

大数据量-大量标签组合指标-实时计算方案



? 日统计指标系统一般采用预计算方式



统计指标从大量明细数据汇总。传统技术实时计算无法达到性能要求，只能预先算好结果供查询使用。

? 以银行为例：日统计指标计算需求

友乾营

过滤条件

日期
币种
机构
...

维度

年龄
客户类型
学历
...

统计

日汇总
同比环比
去年年末
...

并发

用户几千人
集中时间
查询指标

中小银行指标计算体系要实现过滤、维度分组、各种统计计算。查询指标的用户也有几千个，需要较大的并发能力。

? 参与计算的维度标签较多，以贷款为例

友乾营

五级分类

一级贷款
二级贷款
...

四类担保

担保A类
担保B类
...

十种客户

甲种客户
乙种客户
...

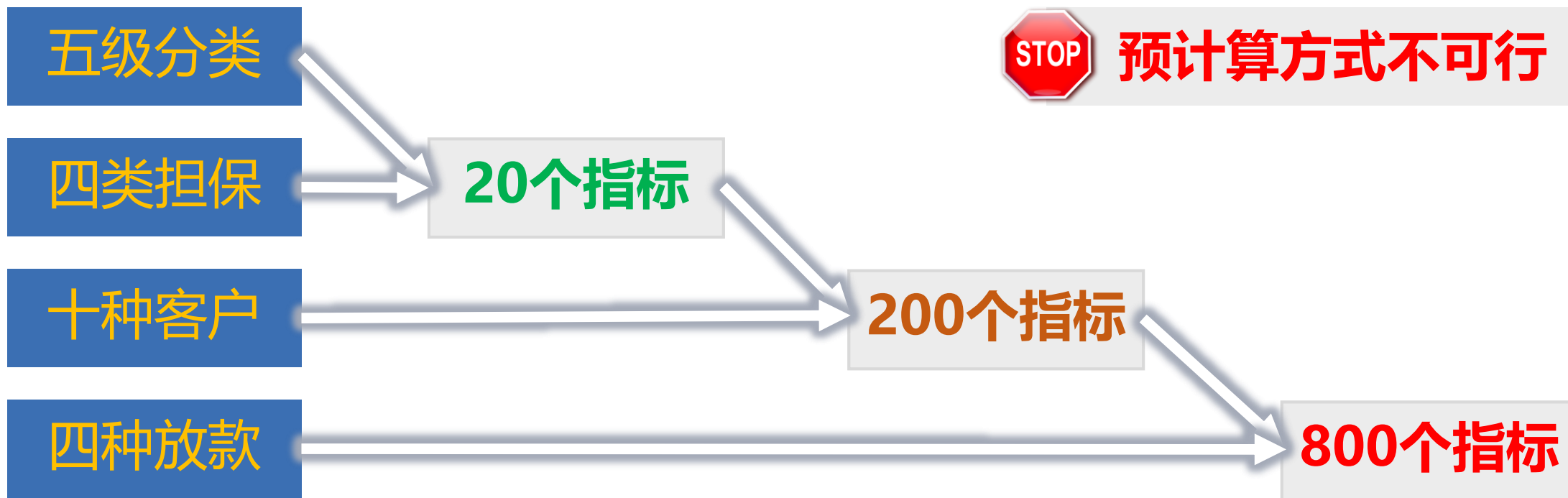
四种放款

壹种放款
贰种放款
...

这里只列举了四种，实际上还有很多种标签。

? 多种维度标签排列组合，会造成指标泛滥

友乾营



贷款四个维度就形成800个指标，还有存款和其他维度。加上日期维度，组合出来的指标太多，预计算方式的结果太多，系统无法承受！

? 以银行为例：日统计指标对应明细数据量很大



实时计算可行吗

存款

每日千万条
十几个字段
保留一年

贷款

每日百万条
几百个字段
保留一年

客户

几百万个
每日存全量
保留一年

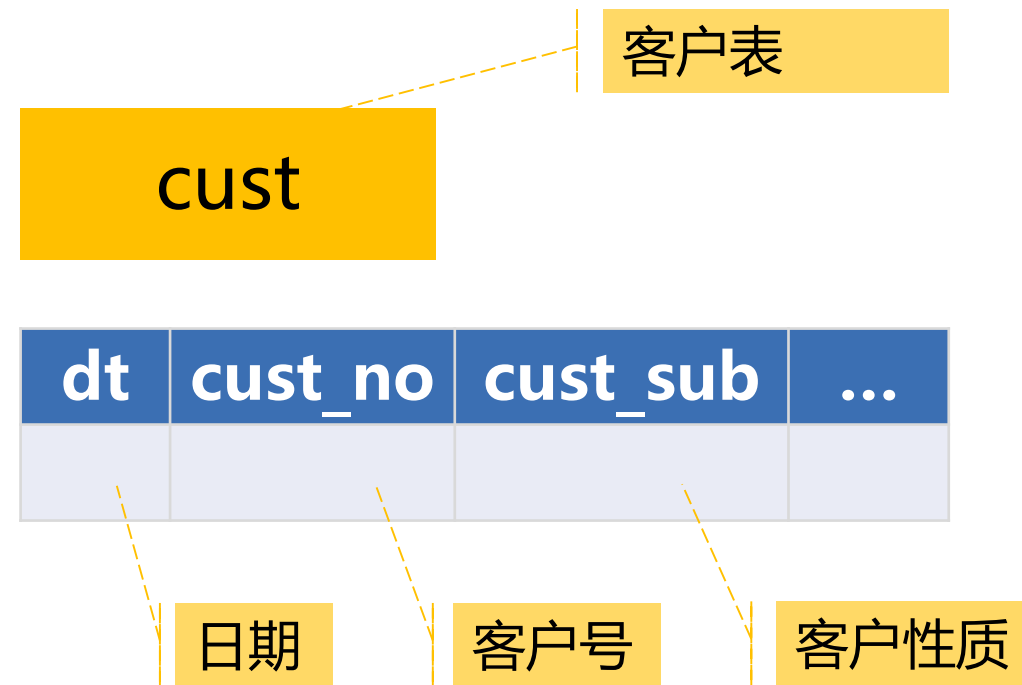
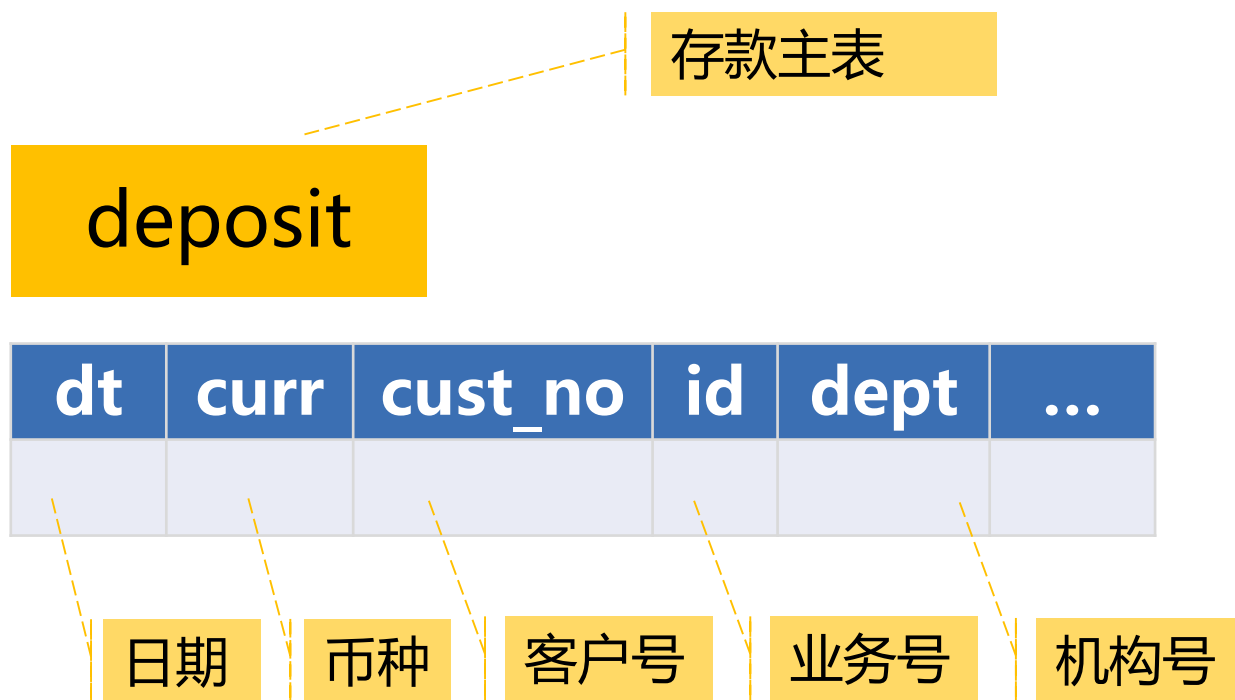
其他

员工千人
机构千个
...

银行指标计算体系涉及的明细数据量非常巨大，要保证指标查询的秒级响应速度是个挑战。

? SQL方案：存款指标

友乾营



存款主表和客户表通过字段日期dt、客户号cust_no关联。存款主表是窄表，十几个字段，每天两千万条记录。每个表存一个月，六亿条数据。

? SQL方案：某日人民币存款的非农业去重客户数

```
select count(distinct c.cust_no)
  from deposit d inner join cust c
    on d.cust_no=c.cust_no and d.dt=c.dt
 where d.dt=to_date('20191001','yyyymmdd')
    and d.curr=1 and c.cust_sub=1
    and d.id<>0
```

计算“某日人民币存款的非农业去重客户数”。几亿数据量的存款主表和几千万的客户表关联，很难达到秒级响应速度。

? SQL方案：贷款指标

友乾营



贷款主表和客户表通过字段日期dt、客户号cust_no关联。贷款主表是宽表，几十个字段，每天三百万条记录。每个表存一个月，九千万条数据。

? SQL方案：某日人民币贷款的农业户满足条件的总金额

友乾营

```
select sum(amt1)
  from loan l
       inner join cust c
         on l.cust_no=c.cust_no and l.dt=c.dt
 where l.dt=to_date('20191001','yyyymmdd')
       and l.curr=1 and c.cust_sub=0 and l.id<>0
       and code11=1 and code12=2
```

计算“某日人民币贷款的农业客户满足code11和code12条件的总金额”。几十字段、千万数据贷款宽表和客户表关联，也难达到秒级响应。

解决办法

改变存储和算法



友乾营

专注数据技术的社群



内存

随机存取快
容量有限



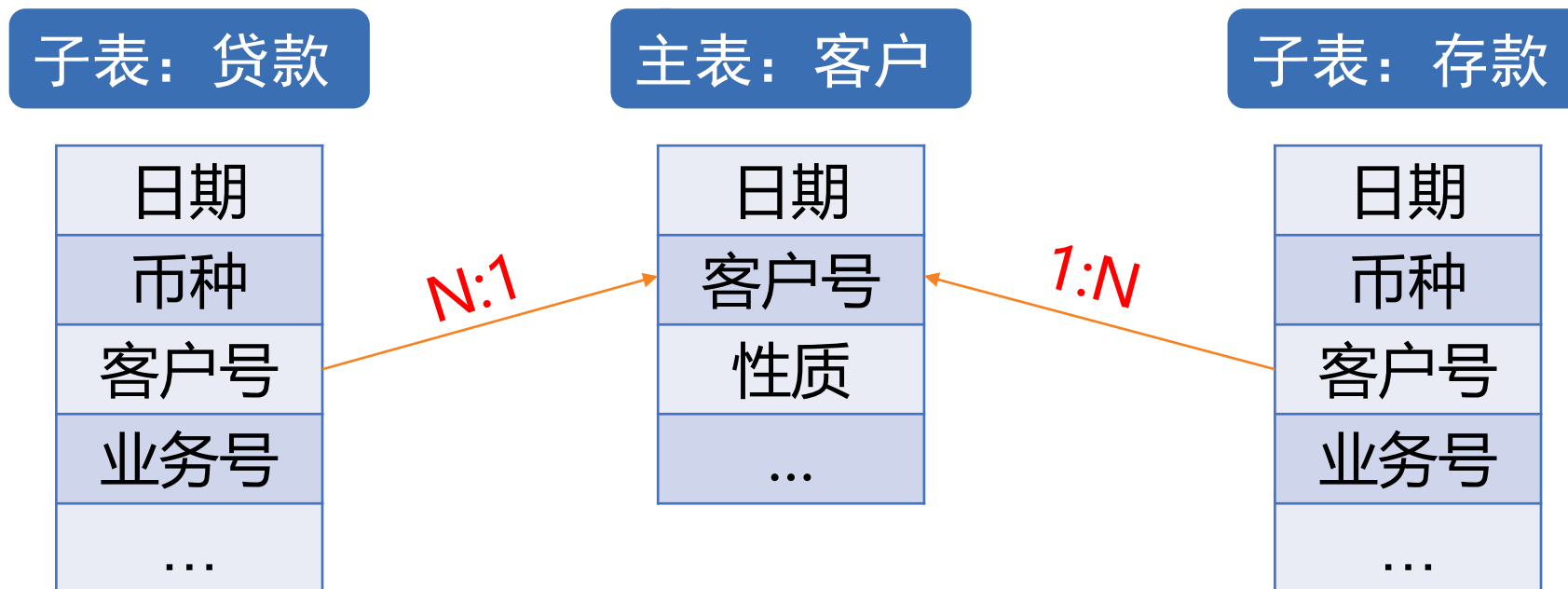
硬盘

读写较慢
容量大

贷款、存款数据随着时间增长很快，一年就有几十亿条记录，要放到硬盘存储。客户表每天保存全量，也要硬盘存储。员工、机构等可以内存。

★ 解题思路-计算特征分析：大表遍历

友乾营



在一天内，指定币种，客户和存款、贷款都是一对多的主子表关系。日统计指标计算本质上是大表关联后，遍历汇总：例如客户数、总金额等。

贷款组表-列存 

dt	cust_no	amt1	...
20191011	1	2345.8	...
20191011	1	100.0	...
20191011	2	1090.0	...
...	...	...	...

贷款组表-行存

dt	cust_no	amt1	...
20191011	1	2345.8	...
20191011	1	100.0	...
20191011	2	1090.0	...
...	...	...	...

存款、贷款、客户数据有十几个到几十个字段，计算每个指标只需要几个字段。采用列存数据， 能根据遍历运算的字段个数， 减少读取数据量。

★ 解题思路-有序列存-有利于压缩

友乾营

贷款组表-有序存储

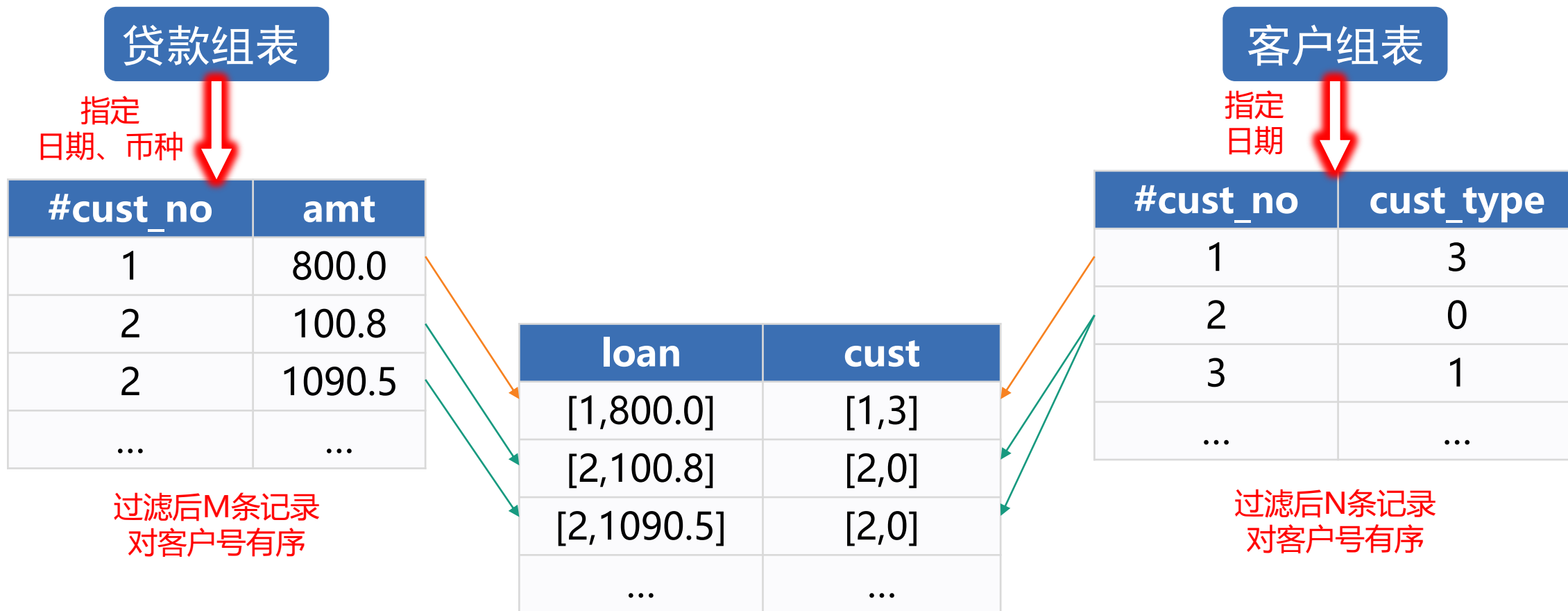
#dt	#curr	#cust_no	amt
20191001	1	1	800.0
20191001	1	2	100.8
20191001	1	2	1090.5
...	...	...	...
20191001	2	1	50.0
20191001	2	6	700.0
...	...	...	...

币种、客户号等采用整形存储，减少存储量，提高计算的速度。

贷款组表，预先按日期、币种、客户有序存储。有序列存有利于重复数据的压缩。按照日期、币种过滤的时候，可以快速跳过不需要的记录。

★ 解题思路-有序列存-有序归并

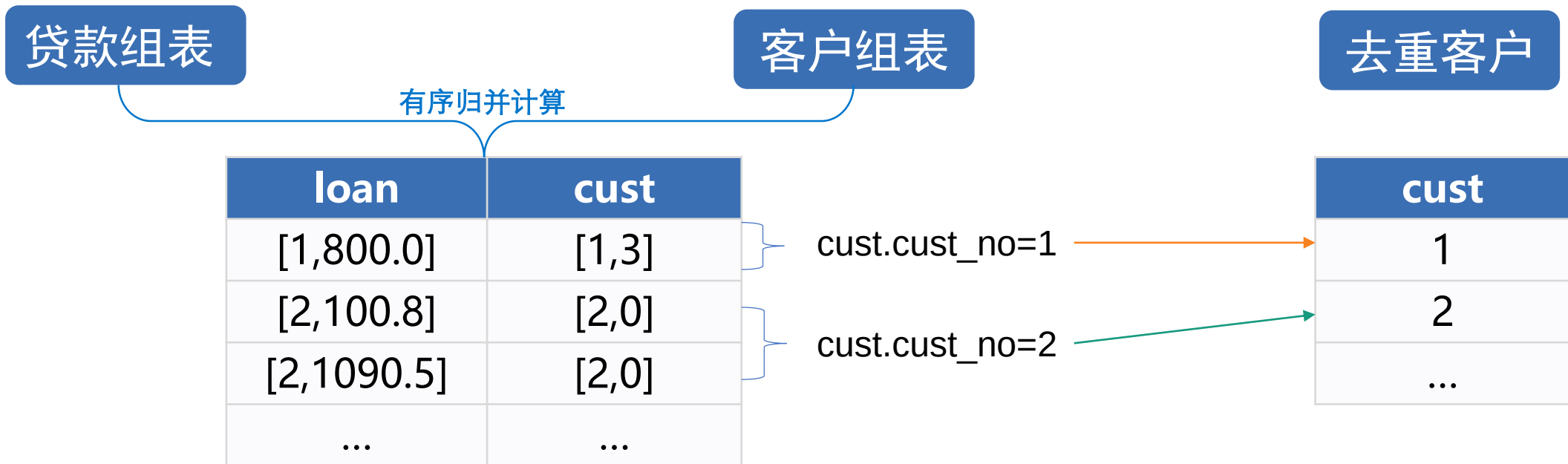
友乾营



主子表都按照客户号有序存储，可以用归并算法一次遍历实现JOIN，复杂度M+N，比外存分段HASH JOIN的复杂度SUM(Ni*Mi)低很多！

★ 解题思路-有序列存-快速去重

友乾营



主子表有序归并的结果，对客户号依然有序。对客户去重计数的时候，只需要和相邻的客户号比较即可。

代码示例

原理很烧脑

代码很简洁



友乾营

专注数据技术的社群

	A
1	<code>=connect("db").cursor("select dt,curr,cust_no,dept... from deposit order by dt,curr,cust_no")</code>
2	<code>=file("deposit.ctx")</code>
3	<code>=A2.create(#dt,#curr,#cust_no,dept,...).append(A1)</code>

A1: 建立数据库连接和游标, 准备有序读取源数据。

A2: 新建一个文件对象, 规定数据有序字段。

A3: 新建列存组表文件, 加入A1游标中的数据。

注: 客户、贷款主表都与存款的代码近似。

读取源数据生成行存组表都仅需两三个函数, 代码非常简洁。组表中的数据可以来自于各种数据源, 本例中是来自于生产数据库。

★ 某日人民币存款-去重农业客户数-输入参数、组表对象

友乾营

	A	B
1	>in_date=20191001	>in_curr=1
2	=file("data/ctx/cust.ctx"). create ()	=file("data/ctx/deposit.ctx"). create()

A1: 输入参数, 日期为20191001。

B1: 输入参数, 币种为1。

A2: 建立客户组表对象。

B2: 建立存款组表对象。

输入参数由前端查询界面传递给集算器, 实际代码中日期和币种改为网格参数即可。

★ 某日存款去重农业客户数-建立游标，定义过滤条件

友乾营

	A	B
3	=A2.cursor(cust_no;dt==in_date && cust_sub==0)	
4	=B2.cursor(cust_no;dt==in_date && curr==in_curr &&id!=0)	

A3：建立客户组表游标，取客户号，过滤条件是日期和客户性质为0（农业户）。

A4：建立存款组表游标，取客户号，过滤条件是日期、币种为1（人民币）且业务号不为零。

A3、A4建立的游标是延迟游标，要取的字段和定义的过滤条件并没有在这里立即执行。条件过滤后的两个游标，都对客户号有序。

★ 某日存款去重农业客户数-归并、分组、计数

	A	B
5	=joinx(A3:cust,cust_no;A4:deposit,cust_no)	
6	=A5.group@1(cust.cust_no)	
7	return A6.skip()	

A5: 客户组表游标与存款游标有序归并, 注意还是没有立即计算。

A6: 归并之后的游标仍然对照客户号有序分组, 只比较相邻的数据分组, 取每组第一个 cust_no 生成新游标, 注意这里仍然没有立即计算。

A7: 对延迟游标计数, 执行之前定义好的所有计算。

B7对延迟游标计数, 一次遍历执行了之前定义的过滤、有序归并和有序分组计算。对硬盘上的文件只需要读取一次。

★ 某日存款去重农业客户数-分组、归并、计数-更快的算法

友乾营

	A	B
5	=A4.group@1(cust_no)	
6	=joinx(A3:cust,cust_no;A5:deposit,cust_no)	
7	return A6.skip()	

A5：存款游标有序有序分组，只比较相邻的数据分组，且取每组第一条，注意没有立即计算。

A6：客户组表游标与分组之后的游标按照客户号有序归并，注意这里仍然没有立即计算。

A7：对延迟游标计数，执行之前定义好的所有计算。

计算顺序改为先分组在归并，减少了存款游标的数据量，性能更快。优化计算顺序是性能提升的重要考虑方向。

★ 某日贷款金额求和-分组、归并、求和

	A	B
5	=joinx(A3:cust,cust_no;A4:loan,cust_no)	
6	=A5.groups(;sum(loan.amt))	
7	return A6	

贷款前半部分和存款基本一致。

A5：客户组表游标与贷款游标有序归并，注意还是没有立即计算。

A6：对延迟游标汇总求和，执行之前定义好的所有计算。

B6对延迟游标汇总求和，一次遍历执行了之前定义的过滤、有序归并和汇总求和计算。对硬盘上的文件只需要读取一次。

效果实测

性能测试



友乾营

专注数据技术的社群

★ 模拟真实环境，大数据量实测

友乾营



6亿存款

一个月明细
每天2千万



9千万贷款

一个月明细
每天3百万



并发访问

12并发，每
并发随即过
滤条件



虚拟服务器

CPU1.7G,
12核,
16G内存

★ 同样的服务器，对比测试

友乾营

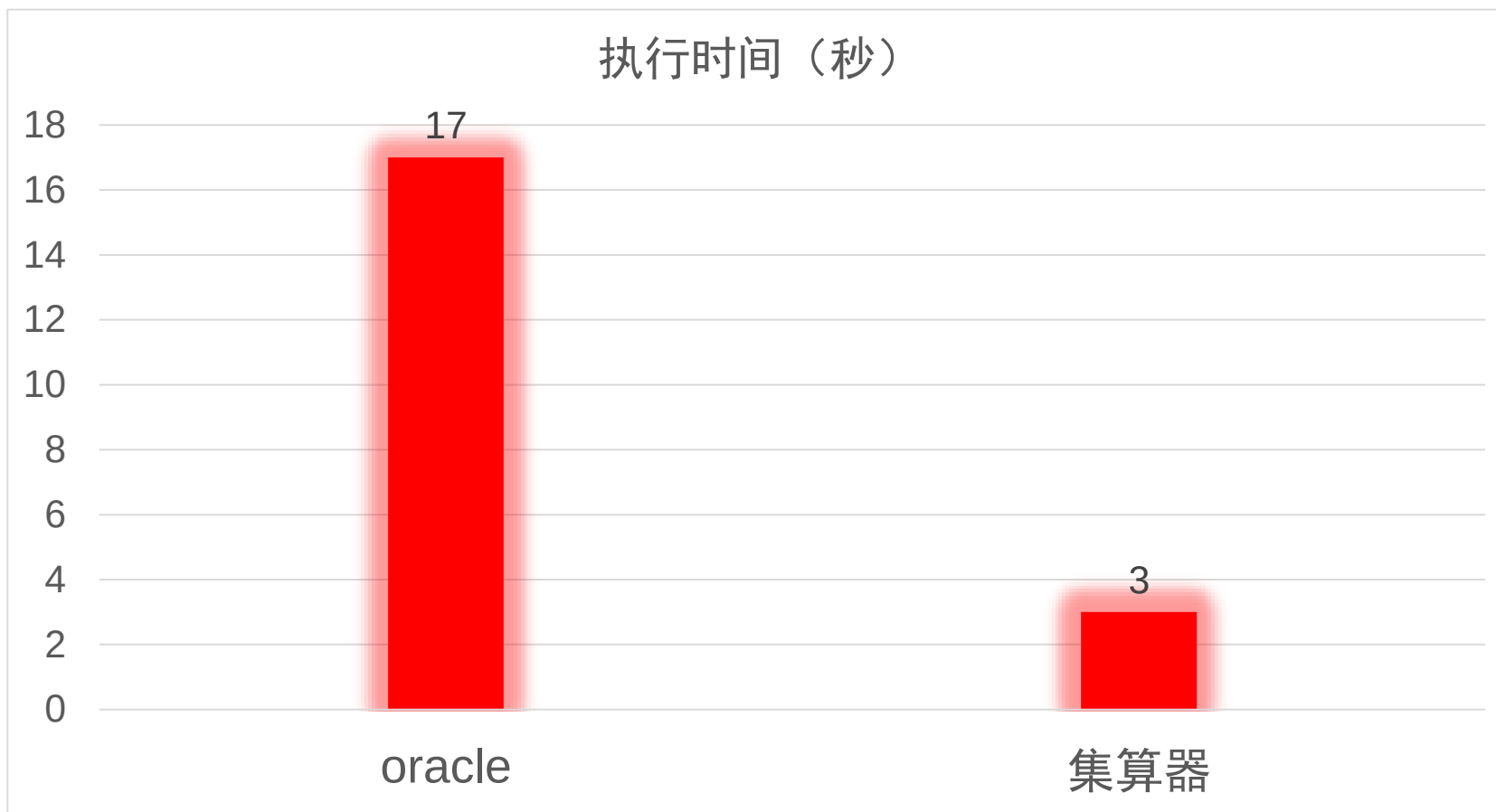


VS

ORACLE®
DATABASE

★ 某日人民币存款去重农业客户数-窄表测试

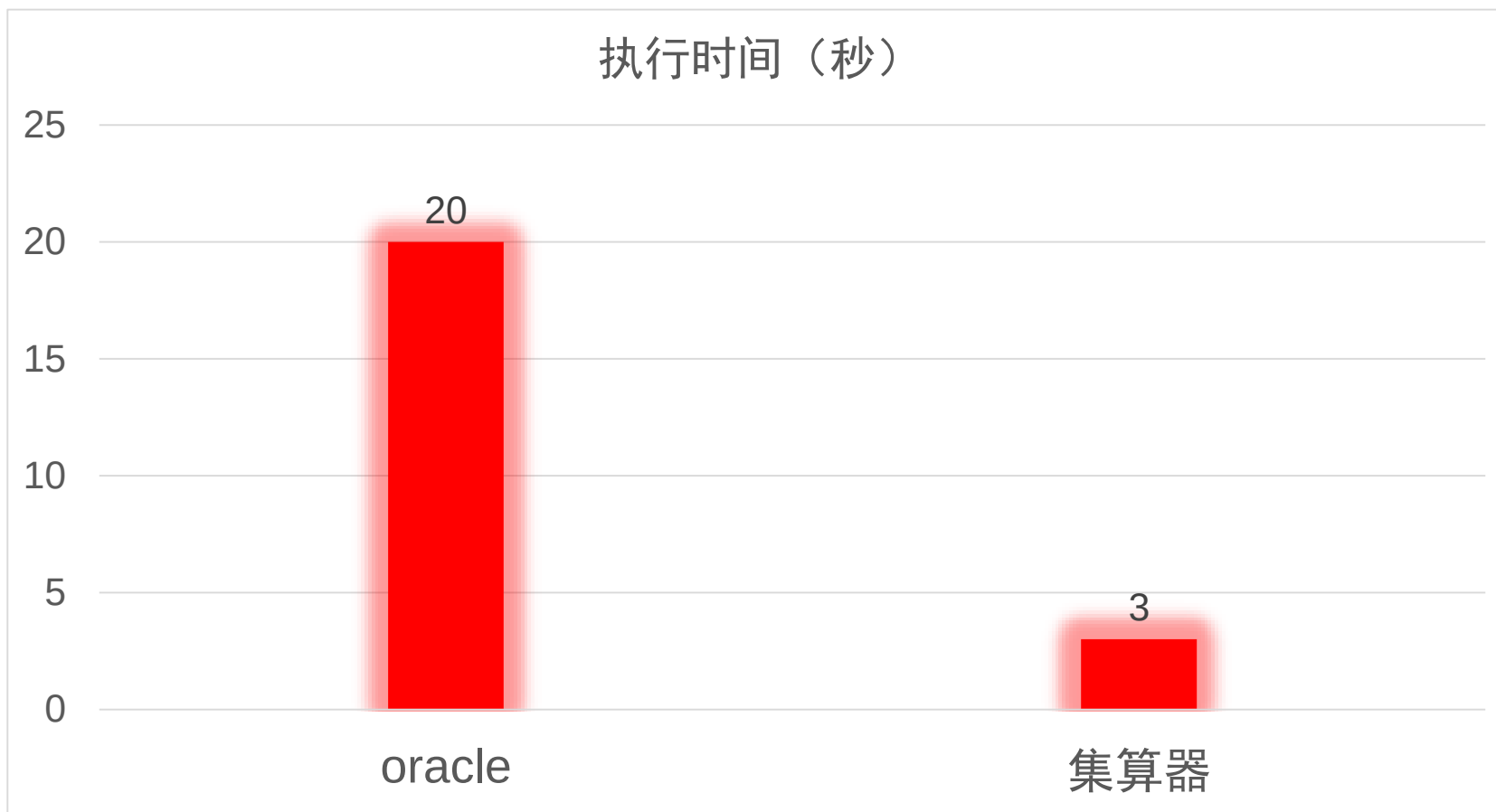
友乾营



12并发的情况下，集算器在低配服务器上也可以达到秒级响应。

★ 某日人民币贷款农业客户总金额-宽表测试

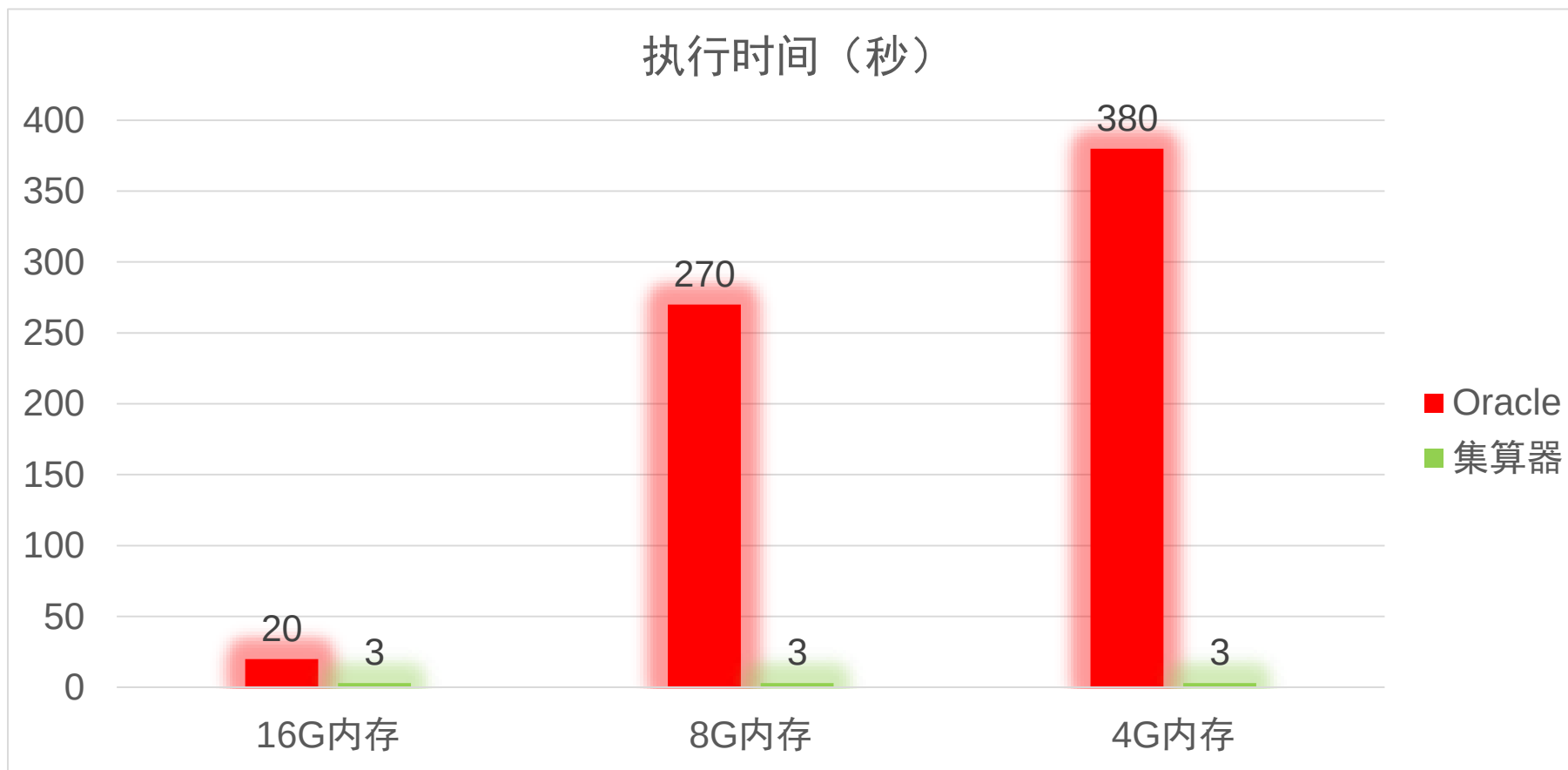
友乾营



集算器横向扩展更简便、成本更低。几百、上千用户并发访问的时候，利用集群计算能力，仍可以满足秒级响应的要求。

★ 某日人民币贷款农业客户总金额-内存大小测试

友乾营



集算器采用外存计算方式，12并发的情况下，在4到16G内存的服务器上都能达到秒级响应。而Oracle在服务器内存变小后，性能下降比较明显。

12并发实时计算指标
3秒

受内存
影响小

低配服务器
性能优异

宽表窄表均
可秒级响应

传统数据无法达到秒级响应的情况。集算器采用新的存储和计算方案，在低配服务器上可达到秒级响应速度，并且受服务器内存变小的影响小。

补充说明

还要考虑什么？



友乾营

专注数据技术的社群

★ 每日新增数据的办法

友乾营

往年组表文件



当年组表文件



每日新增数据



组表文件首先是按日期有序，因此可以在末尾追加每日新增数据。组表文件每年保存一个，到年初的时候新建。

	A
1	<code>=connect("db").cursor("select dt,curr,cust_no,dept... from deposit order by dt,curr,cust_no where dt=? ",etl_dt)</code>
2	<code>=file("deposit.ctx") .create()</code>
3	<code>=A2.append(A1)</code>

A1: 建立数据库连接和游标，取当天新增数据，准备有序读取源数据。

A2: 建立存款文件对象。

A3: 存款组表文件末尾追加A1数据库游标中的数据。

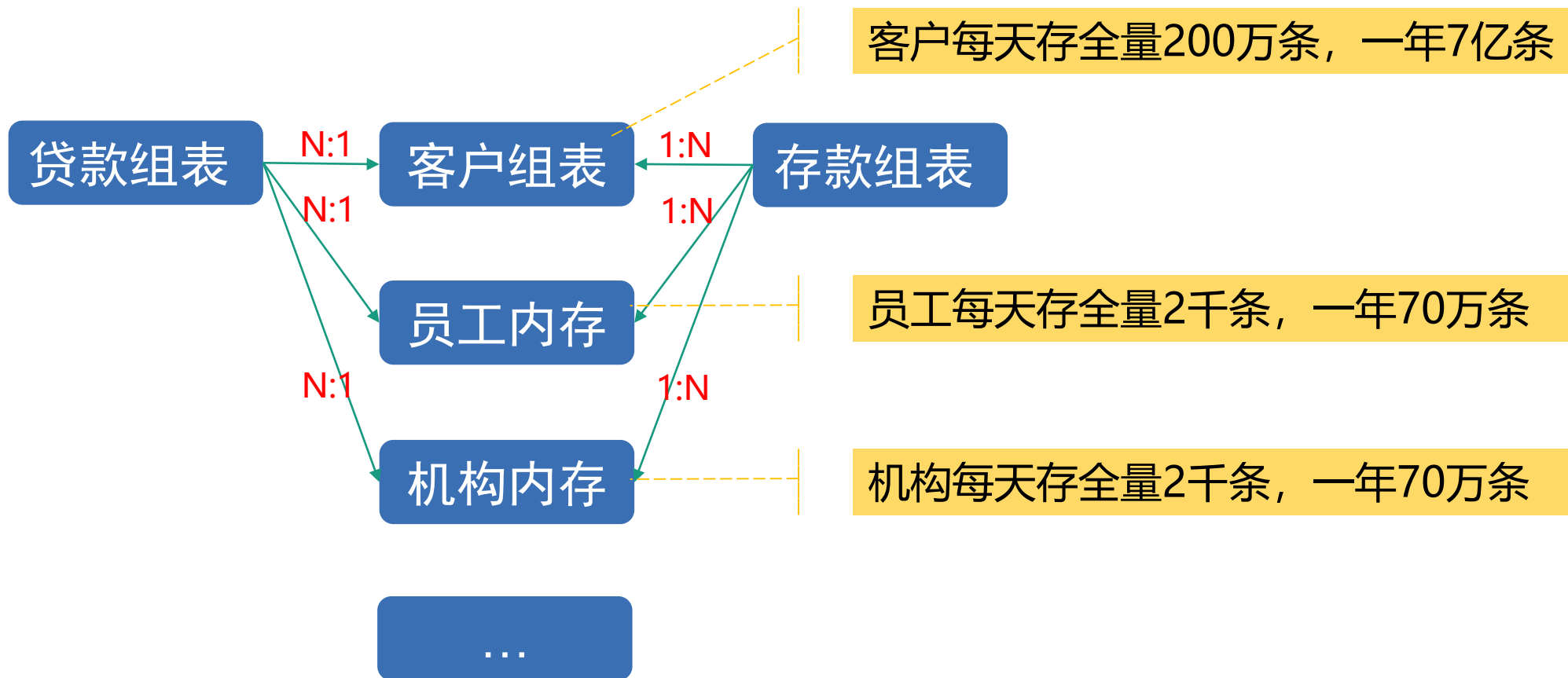
注：客户、贷款主表都与存款的代码近似。

每天新增数据的代码也很简单。每年年初的时候只需简单判断是否要新建组表文件即可，组表文件名可以带上年份。



其他统计方式：员工、机构

友乾营



员工、机构等统计方式和贷款、存款也是主子表关系。但全年只有几十万条记录，可以全内存存储，每天更新增加当日新数据。

	A	B
1	if !ifv(employee)	=file("employee.ctx").import@b(dt,employeeid,employeetype)
2		=env(employee,B1)

A1: 判断全局变量中是否存在employee, 如果存在, 表示已经加载。

B1: 如果全局变量中没有, 那么打开集文件读入。

B2: employee存入全局变量。

员工只需在每天新增数据时加载一次, 结果放到全局变量中缓存。每次查询的时候无需重复加载。

★ 某日一类员工人民币存款总金额

友乾营

	A	B
1	>in_dt=20191001	>in_curr=1
2	=employee.select(dt ==in_dt && employeetype==1)	
3	=file("data/ctx/deposit.ctx").create()	
4	=A3.cursor(employeeid,amt;dt==in_dt&&id!=0&&curr==in_curr)	
5	=A4.switch@i(employeeid,A2:employeeid)	
6	return A5.groups(;sum(amt):amtall)	

A5：主子表的有序关联joinx，改为用游标switch员工内存序表。将员工和存款看做外键关系。
@i选项表示如果员工表中找不到记录的话，就去掉这个存款记录。
注意这个switch在这里也没有立即执行。
A6：对游标执行之前定义好的过滤、外键关联和汇总计算，一次遍历即可完成。
其他和之前的代码类似。

每次实时计算指标的时候可以直接用全局变量employee。这里用cursor.switch@i(...)的方式，比两个组表文件关联速度更快。

好多乾

润乾线上直销系统



好多乾-润乾互联网营销

<http://www.raqsoft.com.cn/wx/hdq-strategy.html>