实时计算新架构处理流程

概述

新实时计算架构的目标是可以对实时任务进行SQL化配置、封装SQL操作通用计算逻辑、使用多层业务主题表管理计算结果、支持计算逻辑口径管理、支持血缘管理、支持移动分析。

本篇文章中，会先给出一段从业务角度出发的需求，分析需求，写出离线SQL任务，最后展示使用新实时计算架构的处理过程。这篇文章，既说明的新实时计算架构思想，又包含了对于元数据及口径的一些管理理念。

需求与口径描述

需求描述

我想统计每个销售每个月的KPI指标数值，包括放款数、放款金额、放款签约金额、进件数、批核数、批核金额、批核签约金额、拒贷数、退件数、待签约数、待签约金额、待放款数、待放款金额。

这些指标都是基于我们当前主要产品统计出来的，每个进件的状态就按照我们通常划分的规则就行。

其中批核相关指标需要加个条件，当前审核状态是非门店申请复议，或者是终定完成。

注意每类指标的时间判断标准不一样，放款类使用约定放款日，进件类使用进件提交时间，批核类和拒贷类使用终定时间。

待签约类和待放款类的指标不需要判断时间，当前只要在这两个状态的进件，都可以算是这个月的指标。

重点信息

我想统计每个销售 每个月的KPI指标数值，包括放款数、放款金额、放款签约金额、进件数、批核数、批核金额、批核签约金额、拒贷数、退件数、待签约数、待签约金额、待放款数、待放款金额。

这些指标都是基于我们当前主要产品统计出来的，每个进件的状态就按照我们通常划分的规则就行。

其中批核相关指标需要加个条件，当前审核状态是非门店申请复议，或者是终定完成。

注意每类指标的时间判断标准不一样，放款类使用约定放款日，进件类使用进件提交时间，批核类和拒贷类使用终定时间。

待签约类和待放款类的指标不需要判断时间，当前只要在这两个状态的进件，都可以算是这个月的指标。

口径列表

放款：lend_requests.state in ('REPAYING','REPAID','EXTENDED','IN_ADVANCE_REPAID')

进件：lend_requests.state in ('SUBMITTED','BRANCH_REJECTED','PROCESSING','CREDIT_TASK_FINISHED','EXTRA_INFORMATION','EXTRA_INFORMATION_COMPLETED','REVIEWING','INIT_REVIEWED','CREDIT_REJECTED','REJECTED','SIGNOFF_AWAIT','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')

批核：lend_requests.state in ('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')

拒贷：lend_requests.state in ('CREDIT_REJECTED')

待签约：lend_requests.state in ('SIGNOFF_AWAIT','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED')

待放款：lend_requests.state = 'LOAN_AWAIT'

退件：audit_results.auditConclusion = 'REJECT' and ar.type in ('FINAL','REVIEW')

金额：lend_requests.amount

签约金额：lend_requests.signedAmount

主要产品：loan_products.type in

('BUSINESS','INSUR_AUTO','INSUR_LIFE','PAYROLL','PERSON','PHONESALE','PHONESALE_MORT','PROP','PROP_MORT','WHITECOLLAR','POS','E_BUSINESS','EASY')

非门店申请复议：audit_results .type <> 'BRANCH' or audit_results .auditConclusion <> 'BRANCH_REVIEWING'

终定完成：audit_results.state = 'SIGNOFF_AWAIT'

约定放款日：lend_requests.loanTime

进件提交时间：lend_requests.submitTime

终定时间：lend_requests.finalAuditTime

离线任务

```
select
lr.seller_id
,sum(case when lr.state in ('REPAYING','REPAID','EXTENDED','IN_ADVANCE_REPAID') and lp.id is not null and date(lr.loanTime) >= '2017-03-01' and  date(lr.loanTime) <= '2017-03-31' then 1 else 0 end) as 放款数
,sum(case when lr.state in ('REPAYING','REPAID','EXTENDED','IN_ADVANCE_REPAID') and lp.id is not null and date(lr.loanTime) >= '2017-03-01' and  date(lr.loanTime) <= '2017-03-31'   then lr.amount else 0 end) as 放款金额
,sum(case when lr.state in ('REPAYING','REPAID','EXTENDED','IN_ADVANCE_REPAID') and lp.id is not null and date(lr.loanTime) >= '2017-03-01' and  date(lr.loanTime) <= '2017-03-31'   then lr.signedAmount  else 0 end) as 放款签约金额
,sum(case when lr.state in
('SUBMITTED','BRANCH_REJECTED','PROCESSING','CREDIT_TASK_FINISHED','EXTRA_INFORMATION','EXTRA_INFORMATION_COMPLETED','REVIEWING','INIT_REVIEWED','CREDIT_REJECTED',

'REJECTED','SIGNOFF_AWAIT','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')
          and lp.id is not null and date(lr.submitTime) >= '2017-03-01' and date(lr.submitTime) <= '2017-03-31' then 1 else 0 end) as 进件数
,sum(case when lr.state in
('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')
          and lp.id is not null and date(lr.finalAuditTime) >= '2017-03-01' and  date(lr.finalAuditTime) <= '2017-03-31' and ar4.id = ar6.id then 1
      when lr.state in
('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')
          and lp.id is not null and date(lr.finalAuditTime) >= '2017-03-01' and  date(lr.finalAuditTime) <= '2017-03-31' and ar4.id <> ar6.id and ar6.state = 'SIGNOFF_AWAIT' then 1
          else 0 end) as 批核数
,sum(case when lr.state in
('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')
          and lp.id is not null and date(lr.finalAuditTime) >= '2017-03-01' and  date(lr.finalAuditTime) <= '2017-03-31' and ar4.id = ar6.id then lr.amount
      when lr.state in
('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')
          and lp.id is not null and date(lr.finalAuditTime) >= '2017-03-01' and  date(lr.finalAuditTime) <= '2017-03-31' and ar4.id <> ar6.id and ar6.state = 'SIGNOFF_AWAIT' then lr.amount
          else 0 end) as 批核金额
,sum(case when lr.state in
('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')
          and lp.id is not null and date(lr.finalAuditTime) >= '2017-03-01' and  date(lr.finalAuditTime) <= '2017-03-31' and ar4.id = ar6.id then lr.signedAmount
      when lr.state in
('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED')
          and lp.id is not null and date(lr.finalAuditTime) >= '2017-03-01' and  date(lr.finalAuditTime) <= '2017-03-31' and ar4.id <> ar6.id and ar6.state = 'SIGNOFF_AWAIT' then lr.signedAmount
          else 0 end) as 批核签约金额
,sum(case when lr.state in ('CREDIT_REJECTED') and lp.id is not null and date(lr.finalAuditTime) >= '2017-03-01' and date(lr.finalAuditTime) <= '2017-03-31' then 1 else 0 end) as 拒贷数
,sum(case when ar.lendRequest_id is not null and lp.id is not null then ar.reject_cnt else 0 end) as 退件数
-- 待签约数
,sum(case when lr.state in ('SIGNOFF_AWAIT','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED') then 1 else 0 end) as 待签约数
```

```
-- 待签约金额
,sum(case when lr.state in ('SIGNOFF_AWAIT','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED') then lr.amount else 0 end) as 待签约金额
-- 待放款
,sum(case when lr.state = 'LOAN_AWAIT' then 1 else 0 end) as 待放款数
-- 待放款
,sum(case when lr.state = 'LOAN_AWAIT' then lr.amount else 0 end) as 待放款金额
from thread.lend_requests lr
-- 过滤产品
left join thread.loan_products lp on lr.appliedProduct_id = lp.id and lp.type in ('BUSINESS','INSUR_AUTO','INSUR_LIFE','PAYROLL','PERSON','PHONESALE','PHONESALE_MORT','PROP','PROP_MORT','WHITECOLLAR','POS','E_BUSINESS','EASY')
-- 计算退件数
left join (select ar.lendRequest_id,count(1) as reject_cnt,ar.createTime from thread.audit_results ar where ar.auditConclusion = 'REJECT' and ar.type in ('FINAL','REVIEW') and date(ar.createTime) >= '2017-03-01' and date(ar.createTime) <= '2017-03-31' group by lendRequest_id)ar on ar.lendRequest_id = lr.id
-- 计算批核数
left join (select max(id) as id,lendRequest_id from thread.audit_results ar2 group by ar2.lendRequest_id) ar2 on ar2.lendRequest_id = lr.id
left join thread.audit_results ar4 on ar4.id = ar2.id
left join (select max(id) as id,lendRequest_id from thread.audit_results ar2 where ar2.type <> 'BRANCH' or ar2.auditConclusion <> 'BRANCH_REVIEWING' group by ar2.lendRequest_id) ar5 on ar5.lendRequest_id = lr.id
left join thread.audit_results ar6 on ar5.id = ar6.id
-- 帅选条件，按日期刷选
where (date(lr.loanTime) >= '2017-03-01' or date(lr.finalAuditTime) >= date_sub('2017-03-01',interval 14 day) or date(lr.submitTime) >= '2017-03-01' or date(lr.createTime) >= '2017-03-01' or date(ar.createTime) >= '2017-03-01')
group by lr.seller_id
```

实时处理流程

处理思路

元数据管理、数据治理的核心内容之一在于口径管理，口径管理的结果不只是通过某种方法把SQL中的口径标识并且展示出来供查询，还应该更进一步对现有SQL任务进行优化改造，把口径以字段的方式固化下来。

在新实时处理平台中，通用口径或者关键口径计算生成的数据，以字段的方式保存在不同业务的主题表中，供其它任务使用。也就是说，一个主题表中的字段即是一个口径。

主题表

在写计算逻辑之前，需要事先定义好主题表，类似Mysql当中的CreateTable操作，Athena平台中的主题表定义不同于传统数据库表的二维表定义，而是可以定义成JSON格式的可嵌套的主题表，字段的类型也可以为List类型。

上述SQL中涉及到三个原始数据表：

- lend_requests：进件主表
- loan_products：进件产品表
- audit_results：审核结果表

针对这个任务，可以只把原始表中用到的字段，导入到数据源主题表中，本任务涉及到的主题表如下。

lend_requests主题表：把lend_requests原表需要的字段导入进主题表

lend_request_id	平行主键
seller_id	子键
state	字段
appliedProduct_id	子键
amount	字段
signedAmount	字段
createTime	字段
submitTime	字段
finalAuditTime	字段
signedTime	字段
loanTime	字段

loan_products主题表：把loan_products原表需要的字段导入进主题表

id	平行主键
type	字段

audit_results主题表：把audit_results原表需要的字段导入进主题表

id	平行主键
type	字段
auditConclusion	字段
createTime	字段

lend主题表：进件宽表，所有跟进件相关的信息都可以保存在这个主题宽表中，在此只展示跟本任务有关的字段

lend_request_id	平行主键
seller_id	子键
state	字段
appliedProduct_id	子键
amount	字段
signedAmount	字段
createTime	字段
submitTime	字段
finalAuditTime	字段
signedTime	字段
loanTime	字段
is_valid_product	字段
is_fangkuan	字段
is_jinjian	字段
is_pihe	字段
is_jvdai	字段
is_daiqianyue	字段
is_daifangkuan	字段
audit_results_max_id	字段
audit_results_match_max_id	字段
min_ar_creat_time	字段
audit_results	JSON List
audit_results.id	平行主键
audit_results.type	字段
audit_results.auditConclusion	字段
audit_results.createTime	字段

seller_date_info主题表：销售按照日期划分的结果表

seller_id	平行主键
date	日期
fangkuan_count	字段
fangkuan_amount	字段
fangkuan_signed_amount	字段
jinjian_count	字段
pihe_count	字段
pihe_amount	字段
pihe_signed_amount	字段
jvdai_count	字段
tuijian_count	字段

seller_info主题表：销售不按照日期划分的结果表

seller_id	平行主键
daiqianyue_count	字段
daiqianyue_amount	字段
daifangkuan_count	字段
daifangkuan_amount	字段

处理逻辑（口径）

口径即是处理逻辑，每个口径包含一到多行类SQL语句，多个口径可以在一个口径组中进行管理，口径组中口径之间的执行顺序，可以以DAG的方式进行安排配置。

口径组1： 源主题表：lend_requests 目标主题表：lend 把lend_requests需要的信息整合到lend表中

```
save lend_requests.id as lend.lend_request_id
save lend_requests.seller_id as lend.seller_id
save lend_requests.state as lend.state
save lend_requests.appliedProduct_id as lend.appliedProduct_id
save lend_requests.amount as lend.amount
save lend_requests.signedAmount as lend.signedAmount
save lend_requests.createTime as lend.createTime
save lend_requests.submitTime as lend.submitTime
save lend_requests.finalAuditTime as lend.finalAuditTime
save lend_requests.signedTime as lend.signedTime
save lend_requests.loanTime as lend.loanTime
```

口径组2： 源主题表：loan_products 目标主题表：lend 把loan_products需要的信息整合到lend表中

```
save loan_products.id as lend.appliedProduct_id
save loan_products.type as lend.product_type
```

口径组3： 源主题表：audit_results 目标主题表：lend 把audit_results需要的信息整合到lend表中

```
save audit_results.id as lend.audit_results.id
save audit_results.type as lend.audit_results.type
save audit_results.auditConclusion as lend.audit_results.auditConclusion
save audit_results.createTime as lend.audit_results.createTime
```

口径组4： 源主题表：lend 目标主题表：lend 计算审核结果最大ID、最早创建时间、符合条件的最大ID

口径分支1：计算审核结果最大ID、最早创建时间

```
group by lend.audit_results
max(audit_results.id) as max_id
min(audit_results.createTime) as min_ar_creat_time
save max_id as lend.audit_results_max_id
save min_ar_creat_time as lend.min_ar_creat_time
```

口径分支2：计算符合条件的审核结果最大ID

```
where audit_results.type <> 'BRANCH' or audit_results.auditConclusion <> 'BRANCH_REVIEWING'
group by lend.audit_results
save max_id as lend.audit_results_match_max_id
```

口径组5： 源主题表：lend 目标主题表：lend 计算待统计的合规产品类型

```
case when where lend.product_type in
```

('BUSINESS','INSUR_AUTO','INSUR_LIFE','PAYROLL','PERSON','PHONESALE','PHONESALE_MORT','PROP','PROP_MORT','WHITECOLLAR','POS','E_BUSINESS','EASY') then 1 else 0 end as is_valid_product
save is_valid_product as lend.is_valid_product

口径组6： 源主题表：lend 目标主题表：lend 计算各种进件状态

case when where lend.state in ('REPAYING','REPAID','EXTENDED','IN_ADVANCE_REPAID') then 1 else 0 end as is_fangkuan

case when where lend.state in ('SUBMITTED','BRANCH_REJECTED','PROCESSING','CREDIT_TASK_FINISHED','EXTRA_INFORMATION','EXTRA_INFORMATION_COMPLETED','REVIEWING','INIT_REVIEWED','CREDIT_REJECTED', 'REJECTED','SIGNOFF_AWAIT','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED','LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED') then 1 else 0 end as is_jinjian

case when where lend.state in ('BRANCH_REJECTED','SIGNOFF_AWAIT','REVIEWING','SIGNOFF_CANCELLED','GIVE_UP','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED', 'LOAN_AWAIT','REPAYING','IN_ADVANCE_REPAID','REPAID','EXTENDED') then 1 else 0 end as is_pihe

case when where lend.state in ('CREDIT_REJECTED') then 1 else 0 end as is_jvdai

case when where lend.state in ('SIGNOFF_AWAIT','LOAN_APPROVE_AWAIT','LOAN_PAY_REJECT','LOAN_REJECTED') then 1 else 0 end as is_daiqianyue

case when where lend.state in ('LOAN_AWAIT') then 1 else 0 end as is_daifangkuan

save is_fangkuan,is_jinjian,is_pihe,is_jvdai,is_daiqianyue,is_daifangkuan as lend.is_fangkuan,lend.is_jinjian,lend.is_pihe,lend.is_jvdai,lend.is_daiqianyue,lend.is_daifangkuan

口径组7： 源主题表：lend 目标主题表：seller_date_info 生成每个销售每天的相关KPI信息

顶层通用口径：where lend.is_valid_product and((date(lend.loanTime) >= month_start() or date(lend.finalAuditTime) >= date_sub(month_start(),interval 14 day) or date(lend.submitTime) >= month_start() or date(lend.createTime) >= month_start() or lend.min_ar_creat_time>=month_start()))

口径分支1：计算放款相关统计结果，以loanTime为时间基准

where lend.is_fangkuan

group by lend.seller_id,date(lend.loanTime)

count(*) as fangkuan_count

sum(lend.amount) as fangkuan_amount

sum(lend.signedAmount) as fangkuan_signed_amount

save fangkuan_count as seller_date_info.fangkuan_count

save fangkuan_amount as seller_date_info.fangkuan_amount

save fangkuan_signed_amount as seller_date_info.fangkuan_signed_amount

口径分支2：计算进件相关统计结果，以submitTime为时间基准

where lend.is_jinjian

group by lend.seller_id,date(lend.submitTime)

count(*) as jinjian_count

save jinjian_count as seller_date_info.jinjian_count

口径分支3：计算批核相关统计结果，以finalAuditTime为时间基准

where lend.is_pihe

where lend.audit_results_max_id == lend.audit_results_match_max_id or (lend.audit_results_max_id <>

lend.audit_results_match_max_id and lend.state = 'SIGNOFF_AWAIT')

group by lend.seller_id,date(lend.finalAuditTime)

count(*) as pihe_count

sum(lend.amount) as pihe_amount

sum(lend.signedAmount) as pihe_signed_amount

save pihe_count as seller_date_info.pihe_count

save pihe_amount as seller_date_info.pihe_amount

save pihe_signed_amount as seller_date_info.pihe_signed_amount

口径分支4：计算拒贷相关统计结果，以finalAuditTime为时间基准

where lend.is_jvdai

group by lend.seller_id,date(lend.finalAuditTime)

count(*) as jvdai_count

save jvdai_count as seller_date_info.jvdai_count

口径分支5：计算退件相关统计结果，以audit_results中的createTime为时间基准

where lend.audit_results.auditConclusion = 'REJECT' and lend.audit_results.type in ('FINAL','REVIEW')

group by lend.seller_id,date(lend.audit_results.createTime)

count(*) as tuijian_count

save tuijian_count as seller_date_info.tuijian_count

口径组8： 源主题表：lend 目标主题表：seller_info 生成每个销售的一些KPI信息

口径分支1：计算待签约相关统计结果

where lend.is_daiqianyue

group by lend.seller_id

count(*) as daiqianyue_count

sum(lend.amount) as daiqianyue_amount

save daiqianyue_count as seller_info.daiqianyue_count

save daiqianyue_amount as seller_info.daiqianyue_amount

口径分支2：计算待放款相关统计结果

where lend.is_daifangkuan

group by lend.seller_id

count(*) as daifangkuan_count

sum(lend.amount) as daifangkuan_amount

save daifangkuan_count as seller_info.daifangkuan_count

save daifangkuan_amount as seller_info.daifangkuan_amount