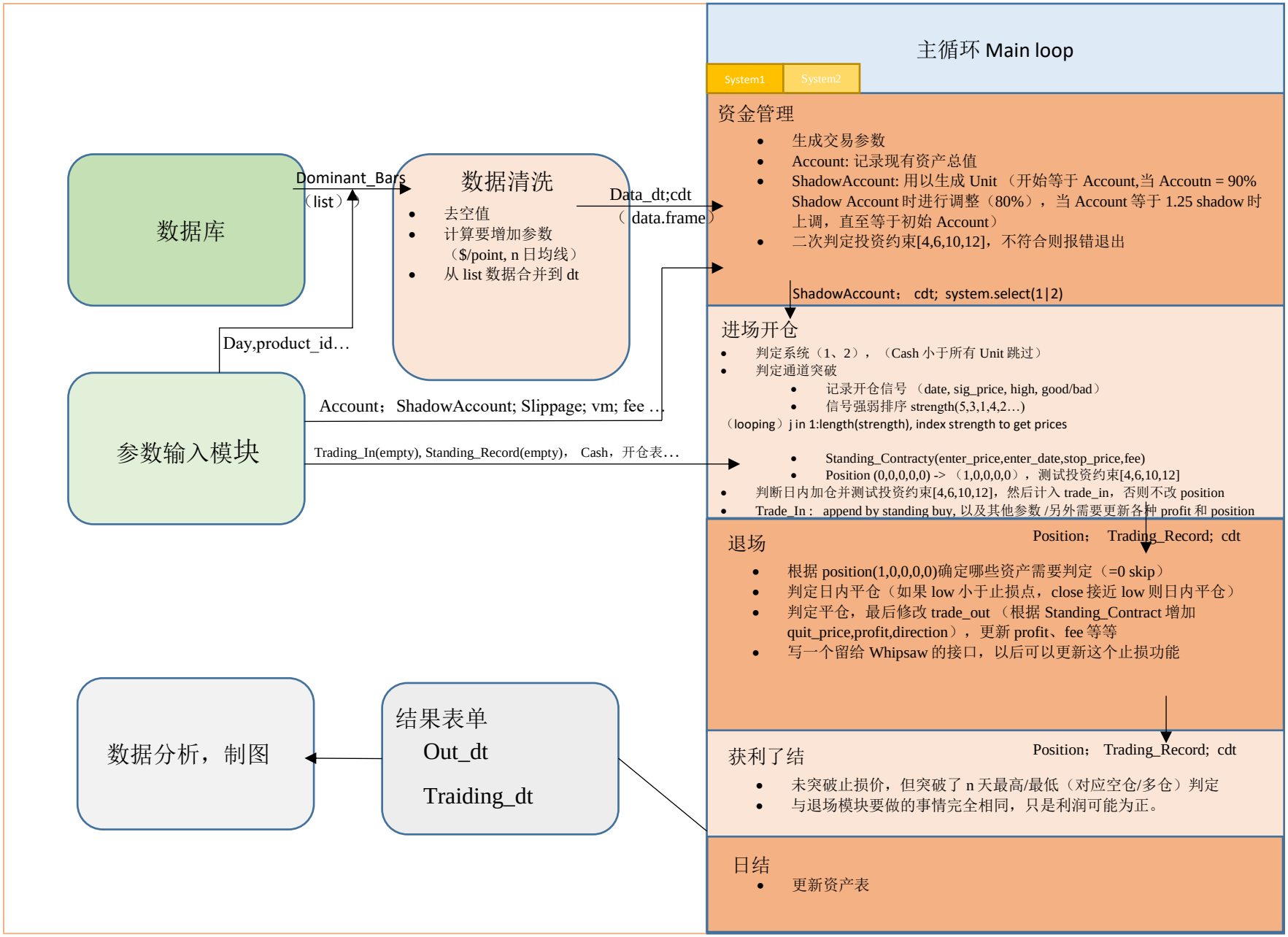
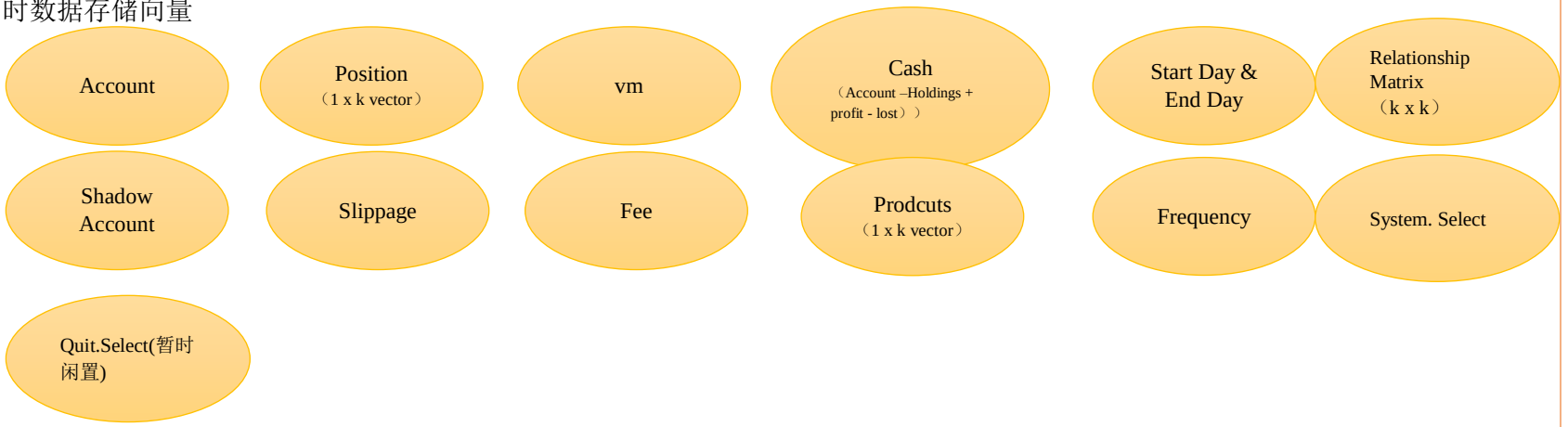
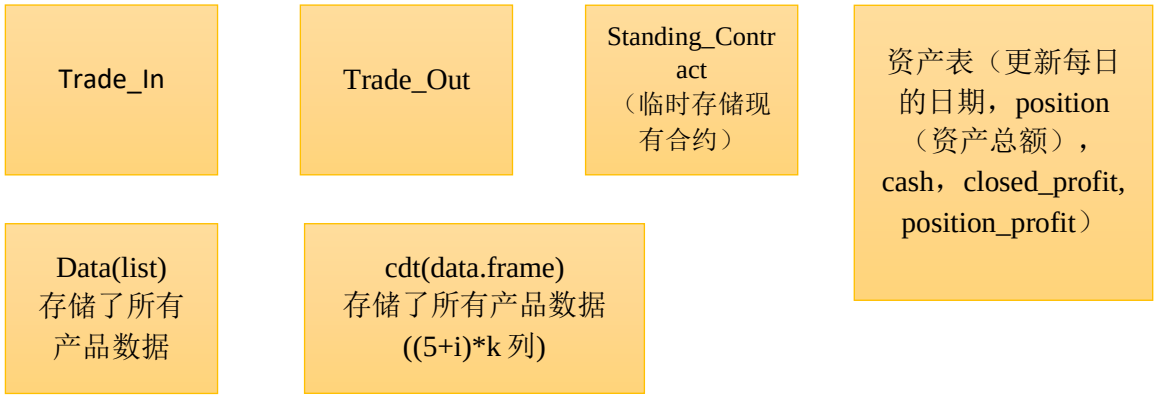

主循环 Main loop
System1
System2
资金管理
生成交易参数
Account: 记录现有资产总值
ShadowAccount: 用以生成 Unit （开始等于 Account,当 Accoutn = 90% Shadow Account 时进行调整（80%），当 Account 等于 1.25 shadow 时上调，直至等于初始 Account）
二次判定投资约束[4,6,10,12]，不符合则报错退出
ShadowAccount； cdt; system.select(1|2)
进场开仓
判定系统（1、2），（Cash 小于所有 Unit 跳过）
判定通道突破
记录开仓信号 （date, sig_price, high, good/bad）
信号强弱排序 strength(5,3,1,4,2…)
（looping）j in 1:length(strength), index strength to get prices
Standing_Contracty(enter_price,enter_date,stop_price,fee)
Position (0,0,0,0,0) -> （1,0,0,0,0），测试投资约束[4,6,10,12]
判断日内加仓并测试投资约束[4,6,10,12]，然后计入 trade_in，否则不改 position
Trade_In： append by standing buy, 以及其他参数 /另外需要更新各种 profit 和 position
退场
Position； Trading_Record; cdt
根据 position(1,0,0,0,0)确定哪些资产需要判定（=0 skip）
判定日内平仓（如果 low 小于止损点，close 接近 low 则日内平仓）
判定平仓，最后修改 trade_out （根据 Standing_Contract 增加 quit_price,profit,direction），更新 profit、fee 等等
写一个留给 Whipsaw 的接口，以后可以更新这个止损功能
获利了结
Position； Trading_Record; cdt
未突破止损价，但突破了 n 天最高/最低（对应空仓/多仓）判定
与退场模块要做的事情完全相同，只是利润可能为正。
日结
更新资产表
数据库
Dominant_Bars
（list）
数据清洗
去空值
计算要增加参数
（$/point, n 日均线）
从 list 数据合并到 dt
Data_dt;cdt
（data.frame）
Day,product_id…
参数输入模块
Account； ShadowAccount; Slippage; vm; fee …
Trading_In(empty), Standing_Record(empty)， Cash，开仓表…
数据分析，制图
结果表单
Out_dt
Traiding_dt

```
matrix(c(1,1,0,1,1,0,0,0,1),3,3,
```

临时数据存储向量




临时数据存储 list