润乾
RAQSOFT

怎样减少
报表开发中对存储过程的依赖

润乾报表库外存储过程

www.raqsoft.com.cn

存储过程在报表开发中的利与弊

存储过程之利

支持多步运算

在存储过程中可以通过多步运算，实现复杂数据处理逻辑

利于界面与逻辑分离

计算逻辑放在数据库中完成利于计算存储层和前端应用层的分离

计算性能高

存储过程采用预编译机制运算性能更高

存储过程之弊

对数据库影响过大

存储过程过多会带来数据库管理问题，同时库内运行对数据库造成过大压力，创建和修改存储过程需要很高权限存在安全问题

开发调试困难（不解释）

难以扩展和移植

分库后无法跨库查询；更换数据库无法移植

共享调用引起高耦合

存储过程可能被共享后，难以确定被哪些应用调用

润乾报表方案

润乾报表通过计算模块可以将原来在库内的存储过程移到库外，解决库内存储过程带来的难题

库外存储过程如何使用?

【例子】根据游戏账户表和状态表查询前后十名用户的本期在线时长、上期在线时长和上上期在线时长，及汇总结果

用户统计日报表

统计周期：天　　2014-01-08

	序号	用户	首次登陆时间	本期在线时长	上期在线时长	上上期在线时长
前十名	1	61728492	2010-10-27 13:50:59	162521	113592	113592
	2	87640156	2010-05-27 21:20:54	120469	67040	67040
	3	83304398	2010-06-11 06:29:52	119761	78821	78821
	4	34840725	2010-06-23 17:31:51	115515	884	884
	5	100295376	2013-12-16 20:28:47	110159	38001	38001
	6	74186743	2010-05-28 22:54:21	109722	46558	46558
	7	88717049	2010-05-09 21:06:01	108895	55191	55191
	8	45385931	2010-07-18 20:31:51	108810	12711	12711
	9	87154466	2013-09-22 19:02:07	107127	55999	55999
	10	51058932	2010-09-04 15:51:24	106112	8086	8086
后十名	242	8202454	2013-12-15 01:15:04	96	225	225
	243	89318634	2013-11-22 10:43:59	96	67365	67365
	244	36427538	2010-06-09 00:16:11	92	62140	62140
	245	67452183	2010-08-23 06:05:44	78	292	292
	246	94465850	2010-05-20 06:09:05	72	404	404
	247	57503665	2011-12-10 21:09:39	32	34	34
	248	89177182	2012-01-14 15:49:36	8	78066	78066
	249	89318744	2013-11-23 03:27:38	2	51133	51133
	250	89319419	2013-11-30 02:33:29	1	29586	29586
	251	8203324	2013-12-22 22:47:01	1	10667	10667
持续活跃数	324	持续活跃用户本期流失数	73			
上期新增用户数	0	上期新增用户本期流失数	0			
上期有效用户回流数	150	上期回归有效用户本期流失数	96			

表样

数据来自于两个数据表：

```
create table T_DW_ZX_ACCOUNT_STATUS_DAY
(
  LOGTIME    DATE,--日志时间
  USERID     NUMBER(12),--用户号
  ACCOUNT    VARCHAR2(50),--账号
  ONLINETIME NUMBER(8),--在线时间
  PAY        NUMBER(11),
  EXPEND     NUMBER(11),
  TOP_LEVEL  NUMBER(4)
);
create table T_DW_ZX_VALID_ACCOUNT
(
  USERID            NUMBER,--用户编号
  FIRST_LOGOUT_TIME DATE,--第一次登出时间
  STANDARD_7D_TIME  DATE,
  STANDARD_14D_TIME DATE,
  ACCOUNT           VARCHAR2(50)
);
```

原数据库存储过程准备数据

```oracle plsql
CREATE OR REPLACE PACKAGE BODY CURSPKG AS
  PROCEDURE sp_query_user_status_day(data_date IN varchar2,
                                     top10     OUT T_CURSOR,
                                     last10    in out t_cursor,
                                     var1      out number,
                                     var2      out number,
                                     var3      out number,
                                     var4      out number,
                                     var5      out number,
                                     var6      out number) IS
    V_CURSOR1 T_CURSOR; --top10
    V_CURSOR2 T_CURSOR; --last10
    V_CURSOR  T_CURSOR; --temp table
    v_ttime   date;
    temp_num  number;

    v_valid_user_conti_act1      number;
    v_valid_user_back1           number;
    v_valid_user_conti_act_lost1 number;
    v_valid_user_active_lost1    number;
    v_valid_user_add_lost1       number;
    v_valid_user_back_lost1      number;

  BEGIN
    v_ttime := to_date(data_date, 'yyyy-mm-dd');

    --for temp table
    select count(1) into temp_num from account_status_day_temp;
    if temp_num > 0 then
      delete from account_status_day_temp;    --delete first
    end if;
    insert into account_status_day_temp
      select *
        from (select v.userid, v.first_logout_time
                from t_dw_zx_valid_account v
               where v.standard_7d_time is not null) a,
             (select userid, sum(onlinetime) onlinetime, max(account)
                from t_dw_zx_account_status_day
               where logtime >= v_ttime
                 and logtime < v_ttime + 1
               group by userid
              having max(account) is not null) b,
             (select userid, sum(onlinetime) onlinetime, max(account)
                from t_dw_zx_account_status_day
               where logtime >= v_ttime - 1
                 and logtime < v_ttime
               group by userid
              having max(account) is not null) c,
             (select userid, sum(onlinetime) onlinetime, max(account)
                from t_dw_zx_account_status_day
               where logtime >= v_ttime - 1 - 1
                 and logtime < v_ttime - 1
               group by userid
              having max(account) is not null) d
        where a.userid = b.userid(+)
          and a.userid = c.userid(+);

     commit;

     --top 10
    open V_CURSOR1 for
      select *
        from (select rownum,
                     a.auserid userid,
                     a.first_logout_time,
                     a.bonlinetime current_onlinetime,
                     a.conlinetime last_onlinetime,
                     a.donlinetime last_last_onlinetime
                from account_status_day_temp a
               order by bonlinetime desc)
       where rownum < 11;

      --last 10
      open V_CURSOR2 for
      select *
        from (select rownum,
                     a.auserid userid,
                     a.first_logout_time,
                     a.bonlinetime current_onlinetime,
                     a.conlinetime last_onlinetime,
                     a.donlinetime last_last_onlinetime
                from account_status_day_temp a
               order by bonlinetime asc)
       where rownum < 11;

       top10 := V_CURSOR1;
       last10 := V_CURSOR2;

     --total
    select
             valid_user_conti_act
           , valid_user_back
           , valid_user_conti_act_lost
           , valid user active lost
           , valid_user_add_lost
           , valid_user_back_lost
     into
             v_valid_user_conti_act1
           , v_valid_user_back1
           , v_valid_user_conti_act_lost1
           , v_valid_user_active_lost1
           , v_valid_user_add_lost1
           , v_valid_user_back_lost1
    from
    (select  count(case when buserid is not null
             and cuserid is not null then 1 else null end) valid_user_conti_act
             , count(case when cuserid is null
             and buserid is not null and first_logout_time  <  v_ttime-1 then 1 else null end)
             , count(case when cuserid is not null
             and buserid is null then 1 else null end)  valid_user_active_lost
             , count(case when duserid is not null and cuserid is not null
             and buserid is null then 1 else null end) valid_user_conti_act_lost
             , count(case when duserid is null and cuserid is not null
             and first_logout_time < v_ttime-1 and buserid is null then 1 else null end) valid_user_back_lost
             , count(case when buserid is null and first_logout_time  >= v_ttime-1
             and  first_logout_time < v_ttime then 1 else null end)  valid_user_add_lost
     from account_status_day_temp);
          var1 := v_valid_user_conti_act1;
          var2 := v_valid_user_back1;
          var3 := v_valid_user_conti_act_lost1;
          var4 := v_valid_user_active_lost1;
          var5 := v_valid_user_add_lost1;
          var6 := v_valid_user_back_lost1;

  END sp_query_user_status_day;
END CURSPKG;
```

润乾报表实现

1. 新建报表，使用脚本数据集准备数据

数据集设置

名称	类型	数据源
ds1,ds2,ds3	脚本数据集	

确定(O)　取消(C)

脚本数据集

缓存文件名

确定(O)　取消(C)　复制(C)　粘贴(P)

	A
2	=A1.query("select userid,first_logout_time,standard_7d_time from t_dw_zx_valid_account where standard_7d_time is not null")
3	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" '"+string(data_date)+"' "+" group by userid having max(account) is not null")
4	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" '"+string(data_date-1)+"' "+" group by userid having max(account) is not null")
5	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" '"+string(data_date-2)+"' "+" group by userid having max(account) is not null")
6	=A1.query("select userid,onlinetime,account,logtime from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+"'"+string(data_date-2)+"'")
7	=join@1(A2:a,USERID;A3:b,USERID;A4:c,USERID;A5:d,USERID)
8	=A7.new(#:no,a.USERID:userid,a.FIRST_LOGOUT_TIME:first_logout_time,b.ONLINETIME:current_time,c.ONLINETIME:last_time,c.ONLINETIME:last_last_time)
9	=A8.to(10)
10	=A8.to(A7.len()-10+1,)
11	=A6.count@b(d!=null && c!=null)
12	=A6.count@b(b==null && c!=null && d!=null)
13	=A6.count@b(a.FIRST_LOGOUT_TIME>data_date-1 && a.FIRST_LOGOUT_TIME<data_date)
14	=A6.count@b(b==null && a.FIRST_LOGOUT_TIME>data_date-1 && a.FIRST_LOGOUT_TIME<data_date)
15	=A6.count@b(d==null && c!=null && a.FIRST_LOGOUT_TIME<data_date-1)
16	=A6.count@b(b==null && c!=null && d==null && a.FIRST_LOGOUT_TIME<data_date-1)
17	=new(A11:num1,A12:num2,A13:num3,A14:num4,A15:num5,A16:num6)
18	>A1.close()
19	result A9,A10,A17

这里使用了脚本数据集

同时注意数据集名称为ds1,ds2,ds3，用于接收脚本返回的三个结果集

在润乾报表中，支持返回并使用多个结果集

脚本解释请见下一页

SPL脚本解释

	A	B
1	=connect("ora")	/连接数据库
2	=A1.query("select userid,first_logout_time,standard_7d_time from t_dw_zx_valid_account where standard_7d_time is not null")	/这5行根据参数分别执行SQL到数据库中取数，为节约数据库JDBC的取数成本，需要在数据库中先进行一部分运算（如：过滤、分组汇总等） 【注意】润乾报表的“库外存储过程”并不是把原始数据全部从数据库取数后再运算（虽然也可以），而是让数据库执行一些基本运算，这样可以减少数据传输的IO成本，提升性能。 此外，必要时候仍然可以由DB去执行复杂SQL，脚本用于控制逻辑过程
3	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" '"+string(data_date)+"' "+" group by userid having max(account) is not null")	
4	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" '"+string(data_date-1)+"' "+" group by userid having max(account) is not null")	
5	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" '"+string(data_date-2)+"' "+" group by userid having max(account) is not null")	
6	=A1.query("select userid,onlinetime,account,logtime from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+"'"+string(data_date-2)+"'")	
7	=join@1(A2:a,USERID;A3:b,USERID;A4:c,USERID;A5:d,USERID)	/关联A2-A5四部分结果
8	=A7.new(#:no,a.USERID:userid,a.FIRST_LOGOUT_TIME:first_logout_time,b.ONLINETIME:current_time,c.ONLINETIME:last_time,c.ONLINETIME:last_last_time)	/根据A7建立新序表，用于读取前后十名记录
9	=A8.to(10)	/通过序号取前10名
10	=A8.to(A7.len()-10+1,)	/后十名
11	=A6.count@b(d!=null && c!=null)	/A11-A17计算汇总值
12	=A6.count@b(b==null && c!=null && d!=null)	
13	=A6.count@b(a.FIRST_LOGOUT_TIME>data_date-1 && a.FIRST_LOGOUT_TIME<data_date)	
14	=A6.count@b(b==null && a.FIRST_LOGOUT_TIME>data_date-1 && a.FIRST_LOGOUT_TIME<data_date)	
15	=A6.count@b(d==null && c!=null && a.FIRST_LOGOUT_TIME<data_date-1)	
16	=A6.count@b(b==null && c!=null && d==null && a.FIRST_LOGOUT_TIME<data_date-1)	
17	=new(A11:num1,A12:num2,A13:num3,A14:num4,A15:num5,A16:num6)	
18	>A1.close()	
19	return A9,A10,A17	将前十名、后十名以及汇总值以多个结果集返回报表

润乾报表实现

2. 报表模板设计

	A	B	C	D	E	F	G
1(标)	用户统计日报表						
2(标)	统计周期:天						=@data_date
3(头)		序号	用户	首次登陆时间	本期在线时长	上期在线时长	上上期在线时长
4	前十名	=ds1.select(no)	=ds1.userid	=ds1.first_logo	=ds1.current_ti	=ds1.last_time	=ds1.last_last_tim
5	后十名	=ds2.select(no)	=ds2.userid	=ds2.first_logo	=ds2.current_ti	=ds2.last_time	=ds2.last_last_tim
6	持续活跃数	=ds3.num1	持续活跃用户本期流失数	=ds3.num4			
7	上期新增用户数	=ds3.num2	上期新增用户本期流失数	=ds3.num5			
8	上期有效用户回流	=ds3.num3	上期回归有效用户本期流失	=ds3.num6			

在报表中直接使用ds1、ds2、ds3呈现数据

如何被其他报表调用?

原来的存储过程可以被多个报表调用，润乾报表通过SPL脚本实现的“库外存储过程”，如何被其他报表使用呢?

通过“润乾报表IDE-工具-脚本编辑“将其保存成独立的dfx文件，供其他报表调用

脚本编辑

	A
1	=connect("ora")
2	=A1.query("select userid,first_logout_time,standard_7d_time from t_dw_zx_valid_account where standard_7d_time is not null")
3	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" "+string(data_date)+"' "+" group by userid having max(account) is not null")
4	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" "+string(data_date-1)+"' "+" group by userid having max(account) is not null")
5	=A1.query("select userid, sum(onlinetime) onlinetime, max(account) account from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+" "+string(data_date-2)+"' "+" group by userid having max(account) is not null")
6	=A1.query("select userid,onlinetime,account,logtime from t_dw_zx_account_status_day where to_char(logtime,'yyyy-MM-dd')="+"'"+string(data_date-2)+"')
7	=join@1(A2:a,USERID;A3:b,USERID;A4:c,USERID;A5:d,USERID)
8	=A7.new(#:no,a.USERID:userid,a.FIRST_LOGOUT_TIME:first_logout_time,b.ONLINETIME:current_time,c.ONLINETIME:last_time,c.ONLINETIME:last_last_time)
9	=A8.to(10)
10	=A8.to(A7.len()-10+1,)
11	=A6.count@b(d!=null && c!=null)
12	=A6.count@b(b==null && c!=null && d!=null)
13	=A6.count@b(a.FIRST_LOGOUT_TIME>data_date-1 && a.FIRST_LOGOUT_TIME<data_date)
14	=A6.count@b(b==null && a.FIRST_LOGOUT_TIME>data_date-1 && a.FIRST_LOGOUT_TIME<data_date)
15	=A6.count@b(d==null && c!=null && a.FIRST_LOGOUT_TIME<data_date-1)
16	=A6.count@b(b==null && c!=null && d==null && a.FIRST_LOGOUT_TIME<data_date-1)
17	=new(A11:num1,A12:num2,A13:num3,A14:num4,A15:num5,A16:num6)
18	>A1.close()
19	return A9,A10,A17

打开(O) 保存(S) 关闭(X) 复制(C) 粘贴(V) 参数(P) 执行(E)

myProc.dfx

SPL脚本既可以内嵌在报表（应用）中，跟随报表存储，也可以形成独立的脚本文件（DFX）存储在应用中供多个报表调用；

这样，报表应用在使用时就获得了相当的灵活性，同时也解决了原来由于高耦合带来的诸多问题

如何被其他报表调用?

其他报表在脚本数据集中调用myProc.dfx获得需要的结果集

在脚本数据集中使用call函数调用“库外存储过程” myProc.dfx

对于dfx中返回的多个结果集可以按需取用，方法是通过序号取得各个结果集

如本例中使用A1(1)和A1(2)获得了myProc.dfx中返回的前两个结果集

能否延续存储过程的优点，改正缺点？

前面我们罗列了存储过程的优缺点，那“库外存储过程”的方式是否能延续优点，改正缺点呢？

延续优点

支持多步运算

在上述的例子中可以发现，润乾报表提供的脚本方式天然支持多步计算

利于界面与逻辑分离

润乾报表提供的计算模块（脚本数据集）将报表（前端应用）和数据库（后端存储）进行了有效隔离，分层设计更利于解耦

计算性能高

SPL脚本可以优化原计算的执行路径（如中间结果不落地），从而可以获得更高性能。如果再结合数据外置则性能更高，请参考：《有效减少数据库中间表的报表开发方法》

改正缺点

对数据库影响过大（解决）

由于是库外运算，对数据库影响很小，管理、性能和安全都不会再对数据库产生影响

开发调试困难（解决）

提供了带有可视化编辑调试功能的独立工具；同时SPL脚本更适合做复杂运算（详见下两页）

难以扩展和移植（解决）

由于计算都是在脚本中完成，数据库更改或分库，只需要改变或增加数据库连接即可，算法几乎不用更改，详见：《在报表中直接使用多样性数据源》

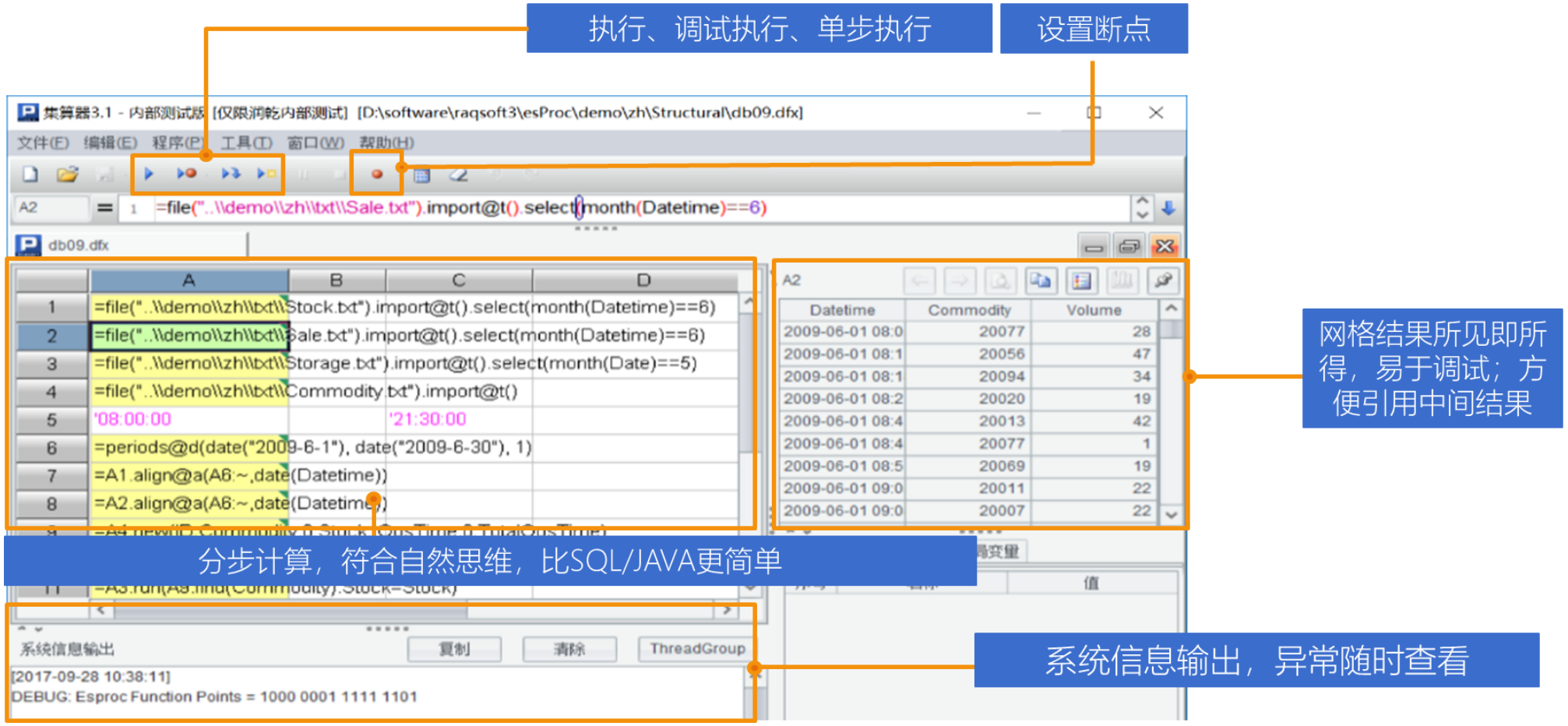
共享带来的耦合与管理问题（解决）

脚本随应用（报表/DFX）存储，随需存放和取用

如何更方便地调试?

润乾提供了单独的脚本编辑工具（集算器），可以从润乾官网（下载地址）下载使用

注意：集算器是单独的商业产品，正式使用需要购买，对于只编辑脚本来说使用开发授权就够了，开发授权同样通过上面的链接处下载



脚本计算的简捷性举例

某支股票最长连续涨了多少交易日

```
select max(连续日数)
from (select count(*) 连续日数
      from (select sum(涨跌标志) over(order by 交易日) 不涨日数
            from (select 交易日,
                         case when 收盘价>lag(收盘价) over(order by 交易日)
                         then 0 else 1 end 涨跌标志
                  from 股价表) )
      group by 不涨日数)
```

SQL

	A
1	=股价表.sort(交易日)
2	=0
3	=A1.max(A2=if(收盘价>收盘价[-1],A2+1,0))

SPL脚本

在比较与上一天的股价中，SPL脚本直接使用了[-1]代表上一天的价格，从而通过3步就轻松写出这样的运算

润乾
RAQSOFT

想要了解更多 请联系我们

技术内容请移步 乾学院

http://c.raqsoft.com.cn

优惠价购买请加入 好多乾

http://sys.misdiy.com/hdq.html

www.raqsoft.com.cn