链滴

证券金融 - 订单管理系统设计与实现

作者：zorkelvll

原文链接：https://ld246.com/article/1594655057767

来源网站：链滴

众所周知，在证券金融行业交易系统设计中都不可避免地涉及到需要一套订单管理系统，以实现买卖双方的交易订单进行交易管理，本文将基于 FIX5.0 协议讨论如何优雅地设计这样一套便于扩展订单管理系统数据模型！

一、订单下单数据模型

数据结构设计

字段	类型	必填	描述	FIX5.0
orderID	String	Y	当前节点系统生成该订单的唯一主键，可携带当前节点信息，简单化解决分布式系统 ID 问题	37
userID	String	Y	该订单所属用户	
tradingAccountID	String	Y	该订单所属用户的交易账户	
clOrdID	String	Y	上游客户下单请求中的唯一请求号	11
transactTime	datetime	Y	下单请求中的客户委托时间	60

<tr>
<td>securityID</td>
<td>String</td>
<td>Y</td>
<td>证券代码</td>
<td>48</td>
</tr>
<tr>
<td>securityIDSource</td>
<td>String</td>
<td>Y</td>
<td>证券代码来源代码：4-ISIN,101-上交所，102-深交所.....</td>
<td>22</td>
</tr>
<tr>
<td>symbol</td>
<td>String</td>
<td>Y</td>
<td>证券名称</td>
<td>55</td>
</tr>
<tr>
<td>securityExchange</td>
<td>String</td>
<td>Y</td>
<td>交易所代码</td>
<td>207</td>
</tr>
<tr>
<td>side</td>
<td>String</td>
<td>Y</td>
<td>买卖方向：BUY-1-看涨，SELL-2-看跌</td>
<td>54</td>
</tr>
<tr>
<td>orderQty</td>
<td>double</td>
<td>Y</td>
<td>委托数量</td>
<td>38</td>
</tr>
<tr>
<td>price</td>
<td>double</td>
<td>Y</td>
<td>委托价格</td>
<td>44</td>
</tr>
<tr>
<td>tradeDate</td>
<td>String</td>
<td>Y</td>
<td>交易日期</td>

<td>75</td>
</tr>
<tr>
<td>cumQty</td>
<td>double</td>
<td>Y</td>
<td>累计成交数量</td>
<td>14</td>
</tr>
<tr>
<td>avgPx</td>
<td>double</td>
<td>Y</td>
<td>平均成交价格</td>
<td>6</td>
</tr>
<tr>
<td>grossTradeAmt</td>
<td>double</td>
<td>Y</td>
<td>累计成交金额</td>
<td>381</td>
</tr>
<tr>
<td>cxlQty</td>
<td>double</td>
<td>Y</td>
<td>撤成数量</td>
<td>84</td>
</tr>
<tr>
<td>leavesQty</td>
<td>double</td>
<td>Y</td>
<td>在途数量</td>
<td>151</td>
</tr>
<tr>
<td>ordStatus</td>
<td>String</td>
<td>Y</td>
<td>订单状态：A-PendingNew、0-New、1-PartiallyFilled、2-Filled；8-Rejected；6-PendingCncel；4-Cancelled</td>
<td>39</td>
</tr>
<tr>
<td>ordRejReason</td>
<td>String</td>
<td>N</td>
<td>订单拒绝原因：102-证券停牌.....</td>
<td>103</td>
</tr>
<tr>
<td>text</td>

<td>String</td>
<td>N</td>
<td>文本</td>
<td>58</td>
</tr>
<tr>
<td>exchangeOrdID</td>
<td>String</td>
<td>N</td>
<td>下游交易所的订单编号 ID</td>
<td></td>
</tr>
<tr>
<td>currentNodeID</td>
<td>String</td>
<td>Y</td>
<td>该记录所属的交易系统节点 ID 标识 for 分布式</td>
<td></td>
</tr>
<tr>
<td>createdAt</td>
<td>datetime</td>
<td>Y</td>
<td>创建时间</td>
<td></td>
</tr>
<tr>
<td>refStrTag1</td>
<td>String</td>
<td>N</td>
<td>扩展 Str 字段 1</td>
<td></td>
</tr>
<tr>
<td>refStrTag2</td>
<td>String</td>
<td>N</td>
<td>扩展 Str 字段 2</td>
<td></td>
</tr>
<tr>
<td>refStrTag3</td>
<td>String</td>
<td>N</td>
<td>扩展 Str 字段 3</td>
<td></td>
</tr>
<tr>
<td>refDoubleTag1</td>
<td>double</td>
<td>N</td>
<td>扩展 Double 字段 1</td>
<td></td>
</tr>

<tr>
<td>refDoubleTag2</td>
<td>double</td>
<td>N</td>
<td>扩展 Double 字段 2</td>
<td></td>
</tr>
</tbody>
</table>
唯一主键：orderID
唯一性索引：tradingAccountID + clOrdID
<h4 id="领域内关键性行为">领域内关键性行为</h4>
<pre><code class="highlight-chroma">   boolean checkStatus(Action action) {
        switch (action.getExecType()) {
            case PendingNew:
                return getOrdStatus() == OrdStatus.PendingNew;
            case New:
                return getOrdStatus() == OrdStatus.PendingNew || getOrdStatus() == OrdStatus.PendingCancel;
            case Trade:
            case PendingCancel:
            case Canceled:
                return getOrdStatus() == OrdStatus.PendingNew || getOrdStatus() == OrdStatus.New || getOrdStatus() == OrdStatus.PartiallyFilled || getOrdStatus() == OrdStatus.PendingCancel;
            case Rejected:
                return getOrdStatus() == OrdStatus.PendingNew || getOrdStatus() == OrdStatus.New || getOrdStatus() == OrdStatus.PendingCancel;
            case CancelRejected:
                return getOrdStatus() == OrdStatus.PendingCancel;
            default:
                LOGGER.warn("no switch case to check this action = {}", action);
                return false;
        }
    }

    void apply(Action action) {
        switch (action.getExecType()) {
            case PendingNew:

```
                setOrdStatus(OrdStatus.PendingNew);
                break;
            case New:
                setOrdStatus(OrdStatus.New);
                setLeavsQty(getOrderQty());
                break;
            case Trade:
                setCumQty(getCumQty() + action.getLastQty());
                setLeavsQty(getOrderQty() - getCumQty() - getCxlQty());
                setCumAmount(getCumAmount() + action.getLastQty() * action.getLastPx());
                computAvgPx();
                if (DecimalUtil.isZero(getOrderQty() - getCumQty())) {
                    setOrStatus(OrdStatus.Filled);
                } else if (DecimalUtil.isZero(getLeavesQty())) {
                    setOrStatus(OrdStatus.Cancelled);
                } else {
                    if (OrStatus.PendingCancel != getOrdStatus()) {
                        setOrdStatus(OrdStatus.PartiallyFilled);
                    }
                }
                break;
            case Rejected:
                setOrdStatus(OrdStatus.Rejected);
                setLeavsQty(0.0d);
                setRejectedReason(action.getRejectedReason());
                break;
            case PendingCancel:
                setOrdStatus(OrdStatus.PendingCancel);
                break;
            case Cancelled:
                setCxlQt(getCxlQty() + action.getCxlQty());
                setLeavsQty(getOrderQty() - getCumQty() - getCxlQty());
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">            if (Deci
alUtil.isZero(getLeavesQty())) {
</span></span><span class="highlight-line"><span class="highlight-cl">                setOr
Status(OrdStatus.Cancelled);
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">            break;
</span></span><span class="highlight-line"><span class="highlight-cl">        case Cance
Rejected:
</span></span><span class="highlight-line"><span class="highlight-cl">            if (getC
mQty() &lt; ConstDefine.TradeManage.XConst.nearlyZero) {
</span></span><span class="highlight-line"><span class="highlight-cl">                setOr
Status(OrdStatus.New);
</span></span><span class="highlight-line"><span class="highlight-cl">            } else {
</span></span><span class="highlight-line"><span class="highlight-cl">                if (Dec
malUtil.isZero(getOrderQty() - getCumQty())) {
</span></span><span class="highlight-line"><span class="highlight-cl">                    se
OrdStatus(OrdStatus.Filled);
</span></span><span class="highlight-line"><span class="highlight-cl">                } else

</span></span><span class="highlight-line"><span class="highlight-cl">                    se
OrdStatus(OrdStatus.PartiallyFilled);
</span></span><span class="highlight-line"><span class="highlight-cl">                }
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">            break;
</span></span><span class="highlight-line"><span class="highlight-cl">        default:
</span></span><span class="highlight-line"><span class="highlight-cl">            LOGGER
warn("no switch case to apply this action = {}", action);
</span></span><span class="highlight-line"><span class="highlight-cl">            break;
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">        LOGGER.info(
applied by actionType={} after result ordStatus={}, orderID={}", action.getExecType(), getOrdS
atus(), getOrderID());
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    boolean compl
te() {
</span></span><span class="highlight-line"><span class="highlight-cl">        return (Deci
alUtil.isZero(getOrderQty() - getCumQty() - getCxlQty())) || getOrdStatus().equals(OrdStatus.Re
ected);
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span></code></pre>
<h3 id="二-订单撤单数据模型">二、订单撤单数据模型</h3>
<h4 id="数据结构设计-">数据结构设计</h4>
<table>
<thead>
<tr>
<th>字段</th>
<th>类型</th>
<th>必填</th>
<th>描述</th>
<th>FIX5.0</th>
</tr>
</thead>
```

<tbody>
<tr>
<td>cxlOrderID</td>
<td>String</td>
<td>Y</td>
<td>当前节点系统生成该撤单记录的唯一主键</td>
<td></td>
</tr>
<tr>
<td>userID</td>
<td>String</td>
<td>Y</td>
<td>该撤单所属用户 冗余 from order</td>
<td></td>
</tr>
<tr>
<td>tradingAccountID</td>
<td>String</td>
<td>Y</td>
<td>该撤单所属用户的交易账户 冗余 from order</td>
<td></td>
</tr>
<tr>
<td>clOrdID</td>
<td>String</td>
<td>Y</td>
<td>撤单请求中的唯一请求号</td>
<td>11</td>
</tr>
<tr>
<td>origClOrdID</td>
<td>String</td>
<td>Y</td>
<td>撤单请求对应的原订单的唯一请求号</td>
<td>41</td>
</tr>
<tr>
<td>orderID</td>
<td>String</td>
<td>Y</td>
<td>撤单请求对应的原订单的订单 ID</td>
<td>37</td>
</tr>
<tr>
<td>tradeDate</td>
<td>String</td>
<td>Y</td>
<td>交易日期</td>
<td>75</td>
</tr>
<tr>
<td>exchangeOrdID</td>
<td>String</td>
<td>N</td>

<td>撤单请求对应的原订单的下游交易所的订单编号 ID 冗余 from order</td>
<td></td>
</tr>
<tr>
<td>cxlRejResponseTo</td>
<td>String</td>
<td>N</td>
<td>撤单拒绝回应类型</td>
<td>434</td>
</tr>
<tr>
<td>cxlRejReason</td>
<td>String</td>
<td>N</td>
<td>撤单拒绝原因</td>
<td>102</td>
</tr>
<tr>
<td>currentNodeID</td>
<td>String</td>
<td>Y</td>
<td>该记录所属的交易系统节点 ID 标识 for 分布式</td>
<td></td>
</tr>
<tr>
<td>createdAt</td>
<td>datetime</td>
<td>Y</td>
<td>创建时间</td>
<td></td>
</tr>
<tr>
<td>refStrTag1</td>
<td>String</td>
<td>N</td>
<td>扩展 Str 字段</td>
<td></td>
</tr>
<tr>
<td>refStrTag2</td>
<td>String</td>
<td>N</td>
<td>扩展 Str 字段 2</td>
<td></td>
</tr>
</tbody>
</table>
唯一主键：cxlOrderID
唯一性索引：tradingAccountID + clOrdID
<h3 id="三-订单报文数据模型">三、订单报文数据模型</h3>
<h4 id="数据结构设计--">数据结构设计</h4>
<table>
<thead>
<tr>

<th>字段</th>
<th>类型</th>
<th>必填</th>
<th>描述</th>
<th>FIX5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>orderID</td>
<td>String</td>
<td>Y</td>
<td>订单 ID</td>
<td>37</td>
</tr>
<tr>
<td>execType</td>
<td>String</td>
<td>Y</td>
<td>报文执行类型：A-PendingNew,0-New,F-Trade;6-PendingCancel,4-Canceled;8-Rejected;5
1-CancelRejected</td>
<td>150</td>
</tr>
<tr>
<td>execID</td>
<td>String</td>
<td>Y</td>
<td>执行回报唯一编号</td>
<td>17</td>
</tr>
<tr>
<td>sequence</td>
<td>double</td>
<td>Y</td>
<td>当日该节点上系统生成的当前报文的序号，严格从 0 开始递增</td>
<td></td>
</tr>
<tr>
<td>userID</td>
<td>String</td>
<td>Y</td>
<td>所属用户 冗余 from order</td>
<td></td>
</tr>
<tr>
<td>tradingAccountID</td>
<td>String</td>
<td>Y</td>
<td>所属用户的交易账户 冗余 from order</td>
<td></td>
</tr>
<tr>
<td>clOrdID</td>
<td>String</td>

<td>Y</td>
<td>下单或撤单请求中的 11</td>
<td>11</td>
</tr>
<tr>
<td>exchangeOrdID</td>
<td>String</td>
<td>N</td>
<td>该报文所属订单的下游交易所的订单编号 ID 冗余 from order</td>
<td></td>
</tr>
<tr>
<td>lastQty</td>
<td>double</td>
<td>Y</td>
<td>当次成交数量</td>
<td>32</td>
</tr>
<tr>
<td>lastPx</td>
<td>double</td>
<td>Y</td>
<td>当次成交价格</td>
<td>31</td>
</tr>
<tr>
<td>cumQty</td>
<td>double</td>
<td>Y</td>
<td>累计成交数量</td>
<td>14</td>
</tr>
<tr>
<td>cxlQty</td>
<td>double</td>
<td>Y</td>
<td>撤成数量</td>
<td>84</td>
</tr>
<tr>
<td>leavesQty</td>
<td>double</td>
<td>Y</td>
<td>在途数量</td>
<td>151</td>
</tr>
<tr>
<td>transactTime</td>
<td>datetime</td>
<td>Y</td>
<td>当次报文达成时间</td>
<td>60</td>
</tr>
<tr>

<td>ordStatus</td>
<td>String</td>
<td>Y</td>
<td>订单状态：A-PendingNew、0-New、1-PartiallyFilled、2-Filled；8-Rejected；6-PendingCncel；4-Cancelled</td>
<td>39</td>
</tr>
<tr>
<td>securityID</td>
<td>String</td>
<td>Y</td>
<td>证券代码</td>
<td>48</td>
</tr>
<tr>
<td>securityExchange</td>
<td>String</td>
<td>Y</td>
<td>交易所代码</td>
<td>207</td>
</tr>
<tr>
<td>side</td>
<td>String</td>
<td>Y</td>
<td>买卖方向</td>
<td>54</td>
</tr>
<tr>
<td>orderQty</td>
<td>double</td>
<td>Y</td>
<td>委托数量</td>
<td>38</td>
</tr>
<tr>
<td>price</td>
<td>double</td>
<td>Y</td>
<td>委托价格</td>
<td>44</td>
</tr>
<tr>
<td>tradeDate</td>
<td>String</td>
<td>Y</td>
<td>交易日期</td>
<td>75</td>
</tr>
<tr>
<td>ordRejReason</td>
<td>String</td>
<td>N</td>
<td>订单拒绝原因类型：102-证券停牌.....</td>

<td>103</td>
</tr>
<tr>
<td>ordRejReasonDesc</td>
<td>String</td>
<td>N</td>
<td>订单拒绝原因</td>
<td>58</td>
</tr>
<tr>
<td>cxlRejResponseTo</td>
<td>String</td>
<td>N</td>
<td>撤单拒绝回应类型</td>
<td>434</td>
</tr>
<tr>
<td>cxlRejReason</td>
<td>String</td>
<td>N</td>
<td>撤单拒绝原因</td>
<td>102</td>
</tr>
<tr>
<td>currentNodeID</td>
<td>String</td>
<td>Y</td>
<td>该记录所属的交易系统节点 ID 标识 for 分布式</td>
<td></td>
</tr>
<tr>
<td>createdAt</td>
<td>datetime</td>
<td>Y</td>
<td>创建时间</td>
<td></td>
</tr>
</tbody>
</table>
唯一主键：orderID + execType + execID
<h3 id="四-客户资金数据模型">四、客户资金数据模型</h3>
<h4 id="数据结构设计---">数据结构设计</h4>
<table>
<thead>
<tr>
<th>字段</th>
<th>类型</th>
<th>必填</th>
<th>描述</th>
<th>FIX5.0</th>
</tr>
</thead>
<tbody>
<tr>

<td>tradingAccountID</td>
<td>String</td>
<td>Y</td>
<td>该资金账户 ID</td>
<td></td>
</tr>
<tr>
<td>currency</td>
<td>String</td>
<td>Y</td>
<td>该资金账户币种</td>
<td>15</td>
</tr>
<tr>
<td>initAmount</td>
<td>double</td>
<td>Y</td>
<td>期初金额：日初始化时 = 前一交易日期末金额，盘中不变</td>
<td></td>
</tr>
<tr>
<td>holdingAmount</td>
<td>double</td>
<td>Y</td>
<td>当前金额：成交时发生变动，holdingAmount= holdingAmount+ amount（有正负） - abs(
mount) * (commission +stamp)</td>
<td></td>
</tr>
<tr>
<td>tradableAmount</td>
<td>double</td>
<td>Y</td>
<td>可用金额(若存在 在途占用金额，则该字段对外验资和显示应为：holdingAmount -(在途买金额
+ 在途买佣金 + 在途买印花税) - 累计冻结金额 + 累计解冻金额) =》在该公式中影响因素发送变动
均需要通过该公式计算该值，且对外显示不是直接 get 该值而是通过当时查询时刻该计算公式计算进
展示！！！已报时该字段值保持该笔单子之前的值不变，但是前台显示的值和下一次验资的值是减去
占用的金额的；废单时对外显示的值与数据库的值一直且均是该笔单子之前的值；部成则对外显示与
据库中的值均是扣除全部占用后的值；全成或撤成，则对外显示与数据库中的值均是扣除实际成交那
分占用的值，未参与实际成交的那部分的占用被回退成功</td>
<td></td>
</tr>
<tr>
<td>endAmount</td>
<td>double</td>
<td>Y</td>
<td>期末金额 == 当前金额</td>
<td></td>
</tr>
<tr>
<td>intradayBoughtAmount</td>
<td>double</td>
<td>Y</td>
<td>当日买入成交金额</td>
<td></td>

</tr>
<tr>
<td>intradayEffectiveEntrustBuyAmount</td>
<td>double</td>
<td>Y</td>
<td>当日买入有效（终态时通过该字段回退占用金额）委托金额：待报则加，废单则减，撤成则减全成则减(因为委托价大于等于成交价)</td>
<td></td>
</tr>
<tr>
<td>intradaySoldAmount</td>
<td>double</td>
<td>Y</td>
<td>当日卖出成交金额</td>
<td></td>
</tr>
<tr>
<td>intradayEffectiveEntrustSellAmount</td>
<td>double</td>
<td>Y</td>
<td>当日卖出有效（终态时通过该字段回退占用金额）委托金额：待报则加，废单则减，撤成则减全成则减(因为委托价小于等于成交价)</td>
<td></td>
</tr>
<tr>
<td>intradayBoughtCommission</td>
<td>double</td>
<td>Y</td>
<td>当日买入成交佣金</td>
<td></td>
</tr>
<tr>
<td>intradayEffectiveEntrustBuyCommission</td>
<td>double</td>
<td>Y</td>
<td>类似于 intradayEffectiveEntrustBuyAmount)</td>
<td></td>
</tr>
<tr>
<td>intradaySoldCommission</td>
<td>double</td>
<td>Y</td>
<td>当日卖出成交佣金</td>
<td></td>
</tr>
<tr>
<td>intradayEffectiveEntrustSellCommission</td>
<td>double</td>
<td>Y</td>
<td>类似于 intradayEffectiveEntrustSellAmount</td>
<td></td>
</tr>
<tr>
<td>intradayBoughtStamp</td>

<td>double</td>
<td>Y</td>
<td>当日买入成交印花税</td>
<td></td>
</tr>
<tr>
<td>intradayEffectiveEntrustBuyStamp</td>
<td>double</td>
<td>Y</td>
<td>类似于 intradayEffectiveEntrustBuyAmount</td>
<td></td>
</tr>
<tr>
<td>intradaySoldStamp</td>
<td>double</td>
<td>Y</td>
<td>当日卖出成交印花税</td>
<td></td>
</tr>
<tr>
<td>intradayEffectiveEntrustSellStamp</td>
<td>double</td>
<td>Y</td>
<td>类似于 intradayEffectiveEntrustSellAmount</td>
<td></td>
</tr>
<tr>
<td>freezedAmount</td>
<td>double</td>
<td>Y</td>
<td>累计冻结金额：根据冻结流水对该值进行增减</td>
<td></td>
</tr>
<tr>
<td>unfreezeAmount</td>
<td>double</td>
<td>Y</td>
<td>累计解冻金额：根据冻结流水对该值进行增减</td>
<td></td>
</tr>
<tr>
<td>createdAt</td>
<td>datetime</td>
<td>Y</td>
<td>创建时间</td>
<td></td>
</tr>
</tbody>
</table>
唯一主键：tradingAccountID+ currency
<h4 id="领域内关键性行为-">领域内关键性行为</h4>
<pre><code class="highlight-chroma"><span class="highlight
cl">   double showTradableAmount() {
        return getHo

```
dingAmount() + getUnfreezeAmount() - getFreezedAmount()
        - (getIntrada
EntrustBuyAmount() - getIntradayBoughtAmount())
        - (getIntrada
EntrustBuyComission() - getIntradayBoughtCommission())
        - (getIntrada
EntrustBuyStamp() - getIntradayBoughtStamp());
    }

    void updateTra
ableAmount() {
        setTradable
mount(showTradableAmount());
    }

    boolean check(
ouble amount, double commission, double stamp) {
        double holdi
gAmountTemp = getHoldingAmount() + amount - commission - stamp;
        double trada
leAmountTemp = holdingAmountTemp + getUnfreezeAmount() - getFreezedAmount()
        - (getIntrada
EntrustBuyAmount() - getIntradayBoughtAmount())
        - (getIntrada
EntrustBuyComission() - getIntradayBoughtCommission())
            - (getInt
adayEntrustBuyStamp() - getIntradayBoughtStamp()));
        return holdin
AmountTemp &gt;= 0 &amp;&amp; tradableAmountTemp &gt;= 0;
    }

    void triggerByF
ows(FlowsBizType type, double amount, double commission, double stamp) {
        switch (type)

        case Freeze
mount:
        setFreezedA
ount(getFreezedAmount() + amount);
        updateIntrad
yAmount();
        break;
        case UnFreez
Amount:
        setUnFreeze
Amount(getUnFreezedAmount() + amount);
        updateIntrad
yAmount();
        break;
            case Sell:
            case Amo
ntIncrease:
            case Buy:
            case Amo
```

```
ntDecrease:
</span></span><span class="highlight-line"><span class="highlight-cl">            setHold
ngAmount(getHoldingAmount() + amount - stamp- commission);
</span></span><span class="highlight-line"><span class="highlight-cl">            setEnd
mount(getHoldingAmount());
</span></span><span class="highlight-line"><span class="highlight-cl">            if (type.
quals(WarrantBuy)) {
</span></span><span class="highlight-line"><span class="highlight-cl">                setInt
adayBoughtAmount(getIntradayBoughtAmount() - amount);
</span></span><span class="highlight-line"><span class="highlight-cl">                setInt
adayBoughtCommission(getIntradayBoughtCommission() + commission);
</span></span><span class="highlight-line"><span class="highlight-cl">            setIntrada
BoughtStamp(getIntradayBoughtStamp() + stamp);
</span></span><span class="highlight-line"><span class="highlight-cl">            } else if (
ype.equals(WarrantSell)) {
</span></span><span class="highlight-line"><span class="highlight-cl">                setInt
adaySoldAmount(getIntradaySoldAmount() + amount);
</span></span><span class="highlight-line"><span class="highlight-cl">                setInt
adaySoldFeeCommission(getIntradaySoldCommission() +  commission);
</span></span><span class="highlight-line"><span class="highlight-cl">            setIntrada
SoldFeeStamp(getIntradaySoldStamp() +  stamp);
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">            updateI
tradayAmount();
</span></span><span class="highlight-line"><span class="highlight-cl">            break;
</span></span><span class="highlight-line"><span class="highlight-cl">            default:
</span></span><span class="highlight-line"><span class="highlight-cl">            break;
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span></code></pre>
<h3 id="五-客户持仓数据模型">五、客户持仓数据模型</h3>
<h4 id="数据结构设计----">数据结构设计</h4>
<table>
<thead>
<tr>
<th>字段</th>
<th>类型</th>
<th>必填</th>
<th>描述</th>
<th>FIX5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>tradingAccountID</td>
<td>String</td>
<td>Y</td>
<td>该资金账户 ID</td>
<td></td>
</tr>
<tr>
<td>securityID</td>
<td>String</td>
<td>Y</td>
```

<td>证券代码</td>
<td>48</td>
</tr>
<tr>
<td>securityExchange</td>
<td>String</td>
<td>Y</td>
<td>交易所代码</td>
<td>207</td>
</tr>
<tr>
<td>initQty</td>
<td>double</td>
<td>Y</td>
<td>期初持仓：日初始化时 = 前一交易日期末持仓，盘中不变</td>
<td></td>
</tr>
<tr>
<td>holdingQty</td>
<td>double</td>
<td>Y</td>
<td>当前持仓：成交时发生变动，holdingQty= holdingQty + quantity（/卖方向则-quantity）</d>
<td></td>
</tr>
<tr>
<td>tradableQty</td>
<td>double</td>
<td>Y</td>
<td>可用持仓(若存在 在途占用持仓，则该字段对外验券和显示应为：holdingQty - 在途卖数量 - 计冻结数量 + 累计解冻数量) =》在该公式中影响因素发送变动时均需要通过该公式计算该值，且对显示不是直接 get 该值而是通过当时查询时刻该计算公式计算进行展示！！！ 类似于可用金额</td>
<td></td>
</tr>
<tr>
<td>endQty</td>
<td>double</td>
<td>Y</td>
<td>期末持仓 == 当前持仓</td>
<td></td>
</tr>
<tr>
<td>intradayBoughtQty</td>
<td>double</td>
<td>Y</td>
<td>当日买入成交数量</td>
<td></td>
</tr>
<tr>
<td>intradayEffectiveEntrustBuyQty</td>
<td>double</td>
<td>Y</td>
<td>当日买入有效委托数量：待报则加，废单则减，撤成则减，全成则减（其实是减 0 因为 -orderQty+cumQty = 0）</td>

<td></td>
</tr>
<tr>
<td>intradaySoldQty</td>
<td>double</td>
<td>Y</td>
<td>当日卖出成交数量</td>
<td></td>
</tr>
<tr>
<td>intradayEffectiveEntrustSellQty</td>
<td>double</td>
<td>Y</td>
<td>当日卖出有效委托数量：待报则加，废单则减，撤成则减，全成则减（其实是减 0 因为 -orderQty+cumQty = 0）</td>
<td></td>
</tr>
<tr>
<td>freezedQty</td>
<td>double</td>
<td>Y</td>
<td>累计冻结数量：根据冻结流水对该值进行增减</td>
<td></td>
</tr>
<tr>
<td>unfreezeQty</td>
<td>double</td>
<td>Y</td>
<td>累计解冻数量：根据冻结流水对该值进行增减</td>
<td></td>
</tr>
<tr>
<td>createdAt</td>
<td>datetime</td>
<td>Y</td>
<td>创建时间</td>
<td></td>
</tr>
</tbody>
</table>
唯一主键：tradingAccountID+ securityID + securityExchange
<h4 id="领域内关键性行为--">领域内关键性行为</h4>
<pre><code class="highlight-chroma"> double showTradableQty() {
      return getHoldingQty() + getUnfreezeQty() - getFreezedQty()
       - (getIntradayEntrustSellQty() - getIntradaySoldQty());
  }

  void updateTradableQty() {
      setTradableQty(showTradableQty());

```
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    boolean check(
ouble quantity) {
</span></span><span class="highlight-line"><span class="highlight-cl">        double holdi
gQtyTemp = getHoldingQty() + quantity(根据买卖加减);
</span></span><span class="highlight-line"><span class="highlight-cl">        double trada
leQtyTemp = holdingQtyTemp  + getUnfreezeQty() - getFreezedQty()
</span></span><span class="highlight-line"><span class="highlight-cl">         - (getIntrada
EntrustSellQty() - getIntradaySoldQty());
</span></span><span class="highlight-line"><span class="highlight-cl">        return holdin
QtyTemp &amp;gt;= 0 &amp;amp;&amp;amp; tradableQtyTemp &amp;gt;= 0;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    void triggerByF
ows(FlowsBizType type, double quantity) {
</span></span><span class="highlight-line"><span class="highlight-cl">        switch (type)

</span></span><span class="highlight-line"><span class="highlight-cl">            case Buy:
</span></span><span class="highlight-line"><span class="highlight-cl">            case QtyIn
rease:
</span></span><span class="highlight-line"><span class="highlight-cl">                setHold
ngQty(getHoldingQty() + quantity);
</span></span><span class="highlight-line"><span class="highlight-cl">                setEndQ
y(getHoldingQty());
</span></span><span class="highlight-line"><span class="highlight-cl">                if (type.
quals(FlowsBizType.WarrantBuy)) {
</span></span><span class="highlight-line"><span class="highlight-cl">                    setInt
adayBoughtQty(getIntradayBoughtQty() + quantity);
</span></span><span class="highlight-line"><span class="highlight-cl">                }
</span></span><span class="highlight-line"><span class="highlight-cl">                updateI
tradayQty();
</span></span><span class="highlight-line"><span class="highlight-cl">                break;
</span></span><span class="highlight-line"><span class="highlight-cl">            case Sell:
</span></span><span class="highlight-line"><span class="highlight-cl">            case QtyD
crease:
</span></span><span class="highlight-line"><span class="highlight-cl">                setHold
ngQty(getHoldingQty() - quantity);
</span></span><span class="highlight-line"><span class="highlight-cl">                setEndQ
y(getHoldingQty());
</span></span><span class="highlight-line"><span class="highlight-cl">                if (type.
quals(FlowsBizType.WarrantSell)) {
</span></span><span class="highlight-line"><span class="highlight-cl">                    setInt
adaySoldQty(getIntradaySoldQty() + quantity);
</span></span><span class="highlight-line"><span class="highlight-cl">                }
</span></span><span class="highlight-line"><span class="highlight-cl">                updateI
tradayQty();
</span></span><span class="highlight-line"><span class="highlight-cl">                break;
</span></span><span class="highlight-line"><span class="highlight-cl">            case FreezeQ
y:
</span></span><span class="highlight-line"><span class="highlight-cl">            setFreezedQt
(getFreezedQty() + quantity);
</span></span><span class="highlight-line"><span class="highlight-cl">            updateTrada
leQty();
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">        break;
</span></span><span class="highlight-line"><span class="highlight-cl">        case UnFreez
Qty:
</span></span><span class="highlight-line"><span class="highlight-cl">        setUnFreeze
Qty(getUnFreezedQty() + quantity);
</span></span><span class="highlight-line"><span class="highlight-cl">        updateTrada
leQty();
</span></span><span class="highlight-line"><span class="highlight-cl">        break;
</span></span><span class="highlight-line"><span class="highlight-cl">            default:
</span></span><span class="highlight-line"><span class="highlight-cl">                break;
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span></code></pre>
<h3 id="六-资金持仓变动流水数据模型">六、资金持仓变动流水数据模型</h3>
<h4 id="数据结构设计-----">数据结构设计</h4>
<table>
<thead>
<tr>
<th>字段</th>
<th>类型</th>
<th>必填</th>
<th>描述</th>
<th>FIX5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>tradingAccountID</td>
<td>String</td>
<td>Y</td>
<td>该资金账户 ID</td>
<td></td>
</tr>
<tr>
<td>flowID</td>
<td>String</td>
<td>Y</td>
<td>流水 ID</td>
<td></td>
</tr>
<tr>
<td>tradeDate</td>
<td>String</td>
<td>Y</td>
<td>流水触发的日期</td>
<td>75</td>
</tr>
<tr>
<td>bizType</td>
<td>String</td>
<td>Y</td>
<td>流水类型：买入；卖出；资金增加；资金减少；持仓增加；持仓减少；冻结资金；解冻资金；
结持仓；解冻持仓</td>
<td></td>
```

</tr>
<tr>
<td>securityID</td>
<td>String</td>
<td>Y</td>
<td>证券代码</td>
<td>48</td>
</tr>
<tr>
<td>securityExchange</td>
<td>String</td>
<td>Y</td>
<td>交易所代码</td>
<td>207</td>
</tr>
<tr>
<td>variableValue</td>
<td>double</td>
<td>Y</td>
<td>变化值，取绝对值，大于等于 0：买卖时为当次成交数量；资金增减及冻结解冻时为当次变化额；持仓增减及冻结解冻时为当次变化数量</td>
<td></td>
</tr>
<tr>
<td>orderID</td>
<td>String</td>
<td>N</td>
<td>买卖时 订单 ID</td>
<td>37</td>
</tr>
<tr>
<td>execID</td>
<td>String</td>
<td>N</td>
<td>买卖时 执行回报唯一编号</td>
<td>17</td>
</tr>
<tr>
<td>lastPx</td>
<td>double</td>
<td>N</td>
<td>买卖时 当次成交价格</td>
<td>31</td>
</tr>
<tr>
<td>orderQty</td>
<td>double</td>
<td>N</td>
<td>买卖时 委托数量</td>
<td>38</td>
</tr>
<tr>
<td>price</td>
<td>double</td>

<td>N</td>
<td>买卖时 委托价格</td>
<td>44</td>
</tr>
<tr>
<td>commission</td>
<td>double</td>
<td>N</td>
<td>买卖时 佣金</td>
<td></td>
</tr>
<tr>
<td>stamp</td>
<td>double</td>
<td>N</td>
<td>买卖时 印花税</td>
<td></td>
</tr>
<tr>
<td>currentNodeID</td>
<td>String</td>
<td>Y</td>
<td>该记录所属的交易系统节点 ID 标识 for 分布式</td>
<td></td>
</tr>
<tr>
<td>createdAt</td>
<td>datetime</td>
<td>Y</td>
<td>创建时间</td>
<td></td>
</tr>
</tbody>
</table>
<h3 id="References">References</h3>

http://www.quickfixengine.org/FIX50.html
https://www.onixs.biz/fix-ictionary/5.0.SP2/index.html
《证券交易数据交换协议》- STEP 20050325

<h3 id="TODO">TODO</h3>

符合 FIX5.0 协议的 下单、撤单、拒绝、报文 接口设计
基于 h2database 关系型内存数据库的极简订单管理系统实现
通用订单管理系统抽象
