

友乾营 · 第十二期

存储过程的
得与失

I 从一条SQL编写规范说起

1. 【强制】禁止使用存储过程，存储过程难以调试和扩展，更没有移植性。
2. 【强制】数据订正（特别是删除、修改记录操作）时，要先select，避免出现误删除，确认无误才能执行更新语句。
3. 【推荐】in操作能避免则避免，若实在避免不了，需要仔细评估in后边的集合元素数量，控制在1000个之内。
4. 【参考】如果有国际化需要，所有的字符存储与表示，均以utf-8编码，注意字符统计函数的区别。

阿里巴巴SQL编写规范

I 公认度高的存储过程优缺点

优点

1. 数据与业务分离

业务无需关心数据，直接调用存储过程；数据处理也无需在业务端完成

2. 高性能

存储过程采用预编译机制

3. 更安全

控制访问权限；防止SQL注入

缺点

1. 开发调试性差

无有效开发调试功能

2. 维护性差

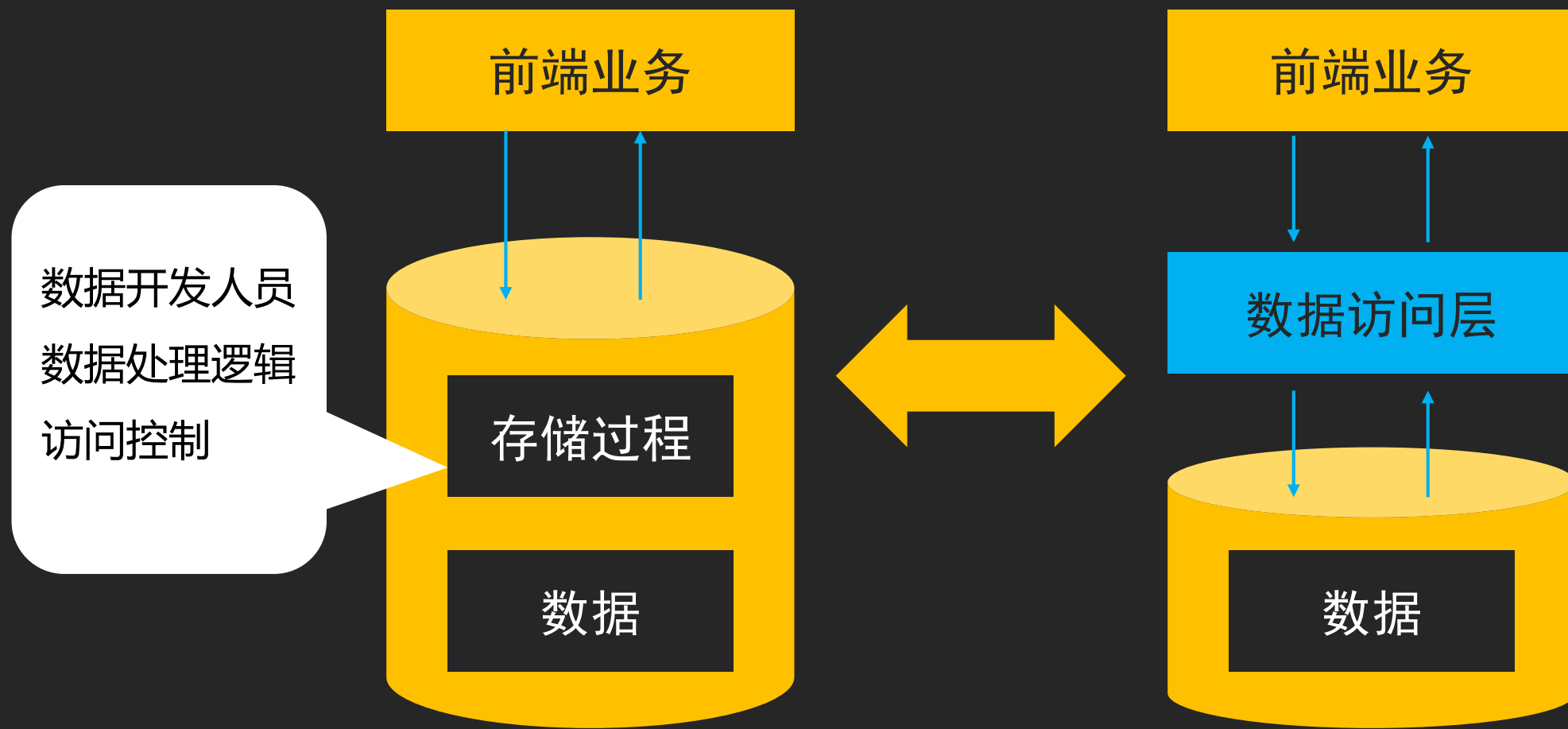
存储过程过长难以理解和维护
业务变化存储过程需要频繁修改

3. 移植性差

分库没法用；换库要重写

I 数据与业务分离

友乾营



I 数据与业务分离

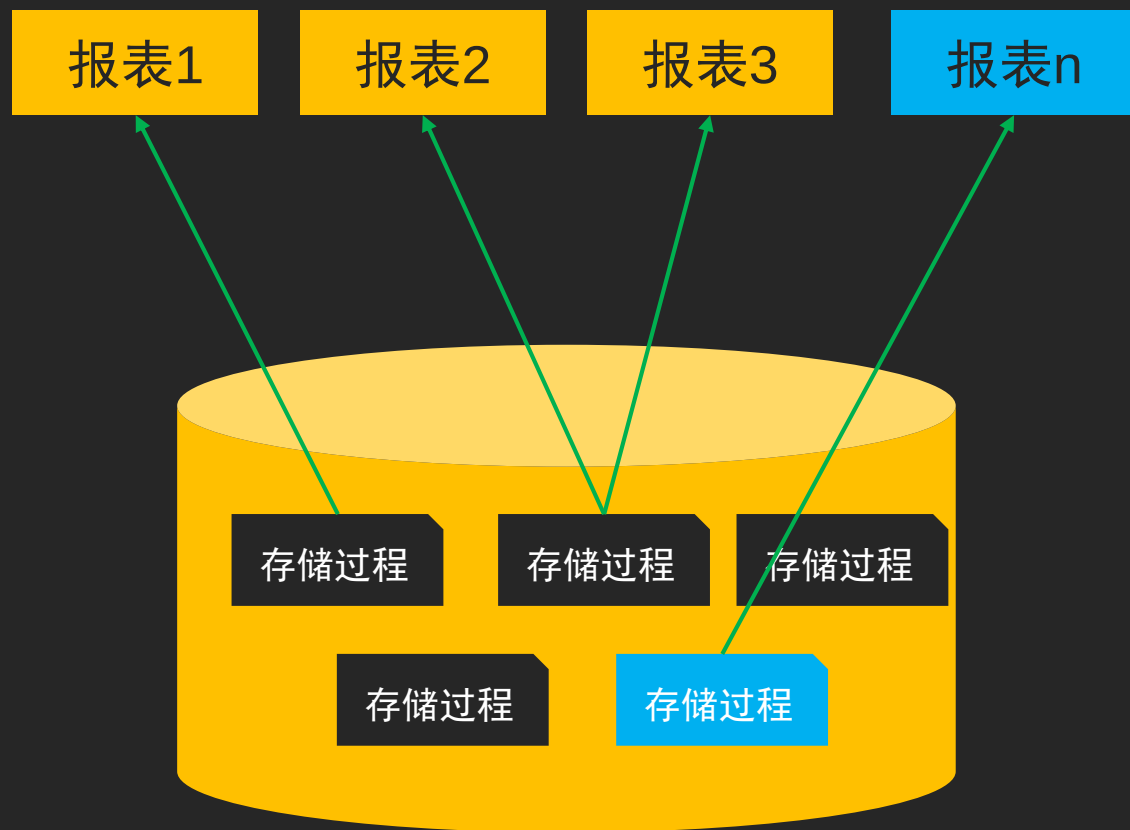
批量结构化数据计算能力

SQL 很强

JAVA 很弱

存储过程实现后台数据逻辑的优势是 SQL 的集合运算能力支撑的，它主要来自于开发便捷，而不是应用结构

I 能不能分离？要看场景



- 报表的数据计算逻辑在存储过程中实现
- 报表与存储过程分别在两个地方存放
- 报表业务不稳定，经常增改，要同步修改存储过程
- 可能涉及不同的人员
- 同一个存储过程可能会被其他报表调用
- 存储过程越来越多，无法删除，影响数据库容量

用存储过程实现报表数据源会破坏应用的模块结构，增大应用的耦合度

I 高性能

预编译机制让存储过程跑得更快？

对于分析型这类计算密集型任务，相对计算时间，预编译的时间甚至可以忽略不计



存储过程之所以比JAVA从数据库读数计算快，在于存储过程计算时，**数据不出库**

I 高性能

存储过程本身性能并不好

【测试】

针对大表的字段求和（一句 SQL 可以完成的运算）。使用存储过程逐行取数计算，与SQL和JAVA（基于文件）比较

存储过程 < SQL

存储过程比SQL慢一个数量级

存储过程 < JAVA

存储过程比JAVA慢很多

看场景：OLTP与OLAP不一样

I 安全性

存储过程不安全的一面

分析型场景下，原则上只需要对数据库有只读权限，但如果数据源是存储过程开发的，则需要向报表开发人员开放编译和运行存储过程的权限，这几乎可以对数据库做一切操作了，安全隐患非常大

不安全

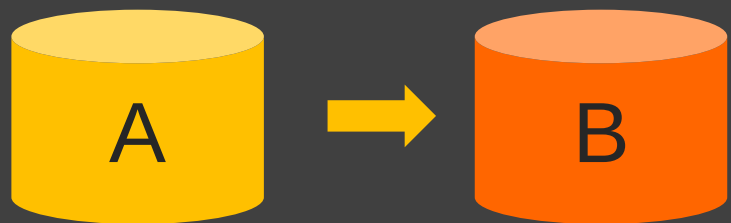


加强管理



低效率

I 可移植性



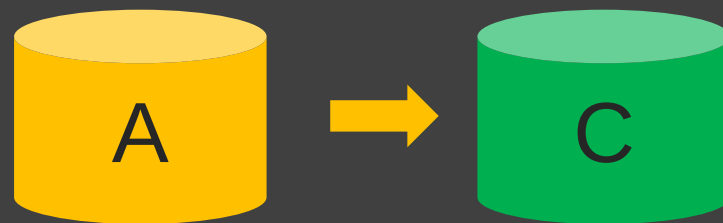
商用数据库

商用数据库

情况一

成熟商用数据库对SQL标准支持较好，
移植复杂度不是非常高

国产化时可能有些困难



商用数据库

开源数据库

情况二

开源数据库对SQL标准支持不好，移植
性很差

I 可移植性

- 大多发生在商用数据库到开源数据库上
- 这种情况下使用存储过程是致命的

观点

从商用数据库更换到开源数据库，建设成本会显著降低，但这并不是技术进步带来的结果，所以总开销并不会有太大变化，因此使用开源数据库的开发和维护成本会比较高

一个主张&我们的做法

库外存储过程

I 库外存储过程



I 库外存储过程特点

1

兼容性

支持多库、异构源计算

2

高性能

实施并行计算，利用多CPU能力

3

低耦合性

脱离数据库，与应用集成；或者作为公共数据服务层（数据访问层）

4

强移植性

计算逻辑与数据库分离，更换数据库只需修改数据库连接

I 库外存储过程特点

友乾营

5

改善数据库

提升安全性，同时减轻数据库压力

6

计算实现简单

具备丰富批量结构化数据计算类库

7

调试方便

具备丰富的编辑调试功能

8

可维护性强

代码简短、可读性好

I 开发环境

执行、调试执行、单步执行

设置断点

The screenshot displays the EsProc 3.1 interface. The top menu bar includes 文件(F), 编辑(E), 程序(P), 工具(T), 窗口(W), and 帮助(H). The toolbar contains icons for file operations and execution. The main editor shows a script with the following content:

```
A2 = 1 =file("..\\demo\\zh\\txt\\Sale.txt").import@t().select month(Datetime)==6)
.....
1 =file("..\\demo\\zh\\txt\\Stock.txt").import@t().select(month(Datetime)==6)
2 =file("..\\demo\\zh\\txt\\Sale.txt").import@t().select(month(Datetime)==6)
3 =file("..\\demo\\zh\\txt\\Storage.txt").import@t().select(month(Date)==5)
4 =file("..\\demo\\zh\\txt\\Commodity.txt").import@t()
5 '08:00:00 '21:30:00
6 =periods@d(date("2009-6-1"), date("2009-6-30"), 1)
7 =A1.align@a(A6:~,date(Datetime))
8 =A2.align@a(A6:~,date(Datetime))
9 =A4.new(ID:Commodity,8:Stock,8:GosTime,8:TotalGosTime)
10 >A9 keys(Commodity)
```

The data grid on the right shows the following data:

Datetime	Commodity	Volume
2009-06-01 08:0	20077	28
2009-06-01 08:1	20056	47
2009-06-01 08:1	20094	34
2009-06-01 08:2	20020	19
2009-06-01 08:4	20013	42
2009-06-01 08:4	20077	1
2009-06-01 08:5	20069	19
2009-06-01 09:0	20011	22
2009-06-01 09:0	20007	22

The system output window at the bottom shows the following text:

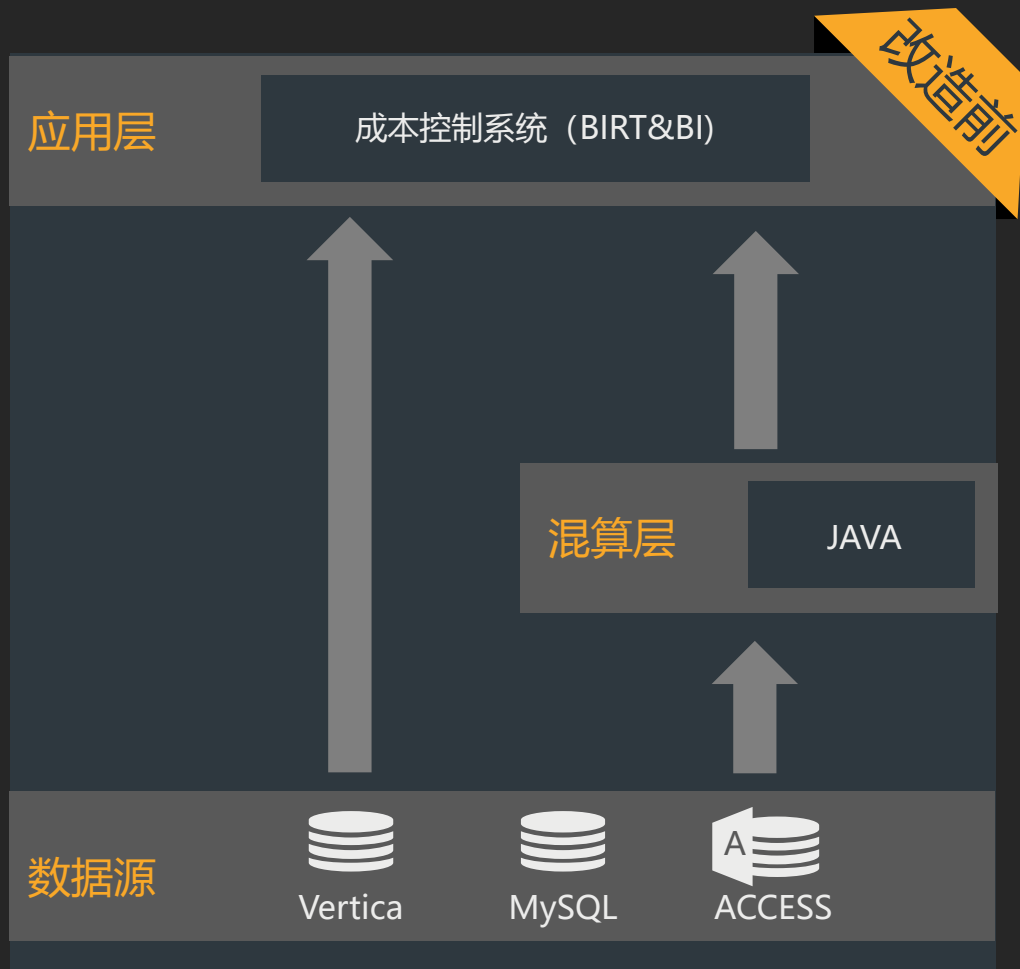
```
系统信息输出
[2017-09-28 10:38:11]
DEBUG: Esproc Function Points = 1000 0001 1111 1101
```

网格结果所见即所得，易于调试；方便引用中间结果

语法简单，符合自然思维，比其他高级开发语言更简单

系统信息输出，异常随时查看

I 一个使用实例



⚠ 尽量用单源，万不得已多源用JAVA硬编码



+ 使Vertica支持存储过程，方便跨源计算

好多乾

润乾线上直销系统



玩转好多乾

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

