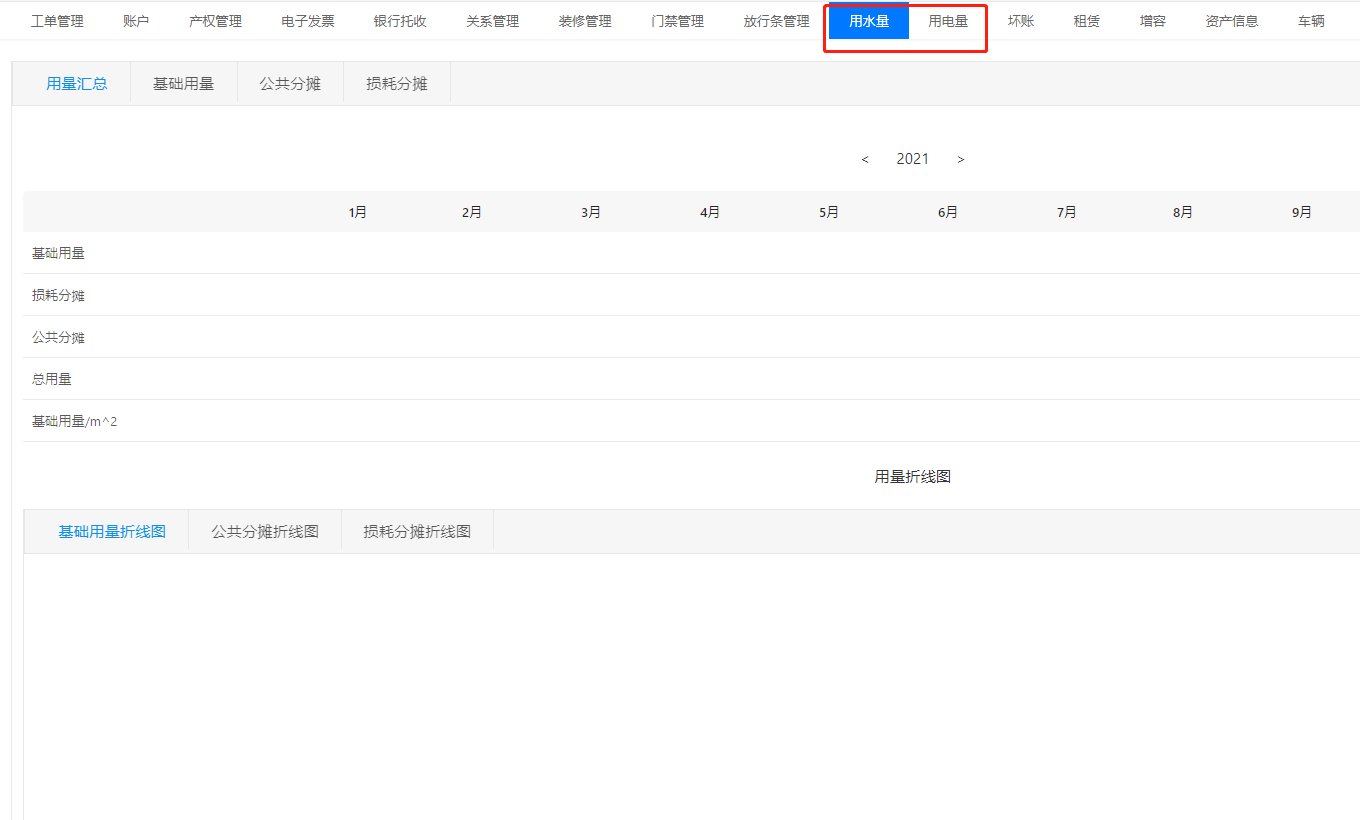
1. 数据召测



除了审核抄表，其余基本都是查询，sql很复杂而且很慢，寄希望于别有需求要大动sql吧。审核抄表通过以及批量通过按钮以及审核通过抄表的修改读数以及重审按钮会涉及到计费中间表(t\_billing\_meter\_data)，这是计费数据的关键数据来源，代码一般没什么问题，但要记得这几个地方会影响到计费中间表的数据。

1. 用水量/用电量（资产搜索入口）





只是查询。

**通常问题：正常不会有啥问题，可能有时候会让去数据库直接改数据，记得面面俱到点。比如修正数据，记得把is\_corrected改成1，代表已修正，填上correction\_time以及把数据修正正确。**

## 财务管理（业务管理-财务管理）

相关表：

t\_project\_account——项目帐户表

t\_project\_cycle\_account——项目周期性账户表(project\_account\_id关联t\_project\_account的id)

t\_project\_cycle\_detail——项目周期性明细表(cycle\_id关联t\_project\_cycle\_account的id)

t\_project\_product\_detail——项目产品明细表(project\_account\_id关联t\_project\_account的id)

t\_project\_refund\_detail——项目退款明细表(project\_account\_id关联t\_project\_account的id)

t\_project\_prestore\_detail——项目预存明细表(prestore\_account关联t\_project\_prestore\_account的id)

t\_project\_prestore\_account——项目预存账户表(project\_account\_id关联t\_project\_account的id)

t\_project\_fine\_detail——项目罚金明细表(project\_account\_id关联t\_project\_account的id)

t\_project\_delay\_detail——项目滞纳金明细表(delay\_account\_id关联t\_project\_delay\_account的id)

t\_project\_delay\_account——项目滞纳金账户表(project\_account\_id关联t\_project\_account的id)

t\_ac\_order——订单表

t\_jg\_staff\_grop——交割成员表(user\_id关联t\_sys\_user的user\_id)

t\_jg\_total\_calculation——银帐交割信息表(关联关系详见后面银帐部分的说明文档)

1. 收入

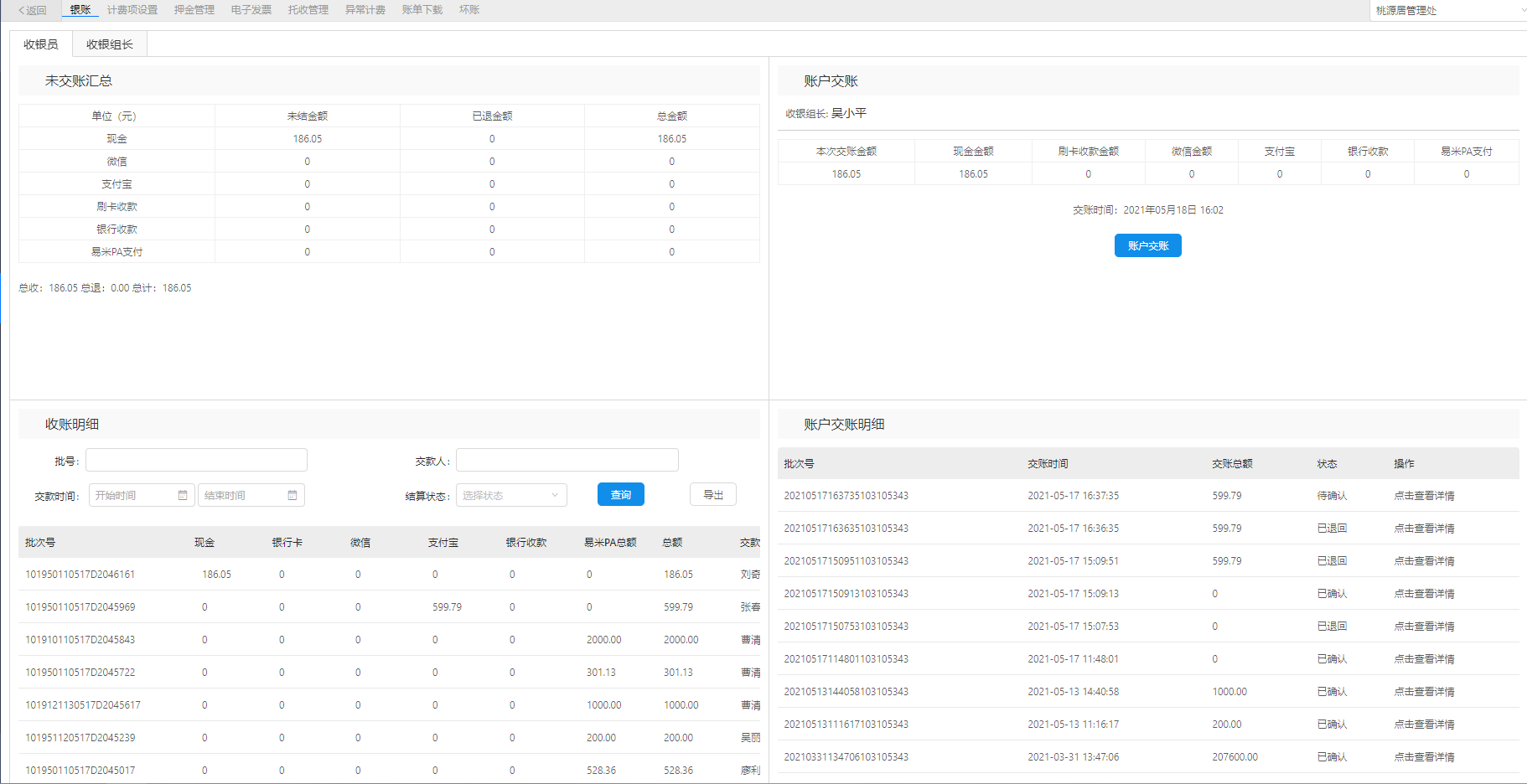
查询出相关表中t\_project\_开头的这部分表，这部分表数据的写入参看com.everwing.autotask.core.service.impl.projectaccount包下的内容，配合注释的内容应该能大致看明白，逻辑很简单，都是从订单(t\_ac\_order)等查出周期性等收入数据放到对应明细表，再整合到账户表中。注意initProjectAccountJobHandler是初始化项目账户的，假如项目账户数据不准确了，可以配置一下需要初始化的公司等参数初始化。

1. 银帐
   1. 管理员银帐界面



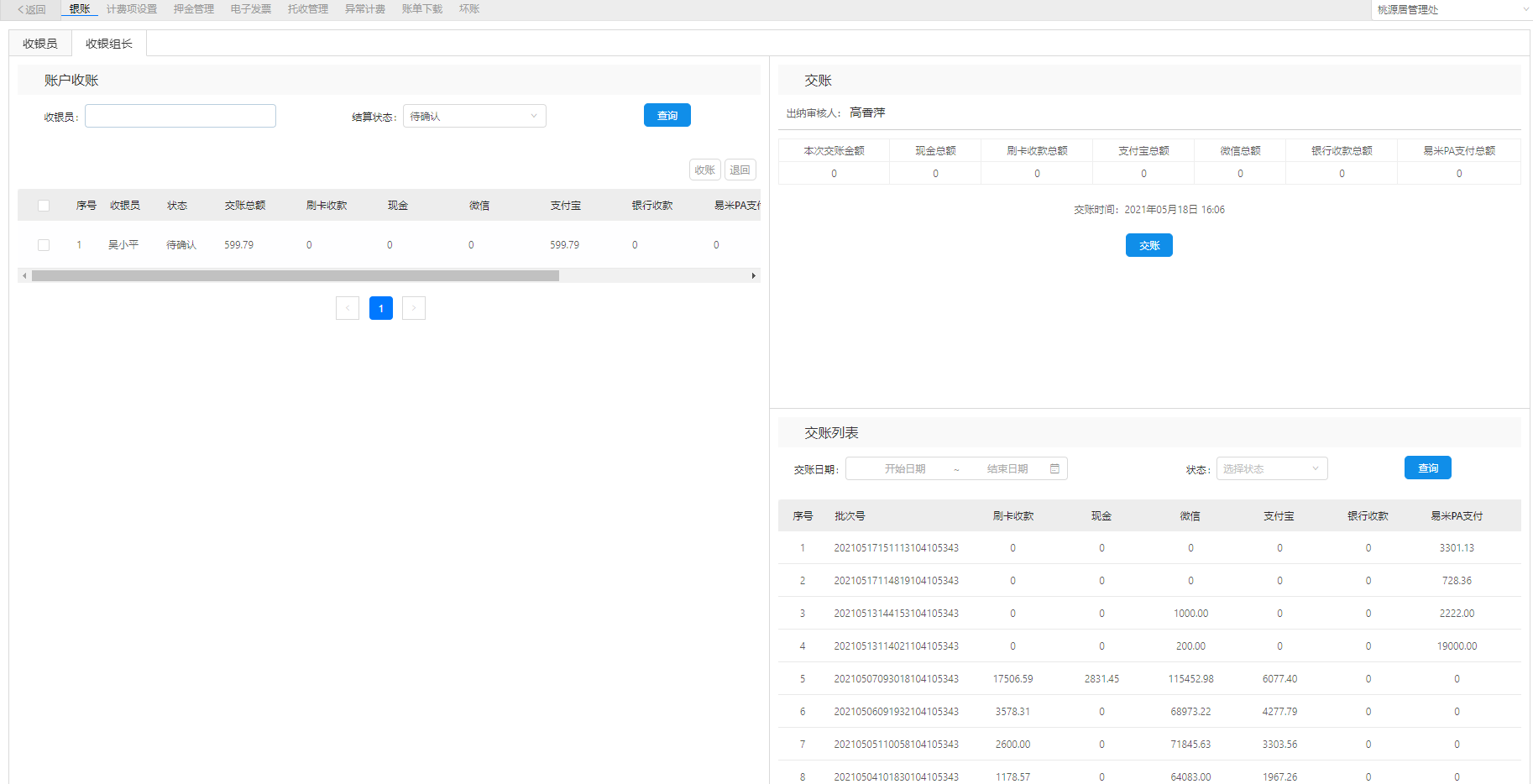
t\_jg\_staff\_grop中role\_lever=0代表是管理员权限，只能通过修改数据库赋予某个组员管理员权限。

* 1. 收银员银帐界面



页面上面的查询很多，具体根据接口参数结果查看具体sql就能理解了。这里的交账会将t\_ac\_order中的数据整合到t\_jg\_total\_calculation中，如购买停车优惠卡使用刷卡方式，那么t\_jg\_total\_calculation的union\_开头的字段就会记录下这个信息。t\_ac\_order会有delivery\_no字段关联t\_jg\_total\_calculation的id，一条银帐交割信息可能会对应多个t\_ac\_order。

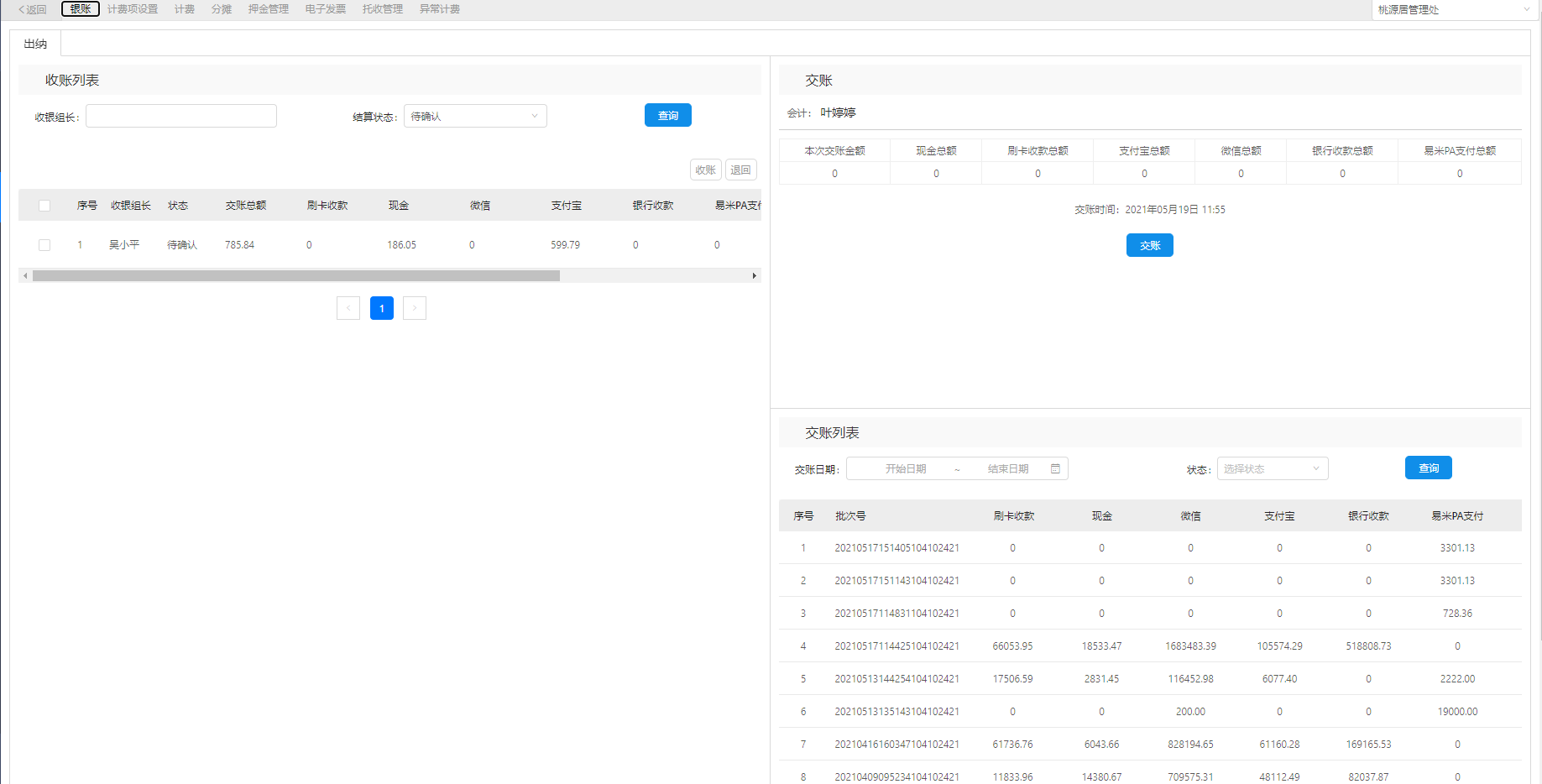
* 1. 收银组长银帐界面



收银员交账后，银帐交割信息会在收银组长的页面被查出，确认和退回会更改t\_jg\_total\_calculation的status字段，同时，退回操作会把t\_ac\_order的delivery\_no字段清空，这导致了退回操作会查不到该交割信息的订单详情。

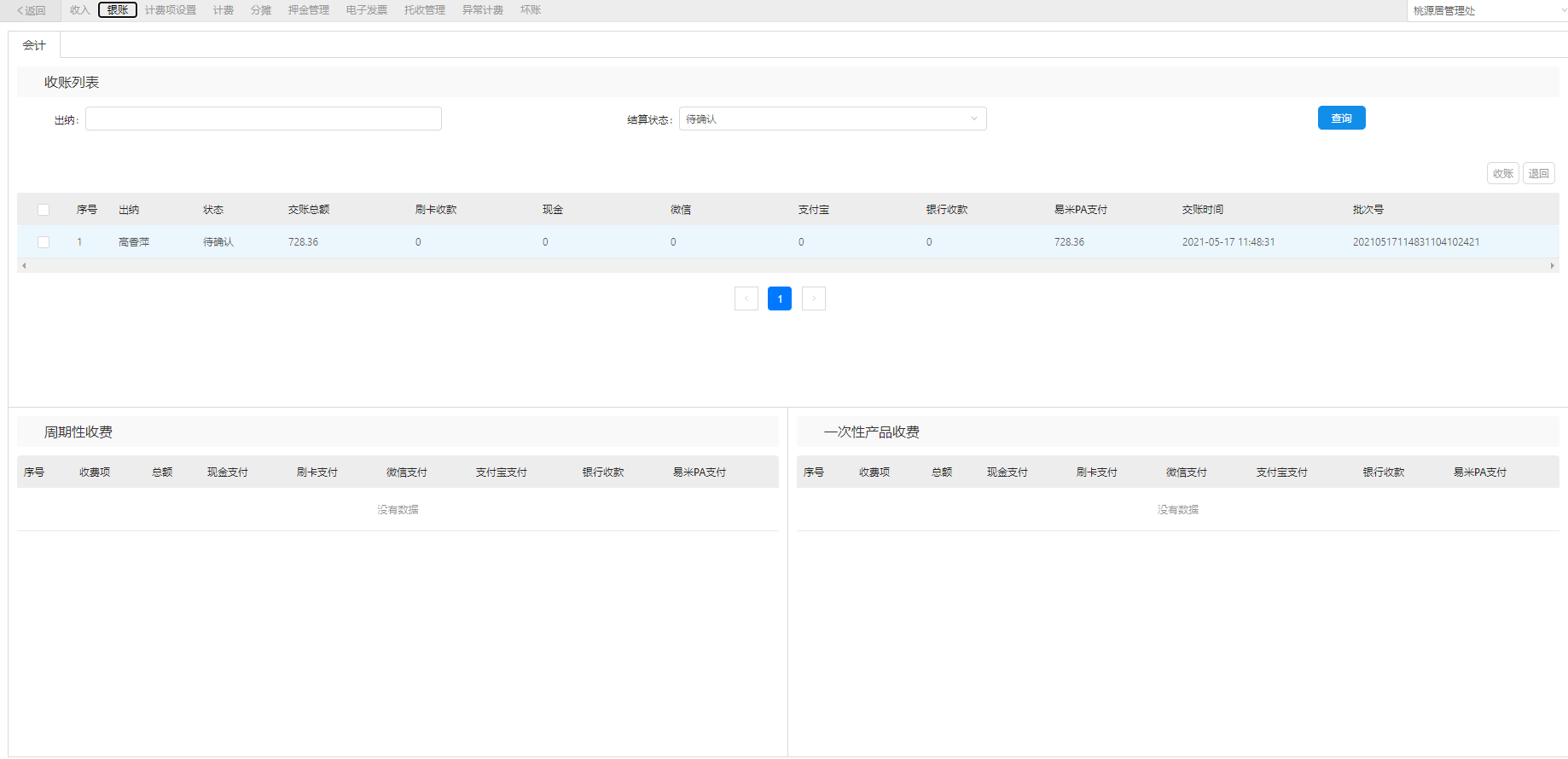
同时这个页面也存在着交账按钮，交账会将生成的t\_jg\_total\_calculation的id字段赋值给原本的t\_jg\_total\_calculation的total\_id字段，同样也是一对多的情况。

* 1. 出纳银帐界面



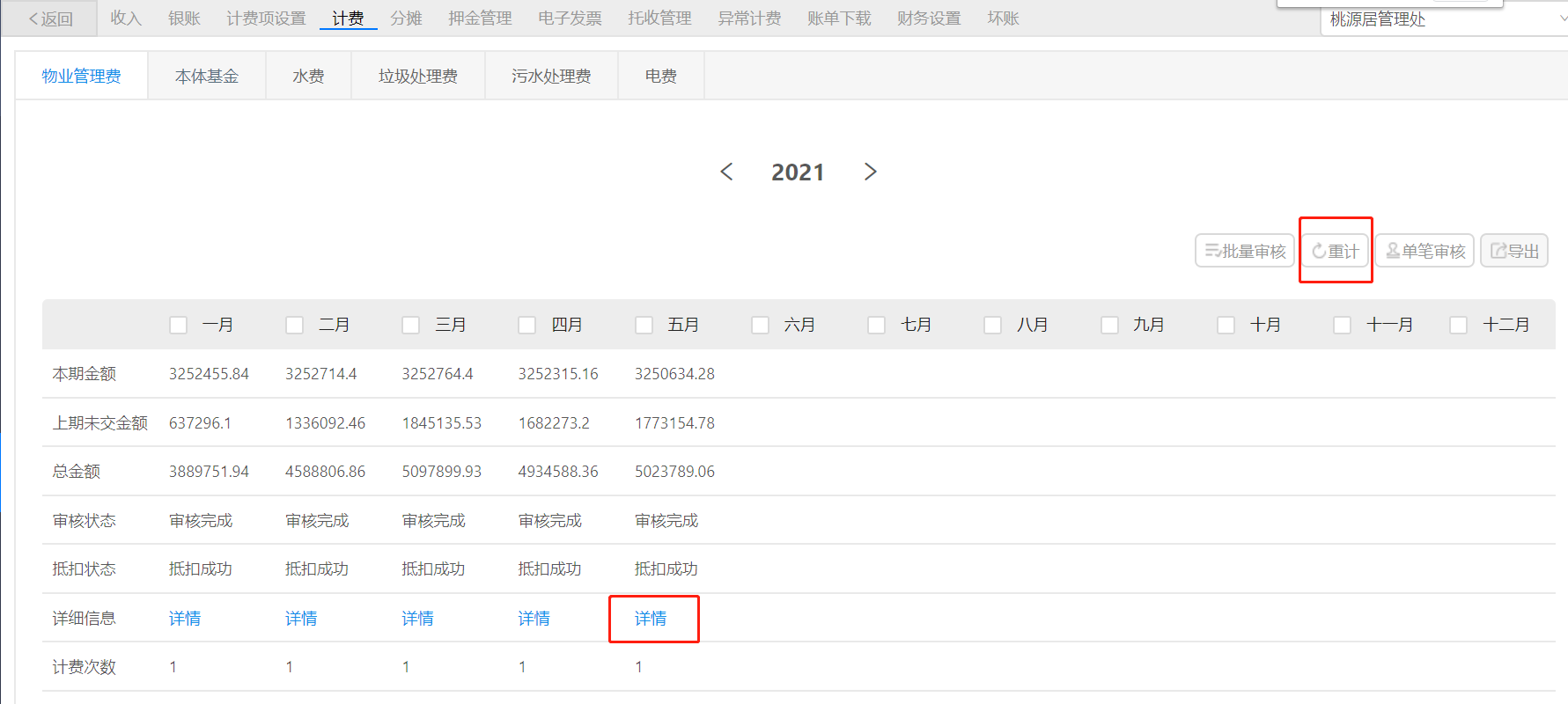
这个和组长银帐界面几乎异曲同工，不做赘述，只是组长交给出纳，出纳会交给会计。

* 1. 会计银帐界面



确认和退回异曲同工，只是下面的周期性收费和一次性产品收费有点小复杂，分别是关联到t\_ac\_cycle\_order\_detail(周期性订单明细表)和t\_ac\_product\_order\_detail(产品订单明细表)，认真看下sql应该能理解的。

1. 计费

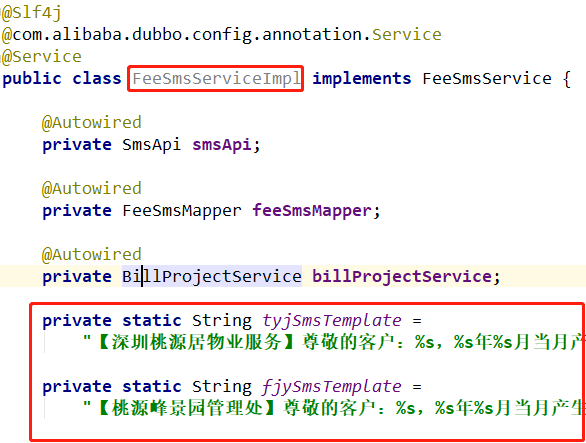


这个页面的查询再加上图中两个圈中按钮的查询，仅是查询而已。

1. 账单下载

主要是短信发送的功能，点击这个发送按钮会给业主发送短信，查询和发送的功能可以去FeeSmsController看下代码，注意这里点发送就会给项目所有业主发短信了，无论是哪个环境，所以不理解的时候不要调试。有时候可能会出现业主说收费短信收不到了，通过queryFeeSmsList找到对应的sql去查询该业主在不在查询出来的结果中，如果存在就解释说可能是短信被短信平台拦截了。





另外配置哪个管理处可以发短信是代码写死的。

**通常问题：两个有时候会扯皮的问题，一个是短信业主收不到咯，只要我们sql查得到那就是短信平台问题，如果查不到排查为什么sql查不到这个业主的信息。另一个问题就是有时候银帐交割账不对，收银员的账不对的话，很大可能性是t\_ac\_order和t\_ac\_business\_opera\_detail(交易明细表，id与t\_ac\_order的opera\_id关联)其中一个表查不到这个收银员表示账不对的数据，组长出纳会计的账就是t\_jg\_total\_calculation值不对，基本都是订单那边状态(pay\_state或order\_state等)或者交易明细表金额不对，然后交账给收银组长时t\_jg\_total\_calculation就开始错了，然后后面交账也跟着错。**

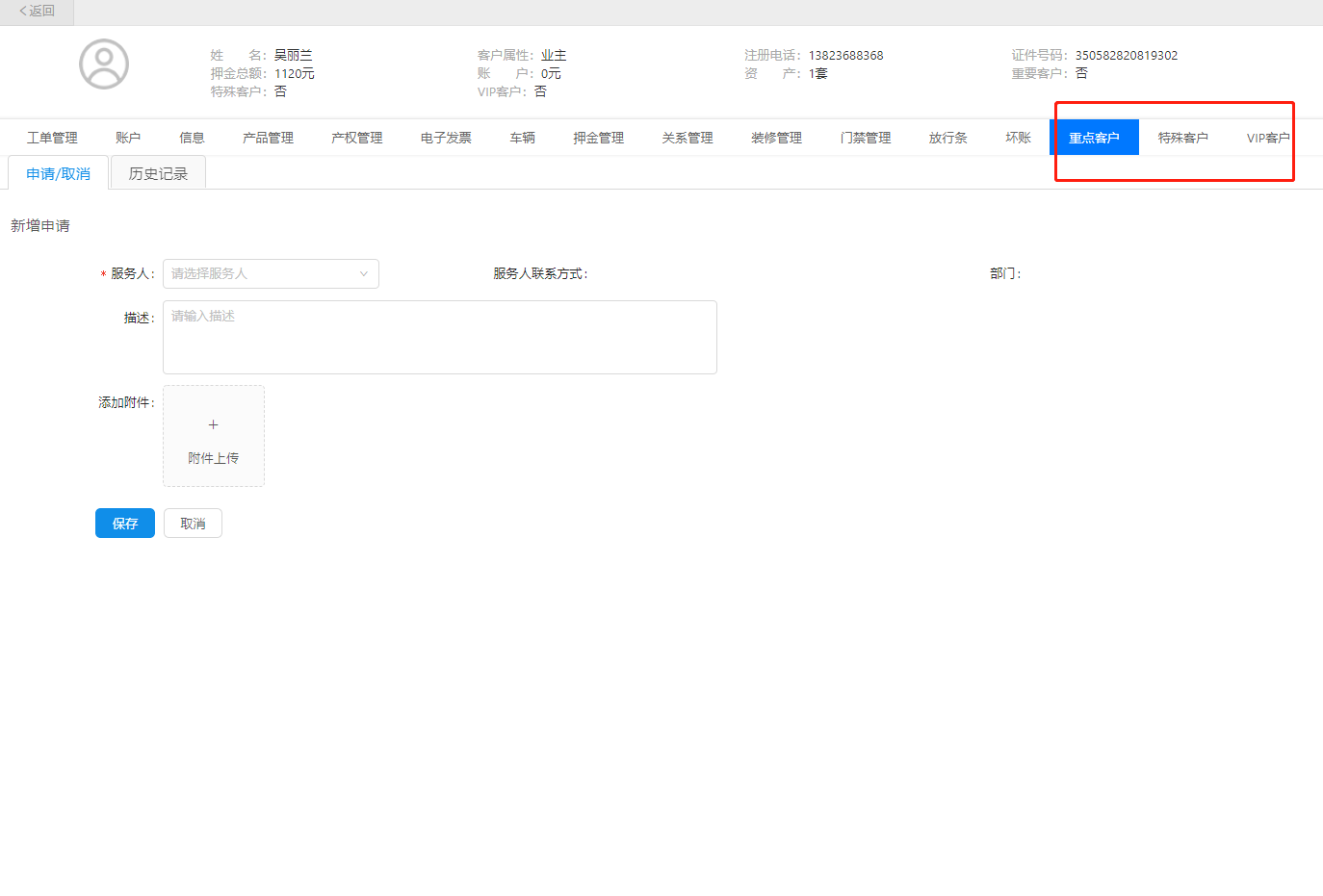
## 重要客户(应用管理-重要客户管理/客户搜索入口-重点|特殊|vip客户)

相关表：

t\_cust\_service\_data——客户服务数据，cust\_id关联tc\_person\_cust的cust\_id

t\_cust\_locus\_log——客户轨迹表，cust\_id关联tc\_person\_cust的cust\_id字段

tc\_person\_cust——个人客户表，关键字段cust\_type，二进制，0否1是，从低位到高位分别代表重点客户/特殊客户/VIP客户，后续sql判断客户是否重点客户等是使用&(逻辑与)去计算





逻辑很简单，基本就是对上述三个表进行增删改查