**1约定**

股票最小单位是股，资金的最小单位是分。

本文档在数据类型描述方面遵循以下的约定：

（一）对于时间字段，均采用yyyyMMddhhmmss的形式。

（二）对于整数数字型字段，不足部分右对齐，左补空格；以NX格式表示，其中X代表数字型字符串总长度。

（三）对于字符型字段，不足部分左对齐，右补空格；以 CX格式表示，其中X代表长度。

各字段的取值，若无特别说明，一律大小写敏感。

**2静态数据模型总图**

静态数据模型主要用来描述业务所基于的数据模型，本身并非实现时的数据结构。此处的若干张表在实现时完全有可能被合并为一张大表以提升效率。

下图用UML的类图表示了各重要业务数据之间的关系，图中只画出了最重要的关联：

****

## **3表信息**

## 3.1机构资金账户表（moneyAccount）

Key是orgName+username，value是：

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| orgName | C10 | 组织名字 |
| username | C10 | strader btrader admin |
| usableMoney | int | 账户可用资金余额 |
| frozenMoney | int | 冻结的账户资金 |

## 3.2用户信息表（user）

Key是orgName+username，value是：

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| orgName | C10 | 用户归属的组织名 |
| username | C10 | 用户名btrader strader admin |

注意：不同的用户有不同的权限

## 3.3股票表(stock)

Key是stockId，value是：

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| stockId | N10 | 股票Id |
| issuer | C10 | 股票发行人 |
| stockTotal | N10 | 股票总量 |
| flag | C1 | 股票状态，例如发行前0，发行中1，发行结束未上市2，上市交易中3，上市停牌中4，摘牌5 |
| tradeDate | yyyyMMdd | 交易日 |
| delist | yyyyMMdd | 摘牌日期 |

注意：股票发行状态的变化只能是0变换到1，1变换到2，2变换到3，3变换到4，3变换到5

## 3.4持仓表（holding）

Key是(orgName+username)+stockId，value是：

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| orgName | C10 | 组织名 |
| username | C10 | strader btrader admin |
| stockId | N10 | 股票ID |
| usableHolding | N10 | 可用持仓数量 |
| frozenHolding | N10 | 冻结的持仓数量 |
| inAcc | int | 最新累计输入持仓量 |
| outAcc | int | 最新累计输出持仓量 |

## 3.5发行表（issue）

Key是issueId，value是：

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| issueId | N10 | 发行流水号 |
| stockId | N10 | 发行的股票Id |
| startTime | C14 | 发行开始时间（yyyyMMddhhmmss） |
| endTime | C14 | 发行结束时间(yyyyMMddhhmmss) |
| bidType | C1 | 招标形式(荷兰式招标)’0’代表荷兰式招标 |

## 3.6投标表(bid)

Key是issueId+orgName+username，value是：

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| bidId | N10 | 投标Id |
| issueId | N10 | 发行流水号 |
| orgName | C10 | 组织名 |
| username | C10 | 用户名btrade strader admin |
| reff | N10 | 投标流水号 |
| stockId | N10 | 股票Id |
| offerPrice | int | 报价，（单位：分） |
| stockNumber | N10 | 股票数量（单位：股） |
| bidTime | C14 | 投标时间(yyyyMMddhhmmss) |

## 3.7 订单表（order）

Key是orderId，value是：

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| orderId | N10 | 订单ID |
| time | C14 | 订单时间yyyyMMddhhmmss |
| buy\_sell | C1 | B买入，S卖出 |
| stockId | N10 | 股票Id |
| price | int | 价格(单位：分) |
| number | N10 | 数量（单位：股） |
| orgName | C10 | 机构名 |
| username | C10 | 用户名 |
| cterpartyOname | C10 | 交易对手方机构名(可以指定也可以不指定) |
| cterpartyUname | C10 | 交易对手方用户名 |

## 备注：当不指定交易对手方时，cterpartyOname为空，cterpartyUname也为空

## 3.8成交表（trade）

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| tradeId | N10 | 成交订单流水号 |
| orgName | C10 | 机构名 |
| username | C10 | 用户名 |
| orderId | N10 | 订单Id |
| cterpartyOname | C10 | 交易对手方机构名（参与交易的机构名） |
| cterpartyUname | C10 | 交易对手方用户名 |
| cterpartyOrderId | N10 | 交易对手方订单Id |
| stockId | N10 | 股票Id |
| tradeNumber | N10 | 成交数量 |
| tradePrice | int | 成交价格 |

## **4业务场景**

## 4.1初始化

**业务事件**

初始化账户资金

**描述**

央行负责入金、出金，初始化机构的资金账户

**输入**

moneyAccount表

**流程**

根据orgname+username获得usableMoney

if orgName==""||orgName!="cb" //cb代表央行

return

if money==""

return

if 入金

usableMoney +=money

else if 出金

if usableMoney<money

return "出金金额必须小于机构余额"

usableMoney -=money

else

return "调用参数必须是入金或者出金"

endif

**输出**

moneyAccount表的usableMoney字段被设置完毕

**备注：**输入参数：orgName（机构名），入金或者出金参数（in或者out），money

方法名 init(orgName,username,in,money)

init(orgName, username ,out,money)

可以用orgName+username当作key

## 4.2用户表方法

**业务事件**

维护用户信息表

**描述**

增加用户，删除用户，修改用户信息，查询用户信息，

**输入**

user表

**流程**

添加用户

if orgName==""

return "机构名不能为空"

if username==""

return "用户名不能为空"

遍历user表

if user表中已经有orgName+username

return "用户名已存在"

根据orgName+username保存用户信息

备注：可以用orgName+Username当作key

更新用户

if orgName==""

return "机构名不能为空"

if username==""

return "用户名不能为空" (用户名具有唯一性)

根据orgName+username判断是否存在该用户

if 不存在

return "您要更新的用户不存在"

if newUsername!=""

if user表中已经有newUsername

return "您更新的用户名已经存在"

username=newUsername

endif

根据orgName+Username更新用户信息

查询用户信息

if orgName==""

return "机构名不能为空"

if username==""

return "用户名不能为空" (用户名具有唯一性)

根据orgName+username查询用户信息

If 查不到

return 查不到此用户

return 查到的用户信息

输出

user表更新

方法名： addUser(orgName,username)

updateUser(orgName,username, newusername)//

queryUser(orgName,username)

## 4.3股票表

**业务事件**

维护股票信息的变化

**描述**

SSE有权限进行股票的添加，删除，改变，查询。

股票的状态有发行前、发行中、发行结束未上市、上市、停牌、复牌、摘牌

**流程**

添加股票

if stockId==""||issuer==""||stockTotal==""tradeDate==""

return

//初始化股票状态

flag="发行前"

delist=””

根据stockId保存新添加的股票信息

删除股票

if stockId==""

return

遍历stock表

if stock表没有stockId

return

根据stockId删除

更新股票信息

if stockId==""

return

根据stockId获得该股票的信息

if stock表中不含有stockId

return

根据stockId更新相应信息

查询股票信息

if stockId==""

return

根据stockId查询股票信息

If 查不到股票信息

Return 查不到

return "查询到的股票"

方法名

addStock(stockId,issuer,stockTotal,flag,tradeDate,delist)

deleteStock(stockId)

updateStock(stockId, issuer,stockTotal,flag,tradeDate,delist)

query(stockId)

## 4.4股票发行

**业务事件**

股票发行

**描述**

上市公司进行股票发行，确定发行开始时间，结束时间，招标方式，SSE进行审核，并修改股票表信息。此时股票处于发行状态。

申购：发行人初始化发行量，投标人输入申购数量和价格，

分配：荷兰式竞价分配，所有出价按照最高价优先处理，分配结果告诉竞标者

**输入** stock表，issue表，bid表，holding表，moneyAccount表

**流程**

**股票发行**

If stockId==""|| beginTime==""|| endTime==""|| bidType==""

return "参数不能为空"

根据stockId获得股票信息

if stock表中没有stockId

return "要发行的股票不存在"

if issue表中已经含有issueId

return "要发行的流水号已经存在"

根据stockId修改stock表中的flag状态为发行

根据发行流水号issueId保存股票发行信息

**投标**

If issueId==""&& orgName==""&& stockId==""&& offerPrice==""&& stockNumber==""&& username==""

return

If issue表不含有issueId

return

If 该机构的账户可用余额<offerPrice\*stockNumber

return

//冻结相应资金

moneyAccount.usableMoney -=offerPrice\*stockNumber

moneyAccount.frozenMoney= offerPrice\*stockNumber

//更新资金账户信息

根据orgName+username更新账户资金信息

//putState("orgNameAccount"," orgName :, usableMoney :,frozenMoney:,username:")

初始化投标时间bidTime

初始化投标bidId

//调用putState保存投标信息

根据bidId保存所有的投标信息

根据issueId保存所有投标的bidId

**撤销投标**

if bidId==””

return

if orgName==””||username==””

return

//只有投标本人可以撤销投标

if orgName!=bidId.orgName||username!=bidId.username

return

根据bidId删除投标

**开标**

//获得所有参加投标的机构名

根据issueId获得某次发行的所有投标bidId

根据bidId获得投表信息

将投标信息json字符串转换成对象

对投标表按照价格orgName.offerPrice从高到低排序得到一个数组

对投标表按照投标时间orgName.bidTime排序

累加数组的前n项stockNumber直到其和大于等于stock. stockTotal

对于中标的机构的冻结的金额划到股票发行者的资金账户，并更新中标机构的持仓信息，对于未中标的机构，将冻结的资金解冻

方法： issue(beginTime,endTime,stockId,issueId,bidType)发行

bid(issueId,orgName, username ,stockId,offerPrice,stockNumber)投标

cancelBid(bidId)撤销投标

assign(issueId,stockId)开标

SSE调用issue和assign，券商调用bid，cancelBid

## 4.5股票上市交易

**业务事件**

股票上市交易

**描述**

发送意愿订单，例如：股票持有者发出转让股票意愿订单，股票投资者发出购买股票意愿订单。

点击成交，投资者对自己感兴趣的意愿订单，进行交易。

**输入 order表 moneyAccount表 holding表**

**流程**

**发送意愿订单**

if username==admin

return

if buy\_sell==""||stockId==""||price==""||number==""||orgName==""

return

//股票必须处于上市交易中才能交易

if stockId.flag!=”3”

return

生成交易流水号orderId

//如果是买入

if buy\_sell=="B"

if moneyAccount.usableMoney<price\*number

return

//冻结相应资产,根据orgName+userType获得资金账户信息

moneyAccount.frozenMoney+= price\*number

moneyAccount.usableMoney-= price\*number

根据 orgName+username更新资金账户

endif

//如果是卖出

else if buy\_sell==”S”

if holding.usableHloding<number

return

//冻结相应股票,根据orgName+stockId+userType获得持仓信息

holding.frozenHolding+= number

holding.usableHolding-= number

根据stockId+orgName+username更新持仓信息

endif

根据orderId保存订单信息（相当于订单意愿）

**点击成交**  (指定对手方)

if cterpartyOrderId==””||stockId==""||price==""||number==""||orgName==""||username==””

return

生成orderId

if cterpartyOrderId.cterpartyOname!=””&& cterpartyOrderId.cterpartyOname!=orgame

return

if cterpartyOrderId.cterpartUname!=””&& cterpartyOrderId.cterpartUname!=username

return

//如果订单意愿是卖出，点击成交则是买入

if cterpartyOrderId.buy\_sell==”S”

if number> cterpartyOrderId.frozenHolding

return

if price < cterpartyOrderId.price

return

if usableMoney<pice\*number

return

cterpartyOrderId.frozenHolding-=number

cterpartyOrderId.moneyAccount.usableMoney+=number\*price

usableHolding+=number

moneyAccount.usableMoney-= number\*order.price

endif

//如果订单意愿是买入，点击成交则是卖出

else if order.buy\_sell==”B”

if number> cterpartyOrderId.number

return

if price> cterpartyOrderId.price

return

if usableHolding< cterpartyOrderId.frozenHolding

return

cterpartyOrderId.usableHolding+=number

cterpartyOrderId.moneyAccount.frozenMoney-= number\*order.price

frozenHolding-=number

moneyAccount.usableMoney+= number\*order.price

endif

根据cterpartyOrderId.orgName+ cterpartyOrderId.username更新资金账户

根据cterpartyOrderId.orgName+ cterpartyOrderId .username+order.stockId更新持仓信息

根据orgName+username更新交易对手方资金账户

根据orgName+username+stockId更新交易对手方持仓信息

if number< cterpartyOrderId.number

根据orderId更新订单信息

endif

生成成交订单流水号

根据成交订单流水号保存成交订单信息

备注：

方法名：sendOrder(stockId,stockNumber,price,orgName,username,buy\_sell)

clickTrade(stockId,number,price,orgName,username,orderId)

trader可以调用

4.6 撤单

业务事件

撤单

描述

投资者调用撤单方法进行撤单操作

输入

order表

流程

if orderId==””

return

//订单的发起者才可以调用取消订单

if orgName!=order.orgName

return

if order表中没有该orderId

return

根据orderId撤单，更新订单表

备注

cancelOrder(orderId)

只有该订单的发起者才可以取消订单

4.7 摘牌

**业务事件**

摘牌

**描述**

SSE对不合格的股票进行摘牌

**输入**

stock表

**流程**

if orgName==""||orgName!="SSE"

return

if stockId==""

return

遍历stock表

if stock表中没有此stockId

return

根据stockId摘牌，更新股票表

输出