

报表怎么总是没完没了?

报表做不完 · 老板一直催 · 客户总抱怨

生活总是这样无奈吗？



需求多变，报表总也做不完，累！

- 很多技术力量投入报表开发，却还是疲于应付
- 技术高手用来做报表，感觉很浪费
- 缺少低成本高效率的应对方案



无报表工具

依靠人肉硬编码
开发慢 成本高
人员要求高



有报表工具

复杂格式报表做不了
导出打印搞不定
还要靠人肉硬编码，费时费力



开源工具

报表工具花了不少钱
开发效率怎么还这么差
大牌敏捷BI系统也上过N套
竟然无济于事



商用工具

为什么呢：无关报表工具



亲，报表为什么会没完没了？

报表的业务稳定性天生很差，业务开展会不断刺激出新的查询统计需求

那就没有办法消灭没完没了的问题吗？

没有！报表多变常态，无法消除，只能适应；但可以利用工具实施低成本应对方法！

那为什么上了专业报表工具、敏捷BI还是搞不定？

报表工具只能搞定呈现环节，但消耗工作量更大却是数据准备环节，报表工具无能为力!!!

那么数据准备怎么办？

也要工具化！全方位简化报表开发后，才能做到低成本高效率应对！

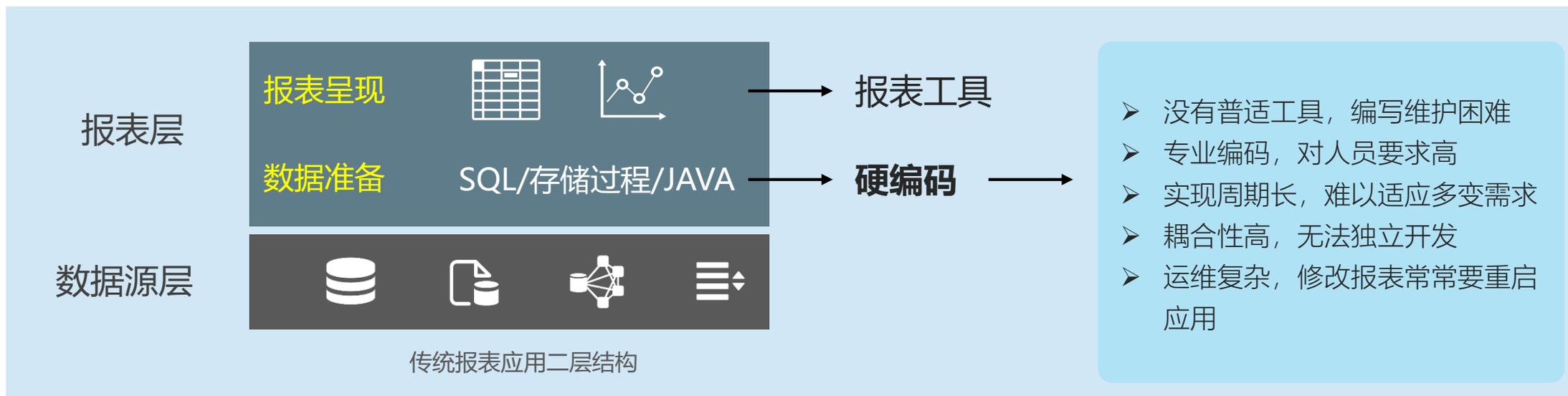
嗯嗯，谢谢亲！



为什么呢：无关报表工具



图解：报表工具只能搞定报表呈现环节，但搞不定数据准备环节



能，能解决吗？



表急，
看解决方案！

报表数据准备环节独立出来，报表开发彻底工具化，实现高效率！



解决问题
一定要从根本入手

报表呈现层



数据准备层



数据源层



新型报表应用三层结构

报表**开发工具化**

报表绘制与数据准备两个环境都可以使用高效工具开发

报表**模块独立化**

报表呈现及数据准备工作从应用程序和数据库中独立出来，单独维护，与应用解耦

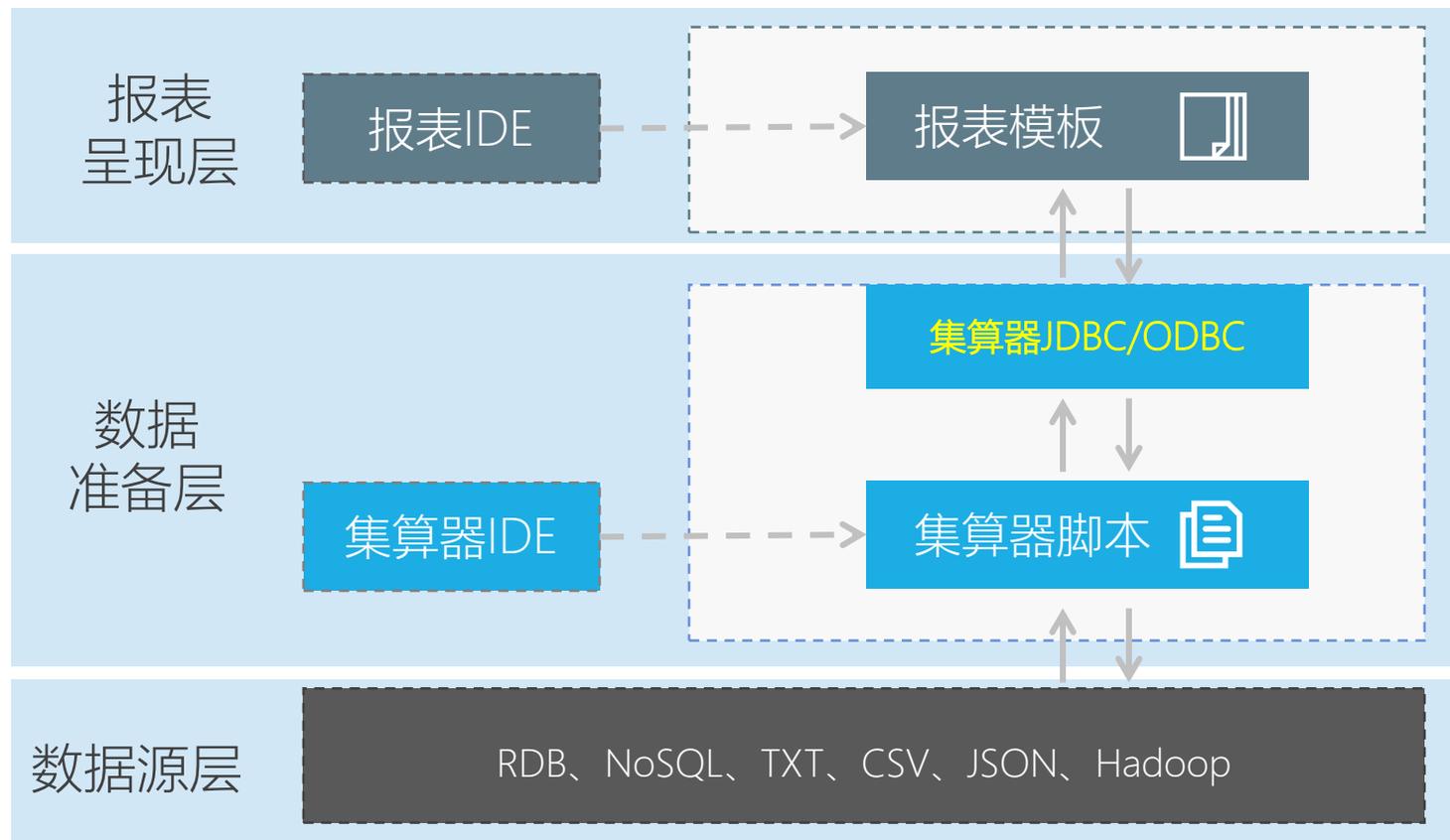
开发**人员广泛化**

一般技术人员都能做，成本低，响应快

集算器实现数据准备层



引入数据准备层后报表应用的体系结构，分成了数据准备和报表呈现两层



任意报表工具（润乾报表/BIRT/JasperReport/其他）

标准接口，直接使用

脚本解释执行，
不停机热切换，
降低运维复杂度

还真有两下子!



报表模板和计算脚本均置于应用系统之外，可**独立团队**开发，报表实施过程全面工具化

集算器—简易开发环境



支持分步计算的可视化的开发环境，适合一般技术人员使用



我发明的，
但不骄傲

执行、调试执行、单步执行 设置断点

分步计算，符合自然思维，比SQL/JAVA更简单

系统信息输出，异常随时查看

网格结果所见即所得，易于调试；方便引用中间结果

A	B	C	D
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))		

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

系统信息输出

[2017-09-28 10:38:11]
DEBUG: Esproc Function Points = 1000 0001 1111 1101



某支股票最长连续涨了多少交易日

```
1 select max(连续日数)
2 from (select count(*) 连续日数
3       from (select sum(涨跌标志) over(order by 交易日) 不涨日数
4             from (select 交易日,
5                   case when 收盘价>lag(收盘价) over(order by 交易日)
6                       then 0 else 1 end 涨跌标志
7                   from 股价表) )
8       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))

集算器SPL

分步执行的语法体系更容易描述人的思路



思考：按照自然思维怎么做？

集算器 - 完备计算类库



提供不依赖数据库的完备数据计算类库，让数据处理逻辑实现更简单

	A	B	C
1	=esProc.query("SELECT 订单ID AS 合同,订购日期 AS 日期,销售金额 AS 销售记录表")		
2	=A1.group(销售)		
3	=create(销售,今年销售额,去年销售额,客户数,大客户数)		
4	for A2	=A4(1).销售	
5		=A4.select(year(日期)==年份).sum(金额)	
6		=A4.select(year(日期)-年份-1).sum(金额)	
7		=A4.group(年份).sum(金额)	
8		=B7.count()	
9		=B7.count(~>=10000)	
10		>A3.insert(0,B4,B5,B6,B8,B9)	
11	result A3		

分组、循环

	A	B	C
1	=esProc.query("select * from 员工表")		
2	=A1.select(性别=="男")		
3	=A1.select(出生日期>=date("1970-01-01"))		
4	=A2^A3	/交运算，统计晚于1970年出生的男员工	
5	=A2&A3	/并运算，统计男员工或者晚于1970年出生的员工	
6	=A2\A3	/运算，统计早于1970年出生的男员工	
7	=A4.sum(工资)		
8	=A5.avg(年龄)		
9	=A6.sort(出生日期)		
10	/集合作为基本数据类型广泛应用		
11			

集合运算

	A	B	C
1	=file("交易记录.txt").import@t()		
2	=A1.sort(客户编码,交易日期)		
3	=A2.select(车辆型号=="捷达" 车辆型号=="迈腾").dup@t()		
4	=A3.derive(interval(交易日期[-1],交易日期),间隔)		
5	=A4.select(车辆型号[-1]=="捷达" && 车辆型号=="迈腾" && 客户编码==客户编码[-1])		
6	=A5.avg(间隔)		
7			
8			
9			
10			
11			

排序、过滤

	A	B	C
1	=esProc.query("select * from 员工表")		
2	=A1.sort(入职日期)		
3	=A2.pmin(出生日期)	/出生最早的员工的记录序号	
4	=A2(to(A3-1))	/直接用序号访问成员	
5	=esProc.query("select * from 股价表 where 股票代码='000062'")		
6	=A5.sort(交易日期)		
7	=A6.pmax(收盘价)	/收盘价最高的那条记录的序号	
8	=A6.calc(A7,收盘价/收盘价[-1]-1)		
9			
10	/直接用序号访问成员		
11			

有序集合



集算器—多样性数据源支持



- RDB: Oracle, DB2, MS SQL, MySQL, PG,
- TXT/CSV, JSON/XML, EXCEL
- Hadoop: HDFS, HIVE, HBASE, SPARK
- MongoDB, REDIS, ...
- HTTP、ALI-OTS
- ES, Kafka
-



通过集算器，报表可以直接使用这些数据源混合计算，无需事先同库

解脱报表困扰的征途

累积报表经验

建设报表知识库论坛。减少沟通中的误解，实现历史工作的复用

独立报表模块

调整应用结构，梳理报表数据源，实现报表模块与数据库及应用解耦

引入报表工具

呈现工具化
解放报表呈现阶段人力

建设报表团队

模块独立后可建设培训独立于应用开发团队外的低成本报表队伍应对多变需求

引入计算工具

数据计算工具化
解放数据准备阶段人力

他们都在这样用



某石油公司

使用集算器将报表开发由29个工作日缩短至2.25个工作日



组队就差你了

加拿大某大型保险公司

使用集算器从Vertica取数，计算后为BIRT提供数据支持，完全替代了硬编码实现过程

某大型银行

使用集算器后代码量由3500行降到80行，开发时间缩短了11倍

某国有银行

使用集算器代码量由4400行降到132行，开发时间由30人天缩短至6人天

[more >>](#)

想要了解更多 请联系我们



官方认证
如假包换



帅哥美女
看你运气

THANK YOU