



集算器

创新大数据计算引擎

解决BIRT动态数据源的若干示例

润乾软件出品





目录

Contents

1

参数决定连接数据源

2

临时拼接SQL与表达式

3

列数不定的结果集

4

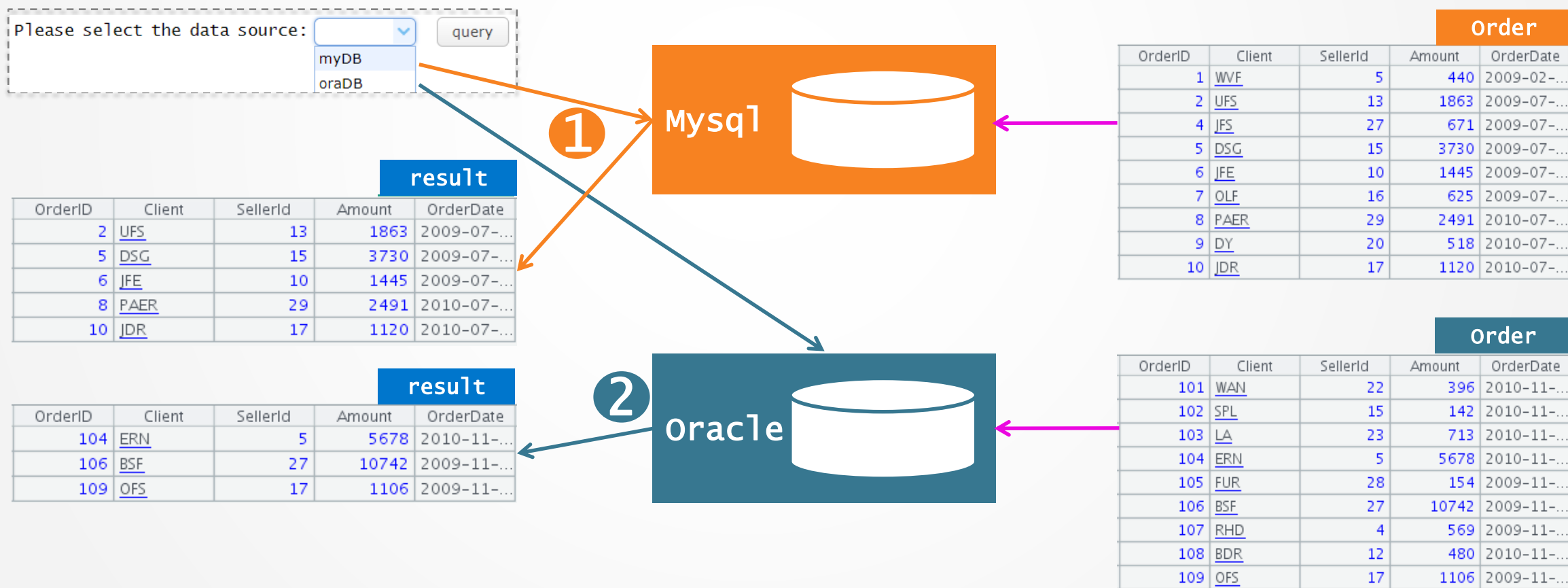
根据参数决定分组



参数决定连接数据源

有时我们需用参数动态指定数据源，或向子报表、table控件动态传入数据源名。BITR等报表工具必须要借助高级语言实现，非常麻烦

数据源myDB和oraDB分别指向不同数据库，两库中有相同结构的表Order，报表要根据参数动态连接数据源，查询Order中金额大于1000的订单





参数决定连接数据源 — 示例

集算器支持动态解析表达式，可以很好地实现这个需求

	A	B
1	<code>=\${pSource}.query("select * from Order where Amount>?",pAmount)</code>	/接收参数进行查询

pSource、pAmount都是报表参数，其中pSource是数据源名，`${...}`表示将字符串解析为表达式。

当pSource="myDB"时，A1的计算结果如右图：

OrderID	Client	SellerId	Amount	OrderDate
2	UFS	13	1863	2009-07-...
5	DSG	15	3730	2009-07-...
6	JFE	10	1445	2009-07-...
8	PAER	29	2491	2010-07-...
10	JDR	17	1120	2010-07-...

当pSource="oraDB"时，A1的计算结果如右图：

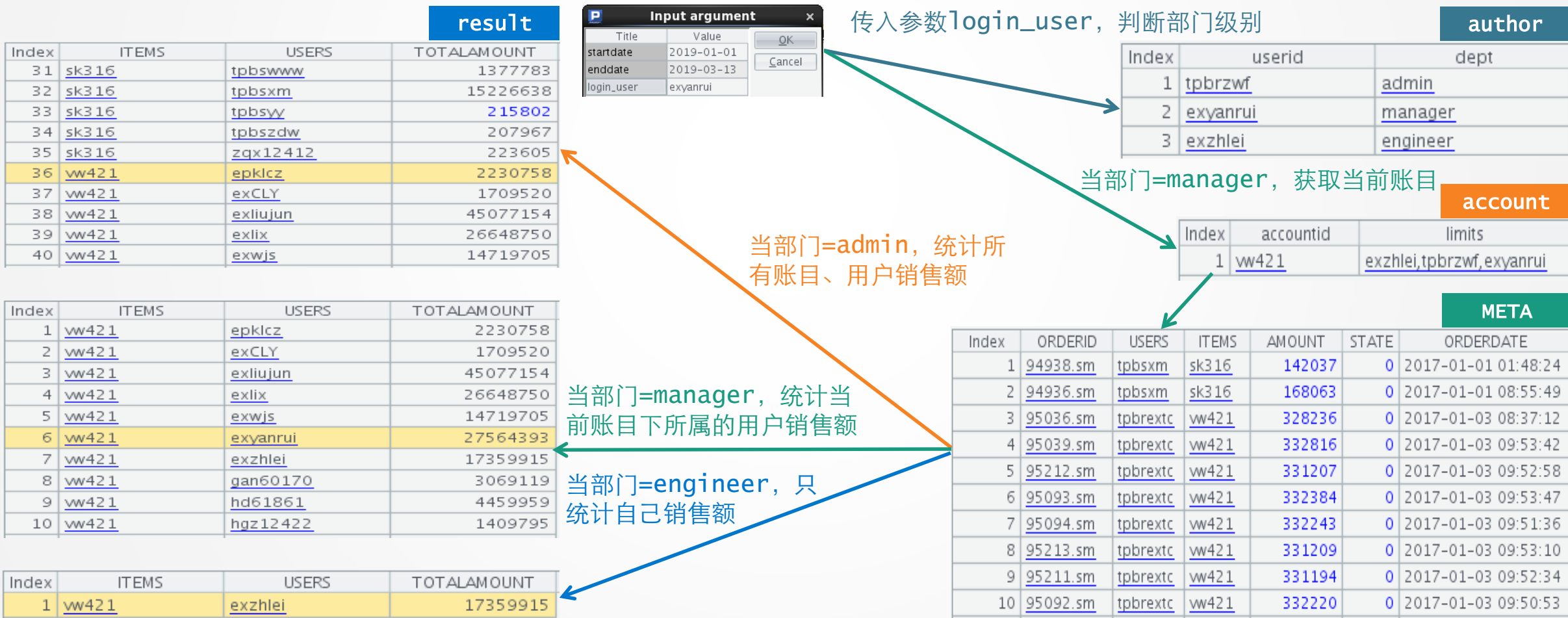
OrderID	Client	SellerId	Amount	OrderDate
104	ERN	5	5678	2010-11-...
106	BSF	27	10742	2009-11-...
109	OFS	17	1106	2009-11-...



临时拼接SQL与表达式 - 权限查询

根据传入的用户名，分级别查询出不同的订单信息。

author存储用户部门信息， account存储账目信息(多个用户可归属于一个账目)， META存储订单明细



临时拼接SQL与表达式 - 权限查询 - 示例



问题实质是根据权限级别动态拼接SQL条件，这在自助查询场景中很常见。使用外部程序或存储过程的代码难写、工作量大；引入集算器，能用较少代码就能完成此逻辑判断；其特有的宏机制极大地提高了代码复用程度

	A	B	C
1	=connect("myDB")	=sql="\select ITEMS,USERS,sum(AMOUNT) as TOTALAMOUNT from META where ORDERDATE>=? and ORDERDATE<=?"	/定义初始sql
2	=flag=A1.query("select dept from author where userid=?",login_user).dept		/用登录用户查询部门
3	if flag=="admin"	>B1=concat(sql," group by ITEMS,USERS\")(/若是admin查看所有
4	else if flag=="manager"	>B1=concat(sql," and ITEMS='",A1.query("select accountid from account where find_in_set(?,limits)",login_user).accountid," group by ITEMS,USERS\")(/若是manager, 则能查看当前项目下的每个用户的销售额
5	else	>B1=concat(sql," and USERS='",login_user," group by ITEMS,USERS\")(/普通用户只能看自己的销售额
6	=A1.query@x(\${B1},startdate,enddate)		/执行拼接好的sql

B3执行后B1拼接的SQL

```
Value
"select ITEMS,USERS,sum(AMOUNT) as TOTALAMOUNT from META where ORDERDATE>=? and ORDERDATE<=? group by ITEMS,USERS"
```

B4执行后B1拼接的SQL

```
Value
"select ITEMS,USERS,sum(AMOUNT) as TOTALAMOUNT from META where ORDERDATE>=? and ORDERDATE<=? and ITEMS='vw42 1' group by ITEMS,USERS"
```

B5执行后B1拼接的SQL

```
Value
"select ITEMS,USERS,sum(AMOUNT) as TOTALAMOUNT from META where ORDERDATE>=? and ORDERDATE<=? and USERS='exzhlei' group by ITEMS,USERS"
```



临时拼接SQL与表达式 – 拼接动态表名

表A、B、C，其中A表中存放了其它表的表名，根据A表中记录的TableName查询对应ID在“TableName表”中的值

result		
Index	ID	val
1	<u>01</u>	<u>13</u>
2	<u>02</u>	<u>92</u>
3	<u>03</u>	<u>15</u>



B table		
Index	ID	Num
1	<u>01</u>	<u>13</u>
2	<u>02</u>	<u>14</u>
3	<u>03</u>	<u>15</u>

A table		
Index	ID	TableName
1	<u>01</u>	<u>b</u>
2	<u>02</u>	<u>c</u>
3	<u>03</u>	<u>b</u>

C table		
Index	ID	Num
1	<u>01</u>	<u>91</u>
2	<u>02</u>	<u>92</u>
3	<u>03</u>	<u>93</u>

用外部程序来构造动态SQL非常繁琐。集算器能很好地支持拼接操作

	A	B
1	=connect("myDB")	/连接数据库
2	=A1.query("select * from A")	/查询A表形成序表
3	=A2.new(ID,A1.query("select * from "+TableName+" where ID ='"+ID+"'").Num:val)	/根据A2每条记录的TableName列动态拼sql并查询



临时拼接SQL与表达式 – 带in条件查询

BIRT不直接支持数组参数参与查询，因而很难处理带in的查询，无法把数组传给DataSet执行带in的SQL

间接方式需要编写复杂的JS脚本做sql拼接，且要根据in中内容的数据类型决定是否拼上引号等，很麻烦

集算器可以直接将数组参数传递给SQL进行查询，最后返回结果给BIRT，非常简单：

	A	B
1	=myDB.query("SELECT * FROM SALES WHERE ORDERID IN (?)", arg1.array())	/传递arg1数组参数，查询数据
2	return A1	/返回查询结果

A1的查询结果

Index	ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE
1	1	UJRNP	17	392.0	2012-11-02 ...
2	2	SJCH	6	4802.0	2012-11-09 ...
3	3	UJRNP	16	13500.0	2012-11-05 ...
4	4	PWQ	9	26100.0	2012-11-08 ...

参数转换为数组

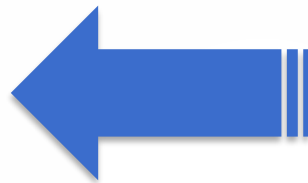
Title	Value
arg1	1,2,3,4

临时拼接SQL与表达式 – 动态解析表达式



在Mysql数据库的tmp表中，其中Size字段的值以字符串形式存储，有时它是三个数字的乘积，有时只是一个数字。我们希望想将此列统一转换为整数值。Mysql虽然可以很好的分割字符串，但却很难计算它

		result
Index	Size	
1		150
2		20
3		3
4		60



		源数据
Index	Size	
1	<u>15X10</u>	
2	<u>5X4</u>	
3	<u>3</u>	
4	<u>2X6X5</u>	

对于任意的四则运算，带多重括号且多个操作数的运算，用SQL很难实现。可以在SPL中用动态计算来解决：

	A	B
1	=myDB.query("select * from tmp")	/查询tmp表
2	>A1.run(Size=eval(replace(Size,"x","*")))	/将字符X替换为乘法符号*，eval函数可对字符串动态解析为表达式并计算



列数不定的结果集

根据传入时间参数生成该时间当月月初至当前的时间数据，比如传入1996-01-03时，生成（1996-01-01、1996-01-02、1996-01-03）三个列，传入1996-01-08则生成8个日期列

result1

Index	O_ORDERSTATUS	O_ORDERPRIOR...	1992-01-01	1992-01-02	1992-01-03
1	F	<u>1-URGENT</u>	1255.35	4169.57	4193.39
2	F	<u>2-HIGH</u>	3778.66	9744.64	6329.68
3	F	<u>3-MEDIUM</u>	6484.37	1531.02	3124.04
4	F	<u>4-NOT SPECIFIED</u>	5038.66	5914.95	(null)
5	F	<u>5-LOW</u>	1253.62	3980.69	1463.16

Input argument

Title	Value
d_date	1992-01-03

OK Cancel

传入日期参数: 1996-01-03

传入日期参数: 1996-01-08

result2

Index	O_ORDE...	O_ORDERPRIORI...	1992-01-01	1992-01-02	1992-01-03	1992-01-04	1992-01-05	1992-01-06	1992-01-07	1992-01-08
1	F	<u>1-URGENT</u>	1255.35	4169.57	4193.39	5538.6	7838.3	3008.37	4082.05	1816.65
2	F	<u>2-HIGH</u>	3778.66	9744.64	6329.68	5260.68	1587.26	8279.45	4737.0	3239.8
3	F	<u>3-MEDIUM</u>	6484.37	1531.02	3124.04	3724.23	8125.21	4037.81	1894.09	(null)
4	F	<u>4-NOT SPECIFIED</u>	5038.66	5914.95	1944.85	2429.96	2571.5	4746.42	4642.6	(null)
5	F	<u>5-LOW</u>	1253.62	3980.69	1463.16	3093.44	2323.7	3955.63	5837.97	2588.15

Input argument

Title	Value
d_date	1992-01-08

OK Cancel

源数据

Index	O_ORDERKEY	O_ORDERSTATUS	O_ORDERPRIO...	O_ORDERDATE	O_TOTALPRICE
1	1	O	<u>5-LOW</u>	1996-01-02	1736.65
2	2	O	<u>1-URGENT</u>	1996-12-01	469.29
3	3	F	<u>5-LOW</u>	1993-10-14	1938.46
4	4	O	<u>5-LOW</u>	1995-10-11	321.52
5	5	F	<u>5-LOW</u>	1994-07-30	1446.59
6	6	F	<u>4-NOT SPECIFI...</u>	1992-02-21	587.5
7	7	O	<u>2-HIGH</u>	1996-01-10	2520.04
8	32	O	<u>2-HIGH</u>	1995-07-16	2086.61
9	33	F	<u>3-MEDIUM</u>	1993-10-27	1632.44
10	34	O	<u>3-MEDIUM</u>	1998-07-21	589.5
11	35	O	<u>4-NOT SPECIFI...</u>	1995-10-23	2537.25
12	36	O	<u>1-URGENT</u>	1995-11-03	682.9
13	37	F	<u>3-MEDIUM</u>	1992-06-03	2066.81
14	38	O	<u>4-NOT SPECIFI...</u>	1996-08-21	825.0
15	39	O	<u>3-MEDIUM</u>	1996-09-20	3417.34
16	64	F	<u>3-MEDIUM</u>	1994-07-16	394.15
17	65	P	<u>1-URGENT</u>	1995-03-18	1106.44
18	66	F	<u>5-LOW</u>	1994-01-20	1037.41
19	67	O	<u>4-NOT SPECIFI...</u>	1996-12-19	1694.05
20	68	O	<u>3-MEDIUM</u>	1998-04-18	3307.94



列数不定的结果集 — 示例

动态列结果集用SQL很难写出来，而集算器能够很灵活地实现动态效果，只需编写简单的代码

	A	B
1	<code>=myDB.query("select * from ORDERS where O_ORDERDATE>=? and O_ORDERDATE<=?", pdate@m(d_date), d_date)</code>	/根据d_date参数过滤查询数据集
2	<code>={\${"create(O_ORDERSTATUS,O_ORDERPRIORITY,"+periods(pdate@m(d_date),d_date,1).concat@c()+")"}</code>	/根据参数创建空序表
3	<code>>A1.group(O_ORDERSTATUS,O_ORDERPRIORITY).run(A2.record(~.O_ORDERSTATUS ~.O_ORDERPRIORITY ~.group(O_ORDERDATE).(~.sum(O_TOTALPRICE))))</code>	/对A1分组汇总，处理后的序列结果集再写入到A2中

A3分组的执行过程

Index	Member
1	[[1248,F,1-URGENT, ...],[3271,F,1-URGENT, ...],[36582,F,1-URGEN...
2	[[3712,F,2-HIGH, ...],[22693,F,2-HIGH, ...],[23010,F,2-HIGH, ...]
3	[[3139,F,3-MEDIUM, ...],[29030,F,3-MEDIUM, ...],[37543,F,3-MEDIU...
4	[[5607,F,4-NOT SPECIFIED, ...],[6657,F,4-NOT SPECIFIED, ...],[20742...
5	[[24167,F,5-LOW, ...],[27015,F,5-LOW, ...],[51648,F,5-LOW, ...]

Index	O_ORDERKEY	O_ORDERS...	O_ORDERP...	O_ORDERD...	O_TOTALPR...
1	1248	F	1-URGENT	1992-01-02	2620.81
2	3271	F	1-URGENT	1992-01-01	1255.35
3	36582	F	1-URGENT	1992-01-03	1869.33
4	47591	F	1-URGENT	1992-01-03	1360.57
5	69378	F	1-URGENT	1992-01-02	1548.76
6	83328	F	1-URGENT	1992-01-03	963.49

Index	O_ORDERKEY	O_ORDERS...	O_ORDERP...	O_ORDERD...	O_TOTALPR...
1	3712	F	2-HIGH	1992-01-02	1811.75
2	22693	F	2-HIGH	1992-01-03	2473.41
3	23010	F	2-HIGH	1992-01-01	973.73
4	27137	F	2-HIGH	1992-01-01	2804.93
5	63906	F	2-HIGH	1992-01-02	3285.97
6	80069	F	2-HIGH	1992-01-03	1405.16
7	90690	F	2-HIGH	1992-01-02	2605.68
8	108992	F	2-HIGH	1992-01-02	2041.24
9	112645	F	2-HIGH	1992-01-03	2451.11



根据参数决定分组 — 统计图相关运算

折线图X轴以时间为刻度，要根据不同参数条件显示不同值。但BIRT统计图上没法实现条件不一样时合计的数据跟着变化效果，一般需要外部程序修改X轴的刻度，代码非常繁琐

参数：day，显示24小时，间隔1小时一条

Index	Member	Value
1	2015-03-12 09:23:21	83.0625
2	2015-03-12 08:23:21	19.7053
3	2015-03-12 07:23:21	19.9969
4	2015-03-12 06:23:21	149.2472
5	2015-03-12 05:23:21	64.6114
6	2015-03-12 04:23:21	41.2842
7	2015-03-12 03:23:21	188.6375
8	2015-03-12 02:23:21	201.9394
9	2015-03-12 01:23:21	21.4358
10	2015-03-12 00:23:21	16.9647
11	2015-03-11 23:23:21	76.0822
12	2015-03-11 22:23:21	115.6825
13	2015-03-11 21:23:21	11.1431
14	2015-03-11 20:23:21	45.7211
15	2015-03-11 19:23:21	198.4781
16	2015-03-11 18:23:21	87.4881
17	2015-03-11 17:23:21	17.7075
18	2015-03-11 16:23:21	48.8608
19	2015-03-11 15:23:21	64.4744
20	2015-03-11 14:23:21	8.9958
21	2015-03-11 13:23:21	30.7856
22	2015-03-11 12:23:21	71.4450
23	2015-03-11 11:23:21	164.5139



参数：hour，显示12条，间隔5分钟一条

Index	Member	Value
1	2015-03-12 10:24:00	(null)
2	2015-03-12 10:19:00	0.9528
3	2015-03-12 10:14:00	(null)
4	2015-03-12 10:09:00	2.3806
5	2015-03-12 10:04:00	3.9444
6	2015-03-12 09:59:00	61.8811
7	2015-03-12 09:54:00	0.9956
8	2015-03-12 09:49:00	4.9547
9	2015-03-12 09:44:00	(null)
10	2015-03-12 09:39:00	5.6578
11	2015-03-12 09:34:00	2.2956
12	2015-03-12 09:29:00	(null)

参数：week，显示7条，间隔1天一条

Index	Member	Value
1	2015-03-11 10:31:02	1767.9681
2	2015-03-10 10:31:02	1274.7014
3	2015-03-09 10:31:02	1905.5183
4	2015-03-08 10:31:02	2184.4608
5	2015-03-07 10:31:02	1985.4283
6	2015-03-06 10:31:02	2053.1322
7	2015-03-05 10:31:02	1872.7825

根据参数决定分组 — 统计图相关运算 — 示例

报表工具擅长处理一致的规则，而这种不一致的动态运算很，更好的做法是先写段程序把数据源准备好再去画图。集算器拥有丰富的集合运算可以方便地完成这类计算

	A	B	C
1	=[]		/需要生成的时间间隔数据集
2	if(type=="hour")	>A1=12.(elapsed@s(now(),-5*60*~))	/统计最近一小时，每5分钟一条，生成12条时间间隔
3	else if(type=="day")	>A1=24.(elapsed@s(now(),-60*60*~))	/统计最近一天，每小时一条，生成24条时间间隔
4	else if(type=="week")	>A1=7.(elapsed(now(),-~))	/统计最近一周，每天一条，生成7条时间间隔
5	=A1.(demo.query("select sum(v) from tv where t>? and t<=?",~, ifn(~[-1],now()))).new(~.v:value)		/按照时间间隔，分别查询统计求和

B2执行后A1的结果

Index	Member
1	2015-03-12 10:24:00
2	2015-03-12 10:19:00
3	2015-03-12 10:14:00
4	2015-03-12 10:09:00
5	2015-03-12 10:04:00
6	2015-03-12 09:59:00
7	2015-03-12 09:54:00
8	2015-03-12 09:49:00
9	2015-03-12 09:44:00
10	2015-03-12 09:39:00
11	2015-03-12 09:34:00
12	2015-03-12 09:29:00

B4执行后A1的结果

Index	Member
1	2015-03-11 10:31:02
2	2015-03-10 10:31:02
3	2015-03-09 10:31:02
4	2015-03-08 10:31:02
5	2015-03-07 10:31:02
6	2015-03-06 10:31:02
7	2015-03-05 10:31:02

B3执行后A1的结果

Index	Member
1	2015-03-12 09:23:21
2	2015-03-12 08:23:21
3	2015-03-12 07:23:21
4	2015-03-12 06:23:21
5	2015-03-12 05:23:21
6	2015-03-12 04:23:21
7	2015-03-12 03:23:21
8	2015-03-12 02:23:21
9	2015-03-12 01:23:21
10	2015-03-12 00:23:21
11	2015-03-11 23:23:21
12	2015-03-11 22:23:21
13	2015-03-11 21:23:21
14	2015-03-11 20:23:21
15	2015-03-11 19:23:21
16	2015-03-11 18:23:21
17	2015-03-11 17:23:21
18	2015-03-11 16:23:21
19	2015-03-11 15:23:21
20	2015-03-11 14:23:21
21	2015-03-11 13:23:21
22	2015-03-11 12:23:21
23	2015-03-11 11:23:21



根据参数决定分组 — 动态日期分组

某企业报表按照开始日期、结束日期统计一段时间内订单，但要根据时间段长短实现按天、按周、按月、按年动态分组。

`var diff = (结束日期 - 开始日期) --相隔的天数`

Index	BeginDate	TotalOrder	TotalOrderAmount
1	2012-07-05	341	5498974.0
2	2013-07-05	491	8392346.0
3	2014-07-05	1	20400.0

Index	BeginDate	TotalOrder	TotalOrderAmount
1	2012-07-05	23	284883.0
2	2012-08-05	26	270621.0
3	2012-09-05	2	39596.0

Index	BeginDate	TotalOrder	TotalOrderAmount
1	2012-07-05	6	44951.0
2	2012-07-12	5	86892.0
3	2012-07-19	5	69952.0
4	2012-07-26	1	7719.0

Index	BeginDate	TotalOrder	TotalOrderAmount
1	2012-07-05	1	1161.0
2	2012-07-06	2	17300.0
3	2012-07-07	1	10260.0
4	2012-07-08	1	11634.0
5	2012-07-09	1	4596.0
6	2012-07-10	1	44499.0

ORDERS			
Index	ORDERID	ORDERDATE	ORDERAMOUNT
1	10249	2012-07-05	1161.0
2	10250	2012-07-08	13166.0
3	10251	2012-07-08	4134.0
4	10252	2012-07-09	10260.0
5	10253	2012-07-10	11634.0
6	10254	2012-07-11	4596.0
7	10255	2012-07-12	44499.0
8	10256	2012-07-15	2794.0
9	10257	2012-07-16	24573.0
10	10258	2012-07-17	14051.0
11	10259	2012-07-18	975.0
12	10260	2012-07-19	5509.0
13	10261	2012-07-19	610.0
14	10262	2012-07-22	14487.0
15	10263	2012-07-23	43818.0
16	10264	2012-08-20	1101.0
17	10265	2012-07-25	5528.0
18	10266	2012-07-26	7719.0
19	10267	2012-07-29	20858.0
20	10268	2012-07-30	19887.0

① if (diff>365) 按年分组

② if (diff>30) 按月分组

③ if (diff>15) 按周分组

④ if (diff<15) 按天分组

根据参数决定分组 – 动态日期分组 – 示例



这个问题实质上还是数据准备，但SQL或scripted data sources的方式代码难写，工作量大；在BIRT中引入集算器后，其敏捷语法体系仅需很少代码就能完成

	A	B	C
1	=[]		/A1存储生成时间间隔的序列
2	=demo.query@x("select ORDERID,ORDERDATE,ORDERAMOUNT from ORDERS where ORDERDATE>=? and ORDERDATE<=?",startDate,endDate)		/从ORDERS表中查询出统计开始日期到结束日期之间的 订单数据，startDate和endDate是日期参数
3	=interval(startDate,endDate)		/计算开始日期和结束日期的间隔天数
4	if A3>365	>A1=startDate A3.(elapse@y(startDate,~))	/若天数间隔大于365天，则按年分组
5	else if A3>30	>A1=startDate A3.(elapse@m(startDate,~))	/若天数间隔大于30&小于等于365天，则按月分组
6	else if A3>15	>A1=startDate A3.(elapse(startDate,7*~))	/若天数间隔大于15&小于等于30天，则按周分组
7	else	>A1=startDate A3.(elapse(startDate,~))	/若天数间隔小于15天，则按天分组
8	=A2.group(A1.pseg(ORDERDATE);~.count(ORDERID):TotalOrder,round(~ .sum(ORDERAMOUNT),2):TotalOrderAmount,A1(#):BeginDate)		/按A1区间对A2分组，统计出订单总数，订单金额总 数并保留两位小数
9	=A8.new(BeginDate:BeginDate,#2:TotalOrder,#3:TotalOrderAmount)		/取出A8中需要的数据列，生成新的结果序表
10	return A9		/返回结果集给BIRT

创新技术

推动应用进步!

