

---

## 7x24 方案

# 1 总体说明

7x24 服务，即全天候提供服务；目前一般用于专指夜间批量处理阶段，保持对客户基本或全方位金融服务。

核心系统在夜间进行业务批量处理的时候，如计提结息需求，报表需求等，这些业务批量处理需要按账户当日日终余额（或其他数据）进行计算，故需要保持这些数据的一定静止状态，而夜间联机交易需要更新账户余额，在没有 7×24 实现机制前，银行都需要在批量运行时间段停止夜间的联机交易，而在批量基本运行结束后再次开始联机交易的对外服务。

一般批量处理与联机处理的冲突区就在账户余额，解决批量用账户日终余额与联机用账户实时余额的存储与使用问题，即可很大程度上实现 7×24 业务服务。

实现 7x24 服务，最关键的要点在于保证两份数据的准确并存：

- A. 动态实时数据（实时余额）：主要是动账及日间查询交易使用
- B. 日切点的静态数据（上日余额）：主要用于批处理：比如计提、结息、总分核对、向外围（尤其是财管、管会）供数等。

目前各厂商主要使用的方案有以下几种：

- A. 单表双余额，国内典型的老联想系系统，如神码、繁德
- B. 双表（双表又分两种：临时表为分户临时表或是流水临时表），典型为中联及大部分国外系统（神码一开始引进的国外系统也是这种方式）

需要考虑的逻辑主要有：

1. 联机交易如何更新余额

- 
2. 日终交易如何获取余额
  3. 账户实时余额的获取（如果临时表是流水临时表，余额需要通过分户账余额与流水临时表汇总计算取得）。
  4. 冲正，必须支持跨日/跨年的冲正

下面将一一说明：

## 2 方案设计

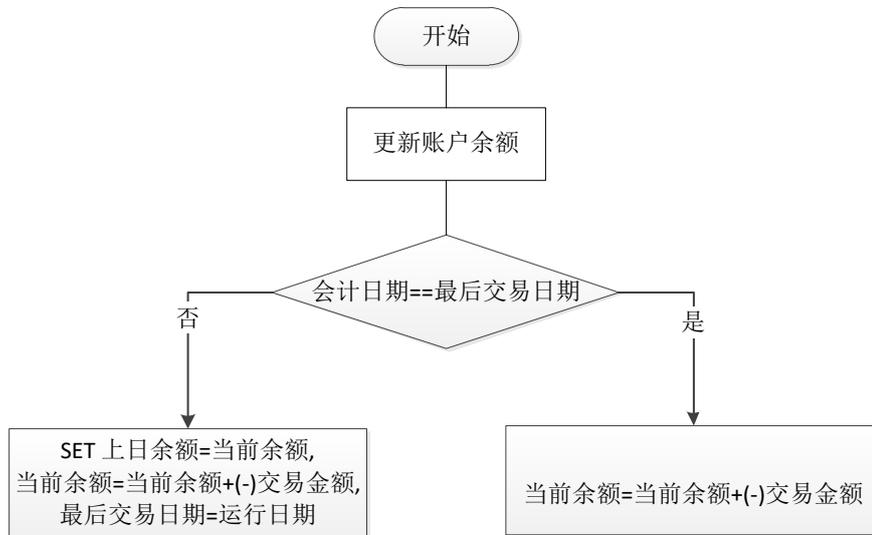
### 2.1 双余额动账更新

#### 2.1.1 总体说明

分户账上设置余额(ACCTUAL\_BAL)、上日余额(PREV\_DAY\_BAL)、最后交易日期(LAST\_TRAN\_DATE)。根据以上字段来实现当前余额、上日余额的读取和更新。

仅在动账交易发生时才可能更新上日余额，即如果该账户长期无动账，在此期间将不用更新上日余额（其实此时的“上日余额”字段从名称上来看与实际是不符的）

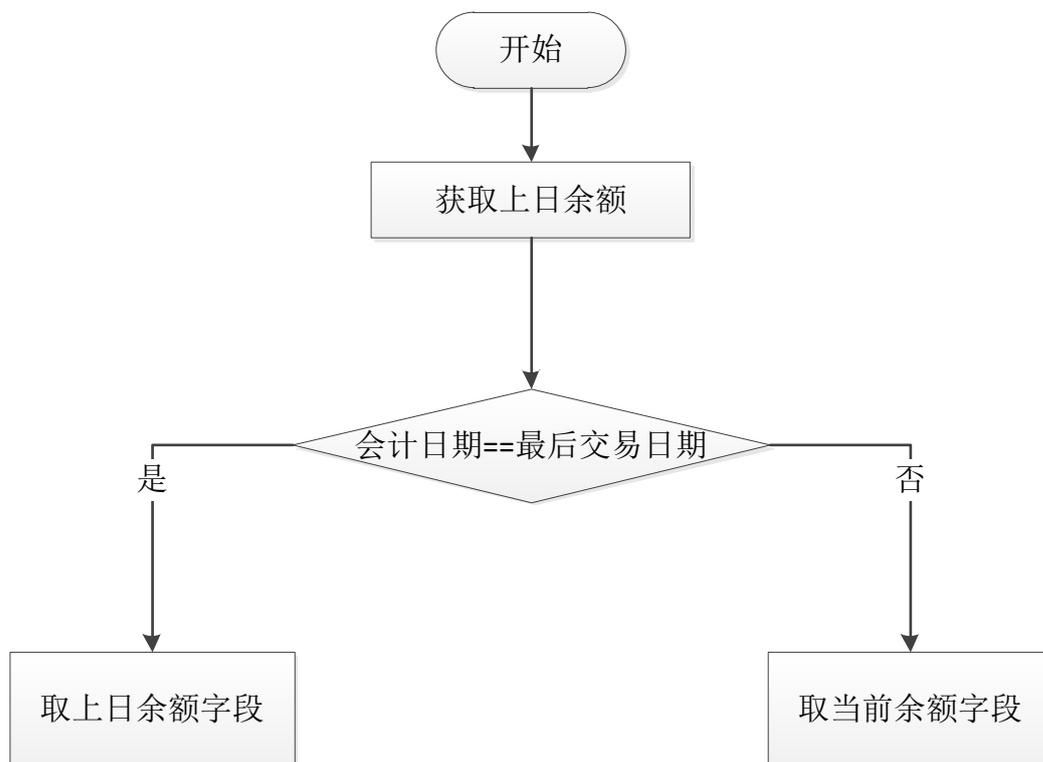
## 2.1.2 动账处理



如上图所示，当日第一笔交易更新上日余额、最后交易日期。

---

### 2.1.3 获取上日余额处理



## 2.2 双余额每日更新

### 2.2.1 总体说明

分户账上设置余额(ACCTUAL\_BAL)、上日余额(PREV\_DAY\_BAL)、最后交易日期(LAST\_TRAN\_DATE)。根据以上字段来实现当前余额、上日余额的读取和更新。

与方案一不同的是，系统每天都会更新上日余额及最后交易日期。

其实此时的“最后交易日期”字段从名称来看与实际不一定相符。

## 2.2.2 日终批量刷新上日余额

取上日余额的场景都在日终，因此在日终切日后一开始就直接批量刷新上日余额，便于后续读取及供数。为避免长时间锁表，该批量任务逐笔处理。

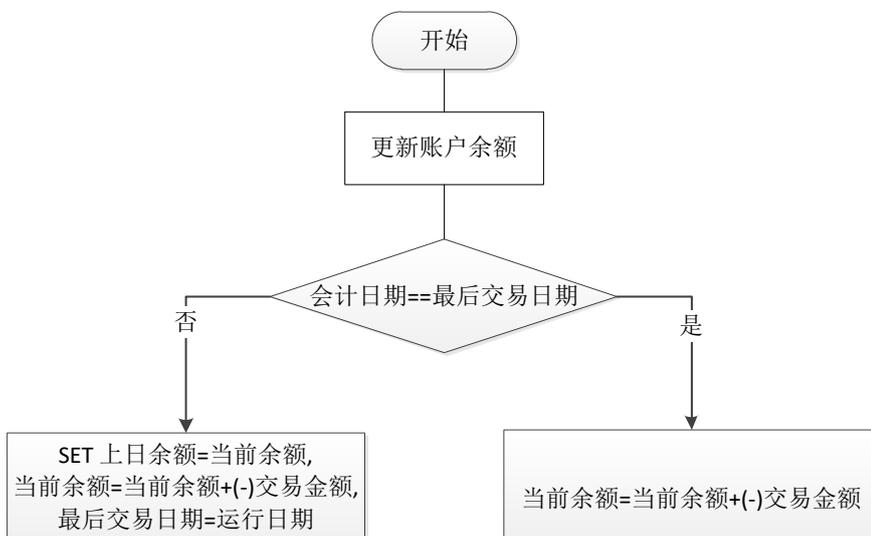
对于一笔分户账，

IF 最后交易日期 < 会计日期

UPDATE 分户账 SET 上日余额=当前余额, 最后交易日期=会计日期

## 2.2.3 动账处理

切日后，日终批量刷新需要一段时间，为确保在此期间的联机交易正常对外服务，动账时仍采用以下处理。



---

如上图所示，当日第一笔交易更新上日余额、最后交易日期。

联机交易与日终批量更新上日余额有极小的可能会出现冲突（同时更新同一账户）。如果发生,解决如下：

- A. 如果批量锁表，联机失败，交易重做将成功。
- B. 如果联机锁表，批量失败，批量重新从断点重跑。

## 2.2.4 获取上日余额处理

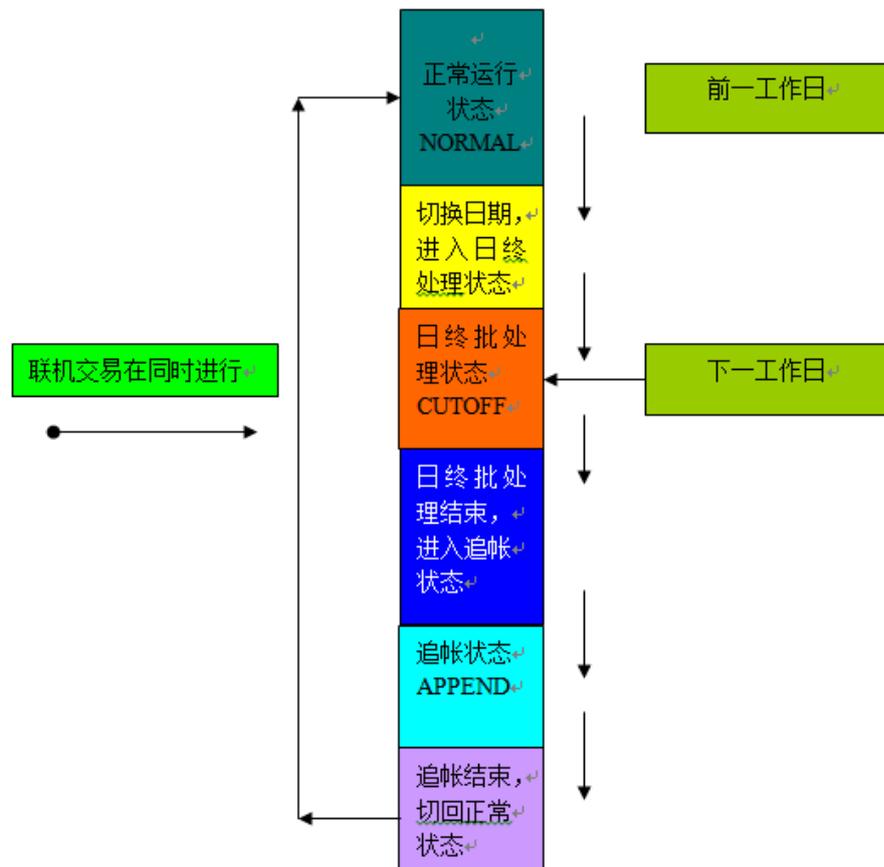
取上日余额的情景都在日终刷新之后，因此此时取上日余额直接取分户账中的上日余额。

## 2.1 双表

### 2.1.1 总体说明

此种方案主要通过系统状态来控制，系统状态分为：

- N：日间运行状态(normal)
- C：日切运行状态(cutoff)
- A：追帐运行状态(append)



数据库表设计上，除分户账外，新增一张影子分户表。

系统做完日终批量处理后，状为日间运行状态，此时所有的交易实时修改分户帐的余额。在日切状态下的所有交易，不修改分户帐的余额，其发生额写入影子分户。保持分户帐余额不变，用于总分平衡检查等日终取上日余额。在追帐状

---

态下的所有交易，实时修改分户帐余额。日终处理的追帐交易，根据影子分户里帐户的发生额进行分户帐余额的修改。在所有分户追帐完成之后，系统状态改为日间运行状态。

因此，总结如下：

- 正常状态或追账状态，实时修改分户账余额。日终批处理状态时，不动分户账余额，只能登记影子分户。不确定的是追账是到底允不允许联机动账交易。如果单笔追账跟同时时间段的联机交易也有小概率冲突，与方案二类似。
- 系统只要不是运行在 N 状态（正常状态），计算帐户可用余额时，必须将影子帐户中的余额和分户帐中的余额一起合并计算。

### 3 方案比较

	优点	缺点
方案一：双余额动账更新	避免追账，尤其是晚间交易量大的银行	表结构设计较为复杂，日终或供数逻辑判断相对复杂。特别是外围取数系统不太能接受。
方案二：双余额每日更新	避免追账 取数逻辑较为清晰	表结构设计较为复杂。
方案三：双表	表结构设计较为简单。 日终取数逻辑较为清晰	追账可能影响较大 计算可用余额的逻辑相对复杂。

---