# 上海单市场ETF交易按参数配置成交设计文档

|  |  |  |  |
| --- | --- | --- | --- |
| **公司名称** | 深圳市金证科技股份有限公司 | **文档编号** |  |
| **文档名称** | 上海单市场ETF交易按参数配置成交设计文档 | **文档版本** | 2.0 |
| **起 草** | 陈俊 | **起草日期** | 2018-08 |
| **审 批** | 何俊 | **审批日期** | 2018-08 |

**修订历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 日期 | 状态 | 修订人 | 摘要 |
| 1.0 | 2018-08 | C | 陈俊 | 初稿 |
| 2.0 | 2018-04 | A | 陈俊 | 细化设计及更新数据库设计 |
|  |  |  |  |  |
|  |  |  |  |  |

状态标识：C – Created A - Added M - Modified D - Deleted

# 成交规则参数配置

对于订单的确认回报和成交回报，可以按照它们这申报订单的对应关系，按配置进行消息的回复。

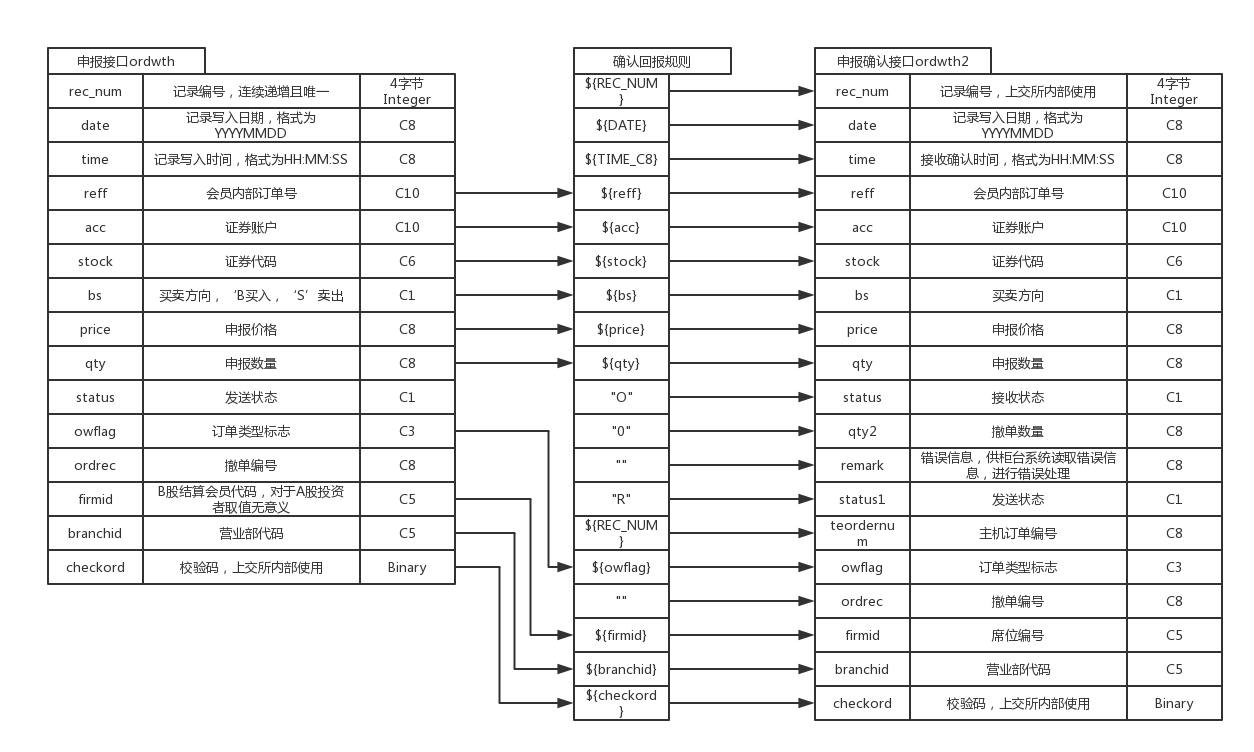


图1.1 确认回报规则

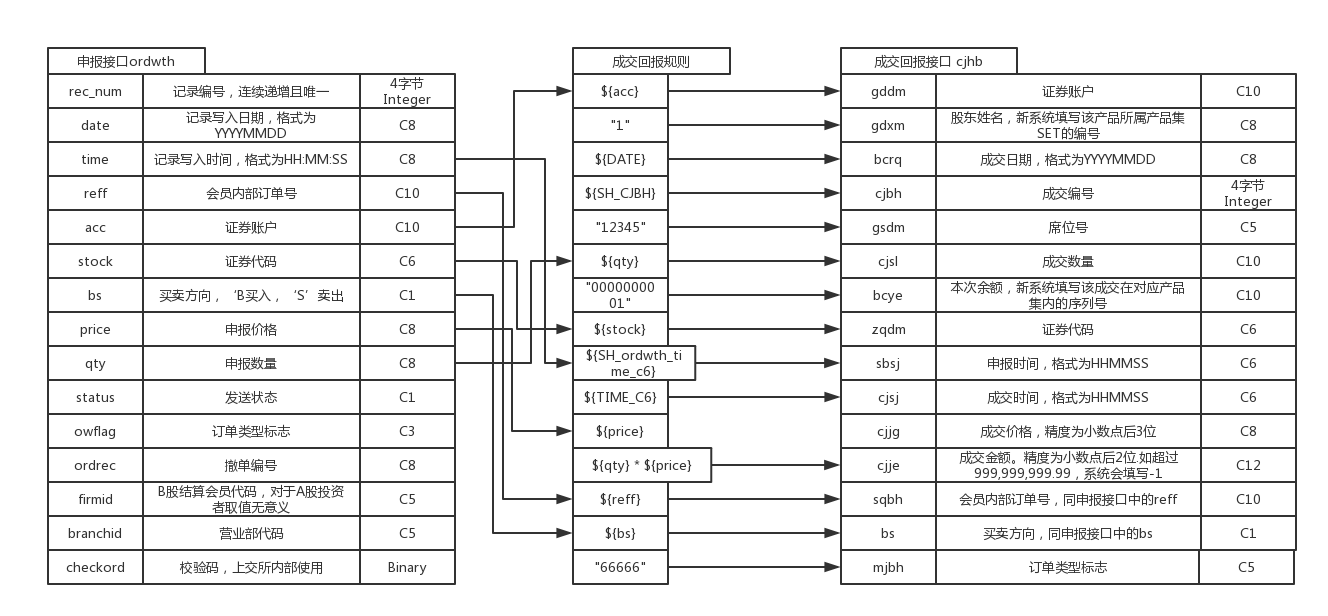


图1.2 成交回报规则

以@标识需要引用原订单的变量：

（1）根据申报订单，rec\_num, date, time, reff, acc, stock, bs, price, qty, status, owflag, ordrec, firmid, branchid, checkord为申报表对应数据。

预设变量，以${?}对预设变量进行包含：

（1）根据订单实际需要，预设通用变量：

${time\_c6}为格式HHMMSS的申报时间

${REC\_NUM}--自增rec\_num

${DATE}--日期，格式为YYYYMMDD.${TIME\_C8}--时间，格式为HH:MM:SS

${TIME\_c6}--时间，格式为HHMMSS。${SH\_CJBH}--自增的成交编号

（2）在表达式内部允许使用四则运算，仅支持+-×/。如果是输出结果是金额的计算，外部引入的单价需转化为厘（小数点后三位）的整型。

转化为配置文件即为：

{  
    **"comment"**:**"${?}为字段替换符。rec\_num,date,time,reff,acc,stock,bs,price,qty,status,owflag,ordrec,firmid,branchid,checkord为申报表对应数据；${time\_c6}为格式HHMMSS的申报时间。${REC\_NUM}--自增rec\_num。${DATE}--日期，格式为YYYYMMDD.${TIME\_C8}--时间，格式为HH:MM:SS。${TIME\_c6}--时间，格式为HHMMSS。${SH\_CJBH}--自增的成交编号。"**,  
    **"judge\_cond"**:{  
        **"stock"**:**"51[0-9]\*"**,  
        **"owflag"**:**"LPT"**,  
        **"price"**:**"\*"**,  
        **"qty"**:**"\*"**  
    },  
    **"confirm"**:{  
        **"rec\_num"**:**"${REC\_NUM}"**,  
        **"date"**:**"${DATE}"**,  
        **"time"**:**"${TIME\_C8}"**,  
        **"reff"**:**"${reff}"**,  
        **"acc"**:**"${acc}"**,  
        **"stock"**:**"${stock}"**,  
        **"bs"**:**"${bs}"**,  
        **"price"**:**"${price}"**,  
        **"qty"**:**"${qty}"**,  
        **"status"**:**"O"**,  
        **"qty2"**:**"0"**,  
        **"remark"**:**""**,  
        **"status1"**:**"R"**,  
        **"teordernum"**:**"${REC\_NUM}"**,  
        **"owflag"**:**"${owflag}"**,  
        **"ordrec"**:**""**,  
        **"firmid"**:**"${firmid}"**,  
        **"branchid"**:**"${branchid}"**,  
        **"checkord"**:**"${checkord}"**  
    },  
    **"match"**:[  
        {  
            **"gddm"**:**"${acc}"**,  
            **"gdxm"**:**"1"**,  
            **"bcrq"**:**"${DATE}"**,  
            **"cjbh"**:**"${SH\_CJBH}"**,  
            **"gsdm"**:**"12345"**,  
            **"cjsl"**:**"${qty}"**,  
            **"bcye"**:**"0000000001"**,  
            **"zqdm"**:**"${stock}"**,  
            **"sbsj"**:**"${time\_c6}"**,  
            **"cjsj"**:**"${TIME\_c6}"**,  
            **"cjjg"**:**"${price}"**,  
            **"cjje"**:**"${qty} \* ${price} / 2"**,  
            **"sqbh"**:**"${reff}"**,  
            **"bs"**:**"${bs}"**,  
            **"mjbh"**:**"66666"**  
        },  
        {  
            **"gddm"**:**"${acc}"**,  
            **"gdxm"**:**"1"**,  
            **"bcrq"**:**"${DATE}"**,  
            **"cjbh"**:**"${SH\_CJBH}"**,  
            **"gsdm"**:**"12345"**,  
            **"cjsl"**:**"${qty}"**,  
            **"bcye"**:**"0000000001"**,  
            **"zqdm"**:**"${stock}"**,  
            **"sbsj"**:**"${time\_c6}"**,  
            **"cjsj"**:**"${TIME\_c6}"**,  
            **"cjjg"**:**"${price}"**,  
            **"cjje"**:**"${qty} \* ${price} / 2"**,  
            **"sqbh"**:**"${reff}"**,  
            **"bs"**:**"${bs}"**,  
            **"mjbh"**:**"66666"**  
        }  
    ]  
}

# 上海单市场ETF按成交规则参数配置架构

在仿真撮合系统接收到上海数据库报盘的ETF交易后，撮合系统根据其股票代码和订单类型（stock = 51\*\*\*\* owflag = LPT），判断其属于上海ETF订单，查找相应的交易规则配置，然后按配置回确认回报和成交回报。（清算文件暂未考虑）。

因为不能光凭股票代码来判断其ETF类型，需要在Web端对ETF类型进行配置。

需要在Web端载入交易所的ETF公告申购赎回清单文件，Web解析后存入数据库，待撮合系统启动时进行载入。

注：ETF代码对应类型

sh db

51\*\*\*\* 的股票代码，

data/stock\_mapping/

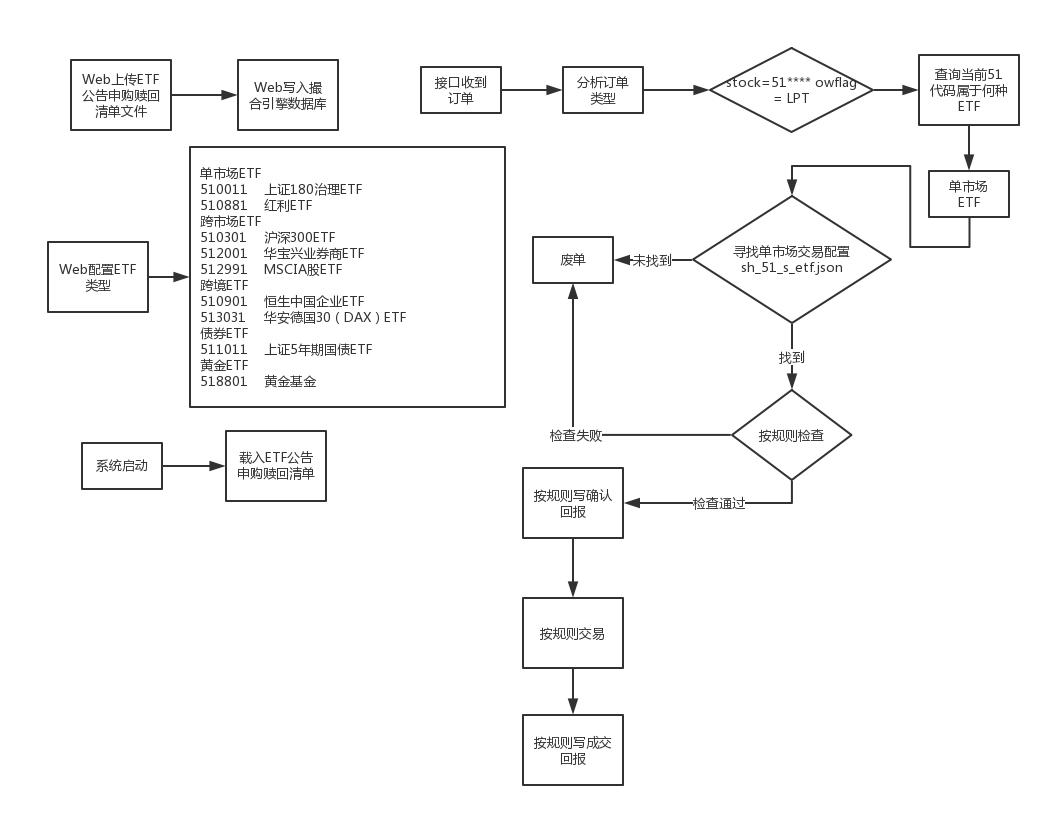
单市场ETF etftype=1

跨市场ETF etftype=2

跨境ETF etftype=3 申赎业务通过综合业务平台接口实现，上海数据库报盘接口不支持

债券ETF etftype=4

黄金ETF etftype=5



# 功能设计

## 3.1 ETF数据储存

### 3.1.1 数据库表设计

以深圳的ETF信息为标准，设计数据库的表，上海的ETF信息经转换后存入数据库。ETF信息表分为etf\_info表（储存ETF信息），eft\_component表（储存ETF成份股）。（注：此信息由Web端解析，下发给撮合引擎储存）。

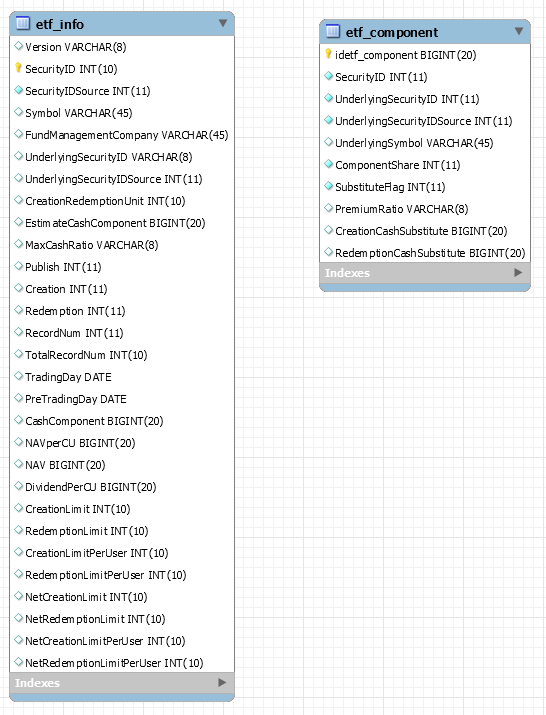


图3.1.1.1 数据库表

建表语句：

CREATE TABLE `etf\_component` (

`idetf\_component` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`SecurityID` int(11) NOT NULL COMMENT '证券代码 SecurityID C8\n-- 对应上海ETF Fundid1',

`UnderlyingSecurityID` int(11) NOT NULL COMMENT '证券代码 UnderlyingSecurityID C8\n-- 对应上海ETF 证券代码',

`UnderlyingSecurityIDSource` int(11) NOT NULL COMMENT '证券代码源 UnderlyingSecurityIDSource C4\n101 = 上海证券交易所\n102 = 深圳证券交易所\n103 = 香港交易所',

`UnderlyingSymbol` varchar(45) COLLATE gbk\_bin DEFAULT NULL COMMENT '证券简称 UnderlyingSymbol C40\n可能包含中文字符，表示最多40个UTF-8 字符\n-- 对应上海ETF 证券简称',

`ComponentShare` int(11) NOT NULL COMMENT '成份证券数 ComponentShare N15(2)\n每个申购篮子中该成份证券的数量\n-- 对应上海ETF 证券数量',

`SubstituteFlag` int(11) NOT NULL COMMENT '现金替代标志 SubstituteFlag C1\n0=禁止现金替代（必须有证券）\n1=可以进行现金替代（先用证券，证券不足时差额部分用现金替代）\n2=必须用现金替代\n-- 对应上海ETF 替代标志\n表示该成份产品是否允许用现金进行替代。\n0表示该证券为沪市证券，禁止现金替代（必须有证券）\n1表示该证券为沪市证券，可以进行现金替代（先用证券，证券不足的话用现金替代）\n2表示该证券为沪市证券，必须用现金替代。\n3表示该证券为深市证券，退补现金替代。\n4表示该证券为深市证券，必须现金替代。\n5表示非沪深市场成分证券退补现金替代。\n6表示非沪深市场成份证券必须现金替代。',

`PremiumRatio` varchar(8) COLLATE gbk\_bin DEFAULT NULL COMMENT '溢价比列 PremiumRatio N7(5)\n证券用现金进行替代的时候，计算价格时增加的比例。例如：2.551％在文件中用0.02551 表示；2.1%在文件中用0.02100 表示。此字段只有现金替代标志为‘1’时才有效\n-- 对应上海ETF 溢价比例',

`CreationCashSubstitute` bigint(20) DEFAULT NULL COMMENT '申购替代金额 CreationCashSubstitute N18(4)\n当某只证券必须用现金替代的时候，申购时该证券所需总金额此字段只有当替代标志为‘2’时才有效，以厘为单位\n-- 对应上海ETF 总金额\n当某只证券必须用现金替代的时候，该证券所需总金额。',

`RedemptionCashSubstitute` bigint(20) DEFAULT NULL COMMENT '赎回替代金额 RedemptionCashSubstitute N18(4)\n当某只证券必须用现金替代的时候，赎回时对应该证券返还的总金额。此字段只有当替代标志为‘2’时才有效，以厘为单位\n例如：2000 在文件中用2000.0000表示。对于跨境ETF、跨市场ETF、黄金ETF 和现金债券ETF，该字段为0.0000。\n-- 对应上海ETF 总金额\n当某只证券必须用现金替代的时候，该证券所需总金额。',

PRIMARY KEY (`idetf\_component`),

UNIQUE KEY `idetf\_component\_UNIQUE` (`idetf\_component`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk COLLATE=gbk\_bin COMMENT='ETF成份股列表 Components，结构采用深圳的信息标准，参考文档《深圳证券交易所数据文件交换接口规范（Ver1.05）.pdf》，《IS101 上海证券交易所竞价撮合平台市场参与者接口规格说明书1.38版\_20180522.docx》';

CREATE TABLE `etf\_info` (

`Version` varchar(8) COLLATE gbk\_bin DEFAULT NULL COMMENT '版本号 Version C8 固定值1.0',

`SecurityID` int(10) unsigned NOT NULL COMMENT '证券代码 SecurityID C8\n-- 对应上海ETF Fundid1',

`SecurityIDSource` int(11) NOT NULL COMMENT '证券代码源\n101=上海证券交易所\n102=深圳证券交易所\n103=香港交易所',

`Symbol` varchar(45) COLLATE gbk\_bin DEFAULT NULL COMMENT '基金名称 Symbol C40 可能包含中文字符，表示最多40\n个UTF-8 字符',

`FundManagementCompany` varchar(45) COLLATE gbk\_bin DEFAULT NULL COMMENT '基金公司名称 FundManagementCompany C30 可能包含中文字符，表示最多30\n个UTF-8 字符',

`UnderlyingSecurityID` varchar(8) COLLATE gbk\_bin DEFAULT NULL COMMENT '拟合指数代码 UnderlyingSecurityID C8',

`UnderlyingSecurityIDSource` int(11) DEFAULT NULL COMMENT '拟合指数代码源 UnderlyingSecurityIDSource C4\n101 = 上海证券交易所\n102 = 深圳证券交易所\n103 = 香港交易所\n9999=其他',

`CreationRedemptionUnit` int(10) unsigned DEFAULT NULL COMMENT '最小申购赎回单位 CreationRedemptionUnit N15(2)\n每个篮子（最小申购赎回单位）对应的ETF 份数，目前只能为正整数\n-- 对应上海ETF CreationRedemptionUnit',

`EstimateCashComponent` bigint(20) DEFAULT NULL COMMENT '预估现金差额 EstimateCashComponent N11(2)\nT 日每个篮子的预估现金差额，以厘为单位\n-- 对应上海ETF EstimateCashComponet',

`MaxCashRatio` varchar(8) COLLATE gbk\_bin DEFAULT NULL COMMENT '最大现金替代比例 MaxCashRatio N6(5)\n最大现金替代比例，例如：5.551％在文件中用0.05551 表示\n-- 对应上海ETF MaxCashRatio',

`Publish` int(11) DEFAULT NULL COMMENT '是否发布IOPV Publish C1\nY=是 -- 1\nN=否 -- 0\n-- 对应上海ETF Publish\n1表示计算IOPV并通过行情发布\n0表示不计算IOPV也无需通过行情发布',

`Creation` int(11) DEFAULT NULL COMMENT '是否允许申购 Creation C1\nY=是 -- 1\nN=否 -- 0\n-- 对应上海ETF CreationRedemption\n申购和赎回允许状态\n0 - 不允许申购/赎回\n1 - 申购和赎回皆允许\n2 - 仅允许申购\n3 - 仅允许赎回',

`Redemption` int(11) DEFAULT NULL COMMENT '是否允许赎回 Redemption C1\nY=是 -- 1\nN=否 -- 0\n-- 对应上海ETF CreationRedemption\n申购和赎回允许状态\n0 - 不允许申购/赎回\n1 - 申购和赎回皆允许\n2 - 仅允许申购\n3 - 仅允许赎回',

`RecordNum` int(11) DEFAULT NULL COMMENT '深市成份证券数目 RecordNum N4\n表示一个篮子中的深市成份证券数目（包含159900 证券）',

`TotalRecordNum` int(10) unsigned DEFAULT NULL COMMENT '所有成份证券数目 TotalRecordNum N4\n表示一个篮子中的所有成份证券数目（包含159900 证券）\n-- 对应上海ETF Recordnum',

`TradingDay` date DEFAULT NULL COMMENT '交易日 TradingDay N8\n格式YYYYMMDD\n-- 对应上海ETF TradingDay',

`PreTradingDay` date DEFAULT NULL COMMENT '前交易日 PreTradingDay N8\nT-X 日日期，格式YYYYMMDD，X 由基金公司根据基金估值时间确定\n-- 对应上海ETF PreTradingDay',

`CashComponent` bigint(20) unsigned DEFAULT NULL COMMENT '现金余额 CashComponent N11(2)\nT-X 日申购赎回基准单位的现金余额，以厘为单位\n-- 对应上海ETF CashComponent',

`NAVperCU` bigint(20) unsigned DEFAULT NULL COMMENT '申购赎回基准单位净值 NAVperCU N12(2)\nT-X 日申购赎回基准单位净值，以厘为单位\n-- 对应上海ETF NAVperCU',

`NAV` bigint(20) unsigned DEFAULT NULL COMMENT '单位净值 NAV N8(4)\nT-X 日基金的单位净值，以厘为单位\n-- 对应上海ETF NAV',

`DividendPerCU` bigint(20) DEFAULT NULL COMMENT '红利金额 DividendPerCU N12(2)\nT 日申购赎回基准单位的红利金额，以厘为单位',

`CreationLimit` int(10) unsigned DEFAULT NULL COMMENT '累计申购总额限制 CreationLimit N18(2)\n当天累计可申购的基金份额上限，为0 表示没有限制，目前只能为整数',

`RedemptionLimit` int(10) unsigned DEFAULT NULL COMMENT '累计赎回总额限制 Redemption Limit N18(2)\n当天累计可赎回的基金份额上限，为0 表示没有限制，目前只能为整数',

`CreationLimitPerUser` int(10) unsigned DEFAULT NULL COMMENT '单个账户累计申购总额限制 CreationLimitPerUser N18(2)\n单个证券账户当天累计可申购的基金份额上限，为0 表示没有限制，目前只能为整数',

`RedemptionLimitPerUser` int(10) unsigned DEFAULT NULL COMMENT '单个账户累计赎回总额限制 RedemptionLimitPerUser N18(2)\n单个证券账户当天累计可赎回的基金份额上限，为0 表示没有限制，目前只能为整数',

`NetCreationLimit` int(10) unsigned DEFAULT NULL COMMENT '净申购总额限制 NetCreationLimit N18(2)\n当天净申购的基金份额上限，为0表示没有限制，目前只能为整数',

`NetRedemptionLimit` int(10) unsigned DEFAULT NULL COMMENT '净赎回总额限制 NetRedemptionLimit N18(2)\n当天净赎回的基金份额上限，为0表示没有限制，目前只能为整数',

`NetCreationLimitPerUser` int(10) unsigned DEFAULT NULL COMMENT '单个账户净申购总额限制 NetCreationLimitPerUser N18(2)\n单个证券账户当天净申购的基金份额上限，为0 表示没有限制，目前只能为整数',

`NetRedemptionLimitPerUser` int(10) unsigned DEFAULT NULL COMMENT '单个账户净赎回总额限制 NetRedemptionLimitPerUser N18(2)\n单个证券账户当天净赎回的基金份额上限，为0 表示没有限制，目前只能为整数',

PRIMARY KEY (`SecurityID`),

UNIQUE KEY `SecurityID\_UNIQUE` (`SecurityID`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk COLLATE=gbk\_bin COMMENT='上海、深圳ETF PCF文件信息，结构采用深圳的信息标准，参考文档《深圳证券交易所数据文件交换接口规范（Ver1.05）.pdf》，《IS101 上海证券交易所竞价撮合平台市场参与者接口规格说明书1.38版\_20180522.docx》';

### 3.1.2 上海ETF信息转存数据库

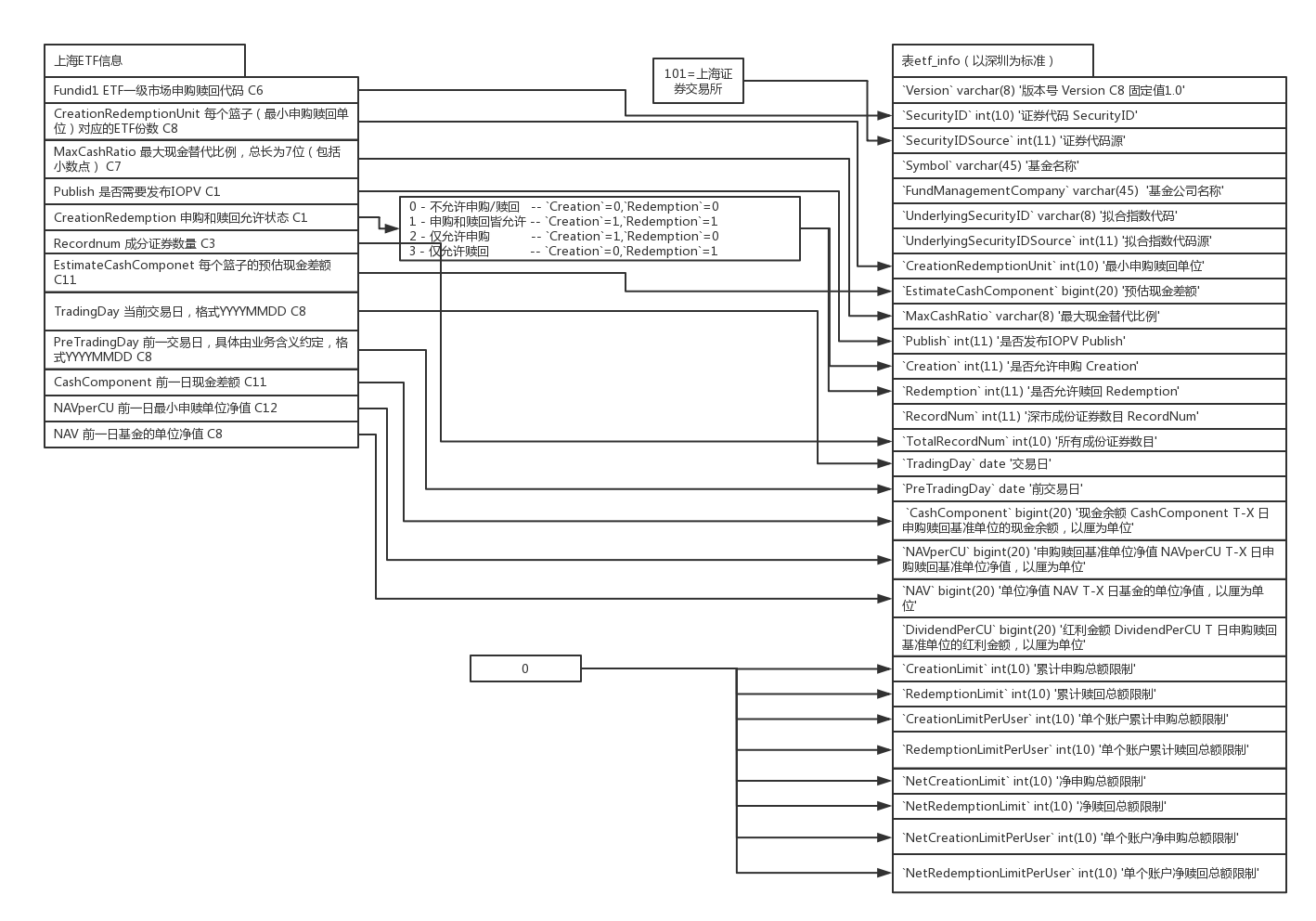


图 3.1.2.1 etf\_info表

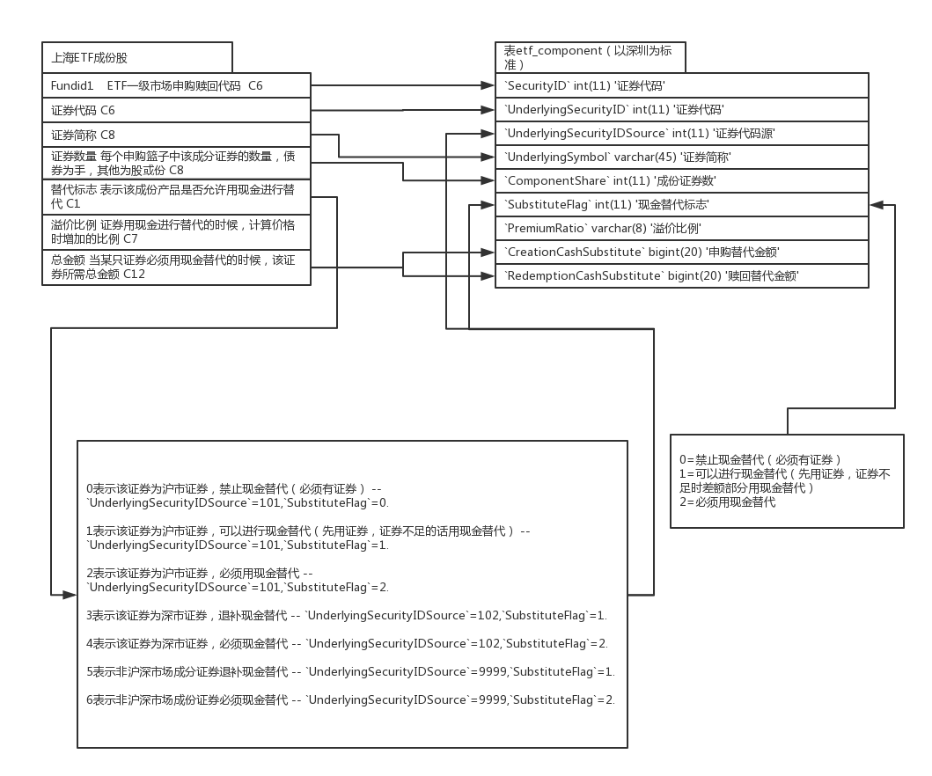


图 3.1.2.2 etf\_component表

### 3.1.3 深圳ETF信息转存数据库

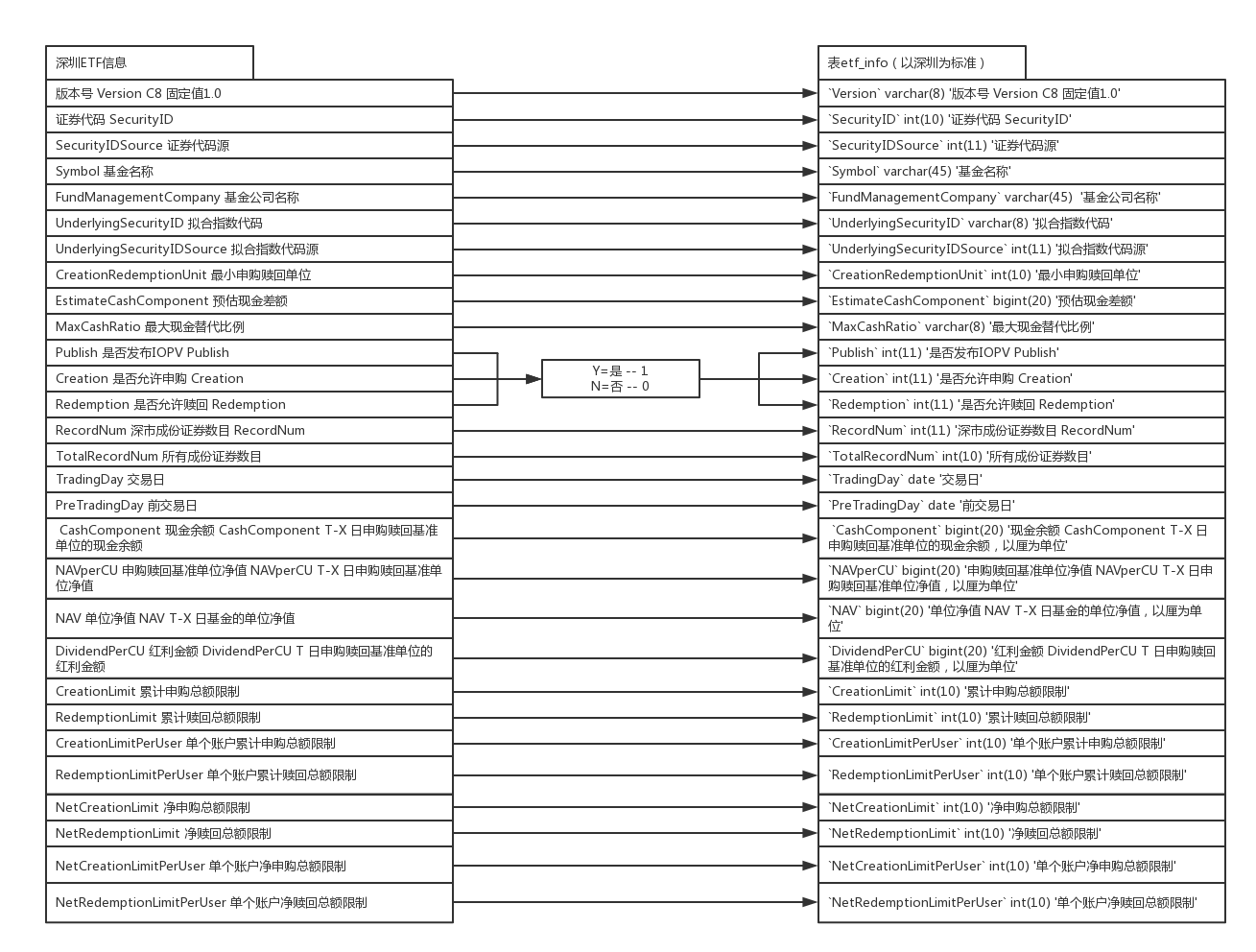


图 3.1.3.1 etf\_info表

## 3.2 上海ETF代码类型

虽然上海ETF代码都是51开头，但是因为历史原因无法直观的通过代码前缀(510、511等)判断ETF类型，需要在数据库中直接储存代码相对应的ETF类型。

单市场ETF etftype=1

跨市场ETF etftype=2

跨境ETF etftype=3 申赎业务通过综合业务平台接口实现，上海数据库报盘接口不支持

债券ETF etftype=4

黄金ETF etftype=5

### 3.2.1 数据库表设计

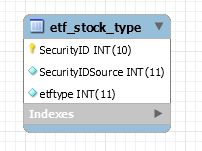


图3.2.1.1 数据库表

建表语句：

CREATE TABLE `etf\_stock\_type` (

`SecurityID` int(10) unsigned NOT NULL COMMENT '证券代码 SecurityID C8\n-- 对应上海ETF Fundid1',

`SecurityIDSource` int(11) NOT NULL COMMENT '证券代码源\n101=上海证券交易所\n102=深圳证券交易所\n103=香港交易所\n9999=其他',

`etftype` int(11) NOT NULL COMMENT '单市场ETF etftype=1\n跨市场ETF etftype=2\n跨境ETF etftype=3 申赎业务通过综合业务平台接口实现，上海数据库报盘接口不支持\n债券ETF etftype=4\n黄金ETF etftype=5',

PRIMARY KEY (`SecurityID`),

UNIQUE KEY `SecurityID\_UNIQUE` (`SecurityID`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk COLLATE=gbk\_bin COMMENT='ETF代码对应的ETF类型';

## 3.3 “成交规则参数配置”配置

上海数据库报盘的成交规则参数配置校验申报数据的stock和owflag字段，按配置中的正则表达式进行匹配，如果两者都匹配上，就按匹配的规则进行成交。（注：此信息由Web端修改，然后下发给在线的撮合引擎）

### 3.3.1 数据库表设计

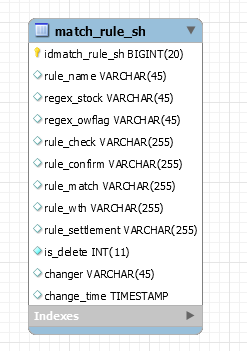


图3.3.1.1 数据库表

建表语句：

CREATE TABLE `match\_rule\_sh` (

`idmatch\_rule\_sh` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`rule\_name` varchar(45) COLLATE gbk\_bin DEFAULT NULL COMMENT '成交规则参数配置别名',

`regex\_stock` varchar(45) COLLATE gbk\_bin DEFAULT NULL COMMENT '识别stock的正则表达式',

`regex\_owflag` varchar(45) COLLATE gbk\_bin DEFAULT NULL COMMENT '识别owflag的正则表达式',

`rule\_check` varchar(255) COLLATE gbk\_bin DEFAULT NULL COMMENT '检查的规则',

`rule\_confirm` varchar(255) COLLATE gbk\_bin DEFAULT NULL COMMENT '确认回报的规则',

`rule\_match` varchar(255) COLLATE gbk\_bin DEFAULT NULL COMMENT '成交回报的规则',

`rule\_wth` varchar(255) COLLATE gbk\_bin DEFAULT NULL COMMENT '撤单回报的规则',

`rule\_settlement` varchar(255) COLLATE gbk\_bin DEFAULT NULL COMMENT '清算文件生成的规则',

`is\_delete` int(11) NOT NULL DEFAULT '0' COMMENT '是否已删除\n1 -- 已删除\n0 -- 未删除',

`changer` varchar(45) COLLATE gbk\_bin DEFAULT NULL COMMENT '更改人',

`change\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '更改时间',

PRIMARY KEY (`idmatch\_rule\_sh`),

UNIQUE KEY `idmatch\_rule\_sh\_UNIQUE` (`idmatch\_rule\_sh`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk COLLATE=gbk\_bin COMMENT='成交规则参数配置';

## 3.4 “成交规则参数配置”验证

考虑到“成交规则参数配置”的匹配与否不直观，需增加一个页面，让用户输入stock和owflag，程序运行后告知客户匹配到哪些规则。

## 3.5 “成交规则参数配置”模板

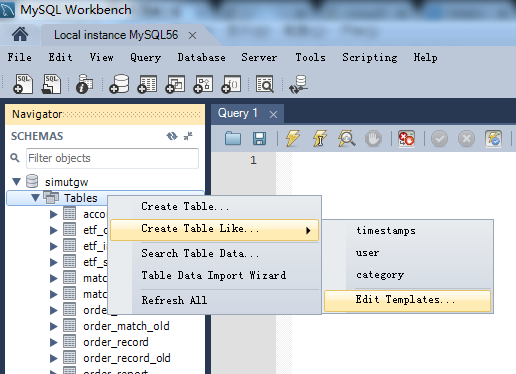
因为“成交规则参数配置”参数多，配置复杂，所以需要配置模板帮助用户可以快速配置：

（1）用户在新建“成交规则参数配置”时可以从模板库按别名选择模板进行新建，选择后在相应模板的各项参数基础上进行修改得到新的配置。

（2）用户可以选择某项“成交规则参数配置”，另存为模板，并设置别名。

模板库的表结构在“成交规则参数配置”上扩展，增加相应所需的字段即可。

例如Mysql中模板的应用：



## 3.6 撮合引擎、交易通道和“成交规则参数配置”

考虑到测试环境与测试配置的差异性，每个撮合引擎有不同的“成交规则参数配置”，各个交易通道也可以指定各自的“成交规则参数配置”（注：此信息由Web端修改，然后下发给在线的撮合引擎）。

### 3.6.1 数据库表设计

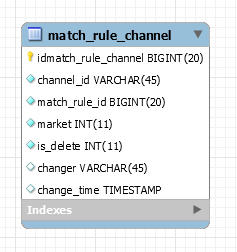


图3.6.1.1 数据库表

建表语句：

CREATE TABLE `match\_rule\_channel` (

`idmatch\_rule\_channel` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT COMMENT '',

`channel\_id` VARCHAR(45) NOT NULL COMMENT '通道ID',

`match\_rule\_id` BIGINT NOT NULL COMMENT '“成交规则参数配置” id',

`market` INT NOT NULL COMMENT '通道市场\n101 = 上海证券交易所\n102 = 深圳证券交易所\n103 = 香港交易所\n9999=其他',

`changer` VARCHAR(45) NULL COMMENT '更改人',

`change\_time` TIMESTAMP NULL DEFAULT now() COMMENT '更改时间',

PRIMARY KEY (`idmatch\_rule\_channel`) COMMENT '')

COMMENT = '通道和“成交规则参数配置”的绑定关系';