SPL Base

集算器教案

序运算

目录

CONTENTS

目录

CONTENTS

SPL

04 循环计算

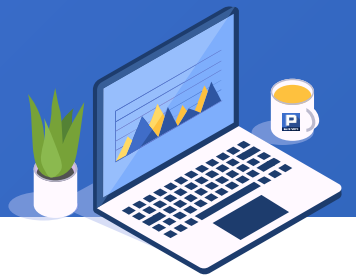
05 定位

06 选出

CONTENTS

01 序号访问

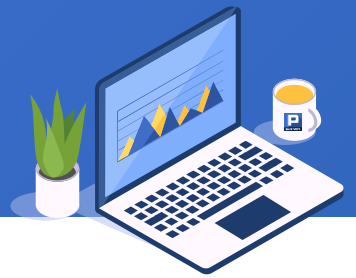
1. 单序号访问



求2019年上证指数第一个交易日和最后一个交易日的交易信息。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

1. 单序号访问

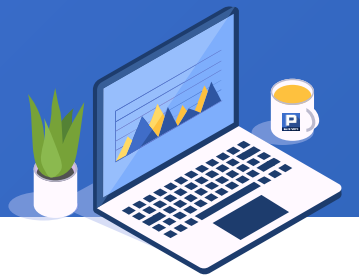


SPL如下，其中使用了A()和A.m()来获取成员：

	A	B
1	=file("000001.csv").import@ct()	/导入数据文件
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年的记录并按日期排序
3	=A2(1)|A2.m(-1)	/取出股市第一个和最后一个交易日的信息

A3	Date	Open	Close	Amount
	2019/01/02	2497.8805	2465.291	9.76E10
	2019/12/31	3036.3858	3050.124	2.27E11

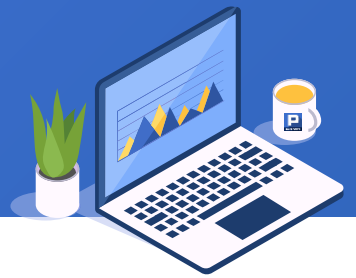
1. 单序号访问



以员工表为例，统计[California, Texas, New York, Florida]各州的平均工资。其他地区的员工存放到新组统计。

ID	NAME	STATE	SALARY
1	Rebecca	California	7000
2	Ashley	New York	11000
3	Rachel	New Mexico	9000
4	Emily	Texas	7000
5	Ashley	Texas	16000
…	…	…	…

1. 单序号访问

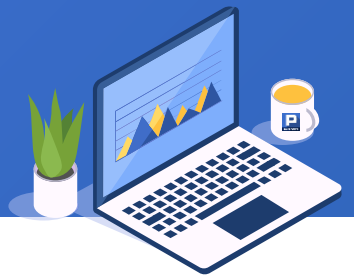


SPL如下，其中用到了A.p(-1) 获取最后一个成员的序号：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from EMPLOYEE")	/查询雇员表
3	[California,Texas,New York,Florida]	/创建地区序列
4	=A2.align@an(A3,STATE)	/雇员表按地区对位分组，@a选项每组返回所有匹配成员，@n选项不匹配成员存放到新组。
5	=A4.new(if (#>A3.p(-1),"Other",STATE):STATE,~.avg(SALARY):AvgSalary)	/统计每组的平均工资，产生新序表。使用A.p(-1)获取最后一个成员的序号，把最后一组的地区更名为Other。

A5	STATE	SALARY
	California	7700.0
	Texas	7592.59
	New York	7677.77
	Florida	7145.16
	Other	7308.1

2. 多序号访问



有一个记录日常考勤信息的表，如下图：

Per_Code	in_out	Date	Time	Type
1110263	1	2013-10-11	09:17:14	In
1110263	6	2013-10-11	11:37:00	Break
1110263	5	2013-10-11	11:38:21	Return
1110263	0	2013-10-11	11:43:21	NULL
1110263	6	2013-10-11	13:21:30	Break
1110263	5	2013-10-11	14:25:58	Return
1110263	2	2013-10-11	18:28:55	Out

每七条数据为一组，想要转换成如下结果：

Per_Code	Date	In	Out	Break	Return
1110263	2013-10-11	9:17:14	18:28:55	11:37:00	11:38:21
1110263	2013-10-11	9:17:14	18:28:55	13:21:30	14:25:58

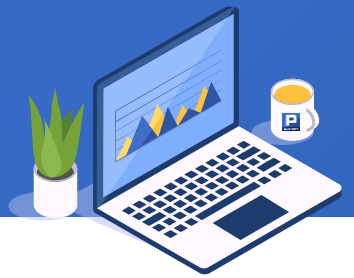
2. 多序号访问

先创建目标数据结构，再把每七条记录整理成需要的结构填入数据。SPL如下，其中用到了A()和A.m()多序号访问成员：

	A	B
1	=connect("db")	/连接数据源
2	=A1.query("select * from DailyTime order by Per_Code,Date,Time")	/查询数据，并按人员编号、日期和时间排序
3	=A2.group(Per_Code,Date)	/按人员编号和日期分组
4	=create(Per_Code,Date,In,Out,Break,Return)	/创建一个存放最后结果的空序表
5	=A3.(~([1,7,2,3,1,7,5,6]))	/对每个组，使用A([1,7,2,3,1,7,5,6])依次取出记录，这就是有序的全天记录。
6	=A5.conj([~.Per_Code,~.Date]|~.(Time).m([1,2,3,4])|[~.Per_Code,~.Date]|~.(Time).m([5,6,7,8]))	/将每条记录的数据全部整理到一个序列中。其中用到了A.m()访问多个成员。
7	>A4.record(A6)	/将记录添加到A4创建的序表中。

A4	Per_Code	Date	In	Out	Break	Return
	1110263	2013-10-11	09:17:14	18:28:55	11:37:00	11:38:21
	1110263	2013-10-11	09:17:14	18:28:55	13:21:30	14:25:58
	…	…	…	…	…	…

2. 多序号访问

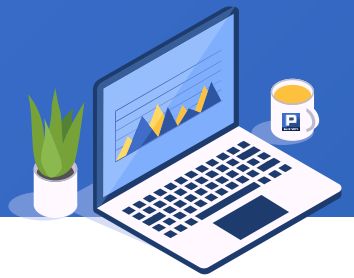


求100以内质数。SPL如下，其中用到了step()函数按固定跨度取成员：

	A	B
1	=to(100)	/创建1到100序列
2	=to(2,10)	/创建2到10序列
3	=A2.(A1.step(~,~*2))	/针对A2中每一个成员，求出它在100以内的n倍数(n>1)
4	=A1.to(2,)\A3.conj()	除去1和所有100以内的合数即为100以内的质数，其中A3.conj()求出100以内的合数

A4	Member
	2
	3
	5
	7
	11
	…

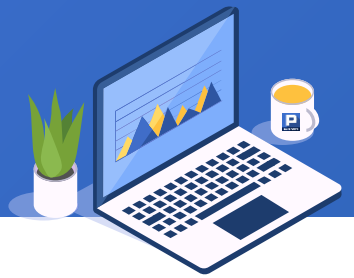
2. 多序号访问



求上证指数2019年最后10个交易日收盘价较前日的涨幅。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

2. 多序号访问

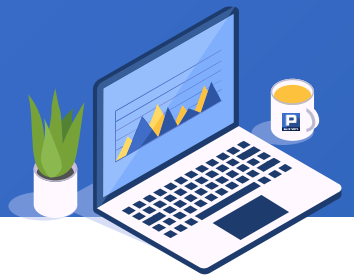


SPL如下，其中用到了A.p()函数返回最后10个成员的序号：

	A	B
1	=file("000001.csv").import@ct()	/导入数据文件
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年的记录并按日期排序
3	=A2.p(to(-10,-1))	/使用p()函数返回最后10个成员的序号
4	=A3.new(A2(~).Date:Date, string(A2(~).Close/A2(~-1).Close-1, "0.000%"):Increase)	/循环计算每个交易日收盘价与前一个交易日的涨幅

A4	Date	Increase
	2019/12/18	-0.178%
	2019/12/19	0.001%
	2019/12/20	-0.402%
	2019/12/23	-1.404%
	2019/12/24	0.673%
	…	…

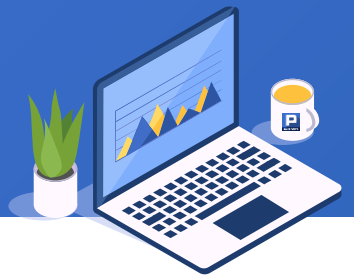
2. 多序号访问



下面是某公司组织结构表，查询北京市场调研小组的所有上级机构。

ID	ORG_NAME	PARENT_ID
1	Head Office	0
2	Beijing Branch Office	1
3	Shanghai Branch Office	1
4	Chengdu Branch Office	1
5	Beijing R&D Center	2
…	…	…

2. 多序号访问



SPL如下，其中用到了rvs()函数使序列倒置：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Organization")	/查询组织结构表
3	>A2.switch(PARENT_ID,A2:ID)	/将PARENT_ID外键映射到该ID所在的记录，实现自连接。
4	=A2.select@1(ORG_NAME=="Beijing Market Research Team")	/选出北京市场调研小组所在记录
5	=A4.prior(PARENT_ID)	/使用prior函数查出上级单位
6	=A5.rvs().(ORG_NAME).concat(" / ")	/使用rvs函数将上级单位按上级到下级排列

A6

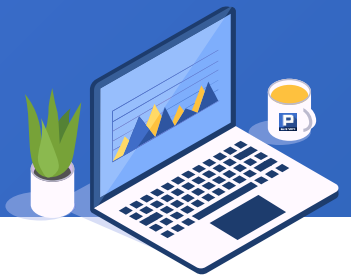
Member
Head Office / Beijing Branch Office / Beijing Marketing Department / Beijing Market Research Team

CONTENTS

02

基本运算

1. 两个序列间的运算：序列比较



以奥运会奖牌榜为例，查询哪几届奥运会中国奖牌榜排名比俄罗斯靠前。

Game	Nation	Medal
30	USA	46,29,29
30	China	38,27,23
30	UK	29,17,19
30	Russia	24,26,32
30	Korea	13,8,7
…	…	…

1. 两个序列间的运算：序列比较

SPL如下，其中用到了">"符号比较序列大小，会顺序比较对位成员：

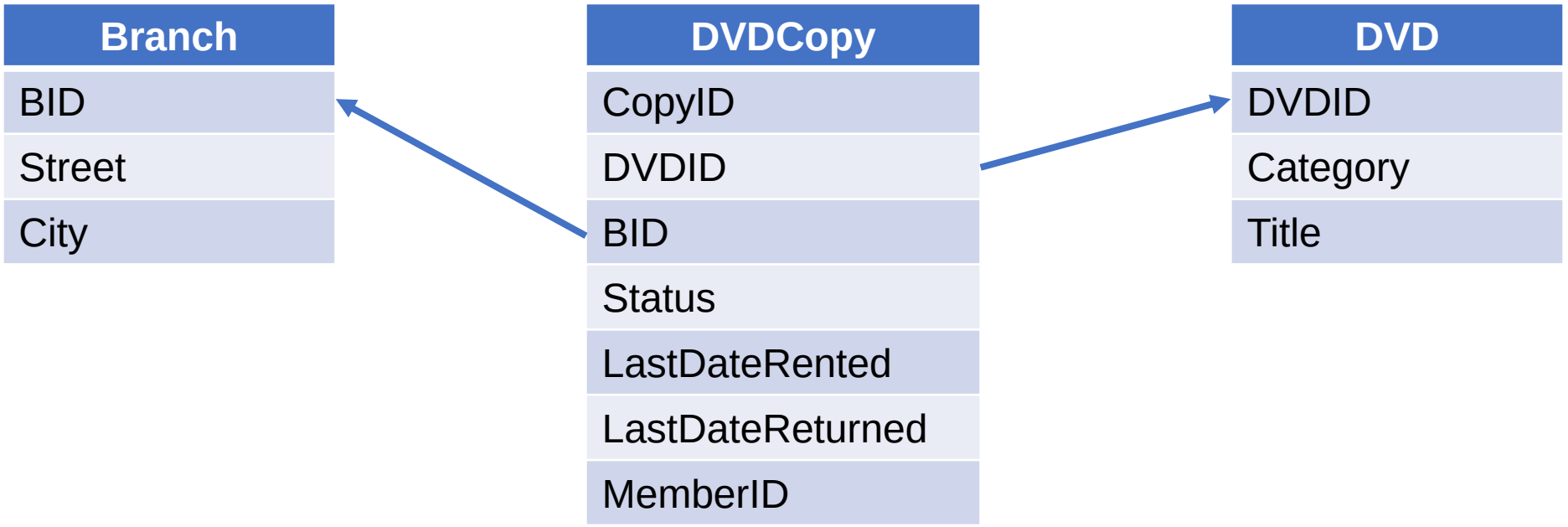
	A	B
1	=file("Olympic.csv").import@cqt()	/导入奥运会历届排名
2	=A1.run(Medal=Medal.split@c())	/奖牌字段按逗号拆分为序列
3	=A2.group(Game)	/按每届分组
4	=A3.select(~.select(Nation=="China").Medal>~.select(Nation=="Russia").Medal)	/用">"符号比较中俄的奖牌序列大小，会按顺序依次比较金牌、银牌和铜牌数量，并选出中国排名更高的届。
5	=A4.(Game)	/列出共有哪几届

A5	Game
	23
	25
	28
	29
	30

类似的，我们还可以使用"<"和"=="来比较序列大小。

1. 两个序列间的运算：和列、差列

查询出缺货的DVD分店，即现存的DVD拷贝不到4类的分店。其中Branch表，存储DVD分店信息；DVD表，存储DVD的标题及分类信息；DVDCopy表，存储DVD的多张拷贝，DVD拷贝是真正的光盘，以实体形式存放于各个分店。



1. 两个序列间的运算：和列、差列

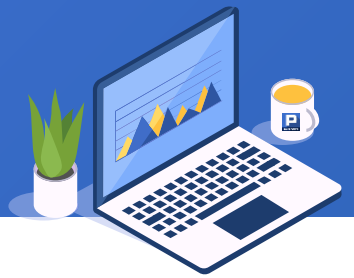
SPL如下，其中用到了“|”求和列，“\”求差列：

	A	B
1	=connect("db")	/连接数据源
2	=Branch=A1.query("select * from Branch")	/读取分店信息，并定义为Branch变量
3	=DVD=A1.query("select * from DVD")	/读取DVD信息，并定义为DVD变量
4	=DVDCopy=A1.query("select * from DVDCopy")	/读取DVDCopy信息，并定义为DVDCopy变量
5	=DVDCopy.switch(DVDID,DVD:DVDID; BID,Branch:BID)	/将DVDCopy的DVDID字段切换成DVD中对应的记录
6	=DVDCopy.select(STATUS!="Miss" && LASTDATERETURNED!=null)	/过滤丢失的和未归还的DVD拷贝
7	=A6.group(BID)	/对过滤后的数据按照BID分组
8	=A7.select(~.icount(DVDID.CATEGORY)<4)	/选出DVD拷贝小于4类的门店
9	=A8.(BID) | (Branch \ A7.(BID))	/缺货的门店。其中A8.(BID)表示DVD拷贝小于4类的门店，Branch \ A7.(BID)表示DVDCopy未出现过的门店。

A9

BID	STREET	CITY
B002	Street2	Houston
B003	Street3	LA
B004	Street4	Lincoln

1. 两个序列间的运算：并列、交列



中国的直辖市有：北京、天津、上海、重庆，一线城市有：上海、北京、深圳、广州。

求直辖市和一线城市共有哪些？哪些城市既是直辖市又是一线城市？

	A	B
1	[Beijing,Tianjin,Shanghai,Chongqing]	/直辖市
2	[Shanghai,Beijing,Shenzhen,Guangzhou]	/一线城市
3	=A1&A2	/求并列，即直辖市和一线城市的并集
4	=A1^A2	/求交列，即直辖市中的一线城市

A3	Member
	Beijing
	Shanghai

A4	Member
	Beijing
	Tianjin
	Shanghai
	Chongqing
	Shenzhen
	Guangzhou

1. 两个序列间的运算：四则运算

求2019年12月24日到26日深证300 (399007) 对深证成指 (399001) 的每日相对收益率。

Date	Code	Name	Open	Close	Amount
2020/2/18	399001	Shenzhen	11244.7651	11306.4863	3.19E+11
2020/2/17	399001	Shenzhen	10974.9328	11241.4993	3.12E+11
2020/2/14	399001	Shenzhen	10854.4551	10916.3117	2.77E+11
2020/2/13	399001	Shenzhen	10936.5011	10864.3222	2.87E+11
2020/2/12	399001	Shenzhen	10735.0475	10940.7952	2.66E+11
…	…	…	…	…	…

1. 两个序列间的运算：四则运算

使用了A ?? B对两序列成员按位进行"?"运算，其中?∈{+,-,*,/,%,\}。 SPL如下：

	A	B
1	=connect("db")	/连接数据源
2	=["399007","399001"].(A1.query("select * from StockIndex where code=? and date between '2019-12-23' and '2019-12-26'",~))	/读取深证300和深证成指在2019年12月23日到26日的数据，取23日是为了计算涨幅
4	=A2.(~.calc(to(2,4),Close/Close[-1]))	/分别计算24到26日每天的涨幅
5	=A3(1)--A3(2)	/两个序列按位相减，即是相对收益率

A5	Member
	0.0031349096521252617
	0.0011897141619391371
	-4.4910504685946595E-4

1. 两个序列间的运算：成员比较

下面是随机抽样后生成的文件，比较两次随机抽样是否选出了相同的序号。文件部分数据如下：

ID	Predicted_Y	Original_Y
10	0.012388464367608093	0.0
11	0.01519899123978988	0.0
13	0.0007920238885061248	0.0
19	0.0012656367468159102	0.0
21	0.009460545997473379	0.0
23	0.024176791871681664	0.0
…	…	…

1. 两个序列间的运算：成员比较

SPL如下，其中用到了cmp()函数对两个序列的成员按位比较：

	A	B
1	=file("p_old.csv").import@ct()	/读取第一次输出的文件
2	=file("p_new.csv").import@ct()	/读取第二次输出的文件
3	=cmp(A1.(ID),A2.(ID))	/比较两次生成的ID是否完全相同（成员值相等且顺序一致）

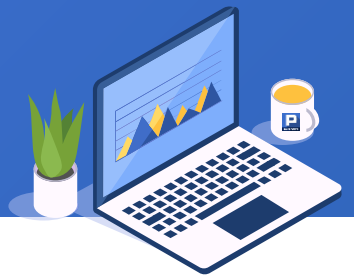
A3	Member
	0

结果为0表示两个文件ID完全一致。

如果ID的顺序可能不同，可以使用eq()函数比较两个序列的成员是否相同：

	A	B
3	=A1.(ID).eq(A2.(ID))	/比较两次生成的ID值是否相同，不要求顺序一致

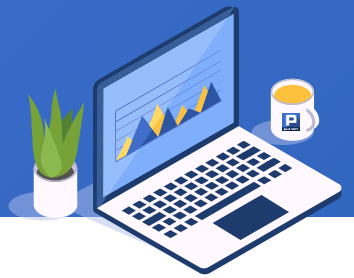
2. 序列与单值间的运算



同样的例子，两个文件的序号顺序相同，想比较有多少条数据完全一致。

ID	Predicted_Y	Original_Y
10	0.012388464367608093	0.0
11	0.01519899123978988	0.0
13	0.0007920238885061248	0.0
19	0.0012656367468159102	0.0
21	0.009460545997473379	0.0
23	0.024176791871681664	0.0
…	…	…

2. 序列与单值间的运算



SPL如下，其中用到了"|"将序列和单值合并到一起：

	A	B	C
1	=file("p_old.csv").import@ct()		/读取第一次输出的文件
2	=file("p_new.csv").import@ct()		/读取第二次输出的文件
3	for A1.len()	=cmp(A1(A3),A2(A3))	/循环比较两个文件的每条记录
4		=@ | B3	/把每次比较的结果与B4格值合并
5	=B4.count(~==0)		/统计有多少个比较结果相等的

B4	Member
	0
	0
	0
	…

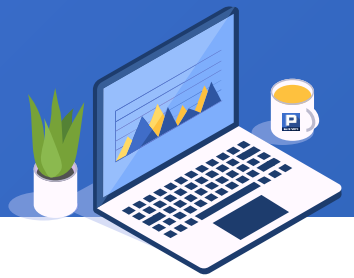
A5	Member
	11302

CONTENTS

03

聚合

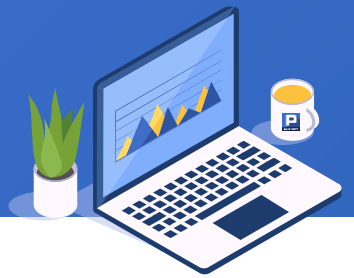
1. 单个序列的聚合：求和



下面是城市GDP表，分别统计直辖市、一线城市、二线城市的人均GDP。

ID	City	GDP	Population
1	Shanghai	32679	2418
2	Beijing	30320	2171
3	Shenzhen	24691	1253
4	Guangzhou	23000	1450
5	Chongqing	20363	3372
…	…	…	…

1. 单个序列的聚合：求和

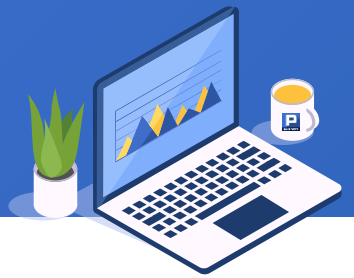


SPL如下，其中用到了sum()函数进行求和运算：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from GDP")	/查询城市GDP表
3	[["Beijing","Shanghai","Tianjing","Chongqing"].pos(?)>0,["Beijing","Shanghai","Guangzhou","Shenzhen"].pos(?)>0,["Chengdu","Hangzhou","Chongqing","Wuhan","Xian","Suzhou","Tianjing","Nanjing","Changsha","Zhengzhou","Dongguan","Qingdao","Shenyang","Ningbo","Kunming"].pos(?)>0]	/枚举直辖市、一线城市和二线城市
4	=A2.enum@r(A3,City)	/按城市枚举分组
5	=A4.new(A3(#):Area,~.sum(GDP)/~.sum(Population)*10000:CapitaGDP)	/统计每组的人均GDP

A5	Area	CapitaGDP
	["Beijing","Shanghai","Tianjing","Chongqing"].pos(?)>0	107345.03
	["Beijing","Shanghai","Guangzhou","Shenzhen"].pos(?)>0	151796.49
	["Chengdu","Hangzhou","Chongqing","Wuhan","Xian","Suzhou","Tianjing","Nanjing","Changsha","Zhengzhou","Dongguan","Qingdao","Shenyang","Ningbo","Kunming"].pos(?)>0	106040.57

1. 单个序列的聚合：最大值、最小值



以订单表为例，要把客户ANATR有重复时间段的订单合并。

OrderID	Customer	SellerId	OrderDate	FinishDate
10308	ANATR	7	2012/09/18	2012/10/16
10309	ANATR	3	2012/09/19	2012/10/17
10625	ANATR	3	2013/08/08	2013/09/05
10702	ANATR	1	2013/10/13	2013/11/24
10759	ANATR	3	2013/11/28	2013/12/26
…	…	…	…	…

1. 单个序列的聚合：最大值、最小值

SPL如下，其中用到了max()函数取最大值，min()函数取最小值：

	A	B
1	=connect("db")	/连接数据源
2	=A1.query("select * from Orders where Customer='ANATR' order by OrderDate")	/选出客户ANATR的订单信息，按订单日期排序
3	=A2.group@i(OrderDate>max(FinishDate[,-1]))	/把订购日期小于上一个完成日期的订单分到一组
4	=A3.new(Customer,~.min(OrderDate):OrderDate,~.max(FinishDate):FinishDate)	/将每组最早的订购日期作为订购日期，最晚的完成日期作为完成日期

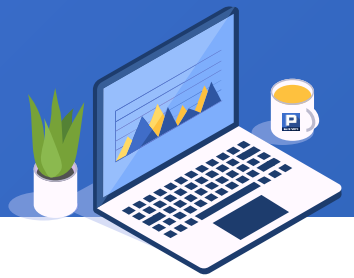
A3

Member
[[10308,ANATR,7,…], [10309,ANATR,3,…]]
[[10625,ANATR,3,…]]
[[10702,ANATR,1,…]]
[[10759,ANATR,3,…], [11079,ANATR,7,…]]
…

A4

Customer	OrderDate	FinishDate
ANATR	2012/09/18	2012/10/17
ANATR	2013/08/08	2013/09/05
ANATR	2013/10/13	2013/11/24
ANATR	2013/11/28	2013/12/29
…	…	…

1. 单个序列的聚合：平均



员工表部分数据如下：

ID	NAME	DEPT	STATE	SALARY
1	Rebecca	R&D	California	7000
2	Ashley	Finance	New York	11000
3	Rachel	Sales	New Mexico	9000
4	Emily	HR	Texas	7000
…	…	…	…	…

统计各部门在不同地区的平均工资，想要转换成如下结果：

DEPT	California	Florida	New York	Texas	…
Finance	8000	10000	7500	8166.67	…
HR	10000	7000	5000	6500	…
…	…	…	…	…	…

1. 单个序列的聚合：平均

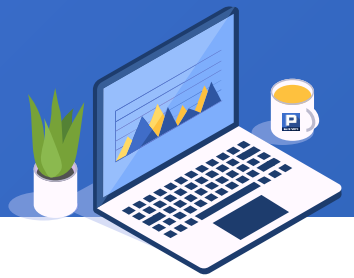
SPL如下，其中用到了avg() 函数计算平均值：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from EMPLOYEE")	/查询员工表
3	=A2.groups(DEPT,STATE;avg(SALARY):AvgSalary)	/分组汇总，使用avg函数计算各部门在各地的平均薪资
4	=A3.pivot(DEPT;STATE, AvgSalary)	/按照目标表格进行转置

A4

DEPT	California	Florida	New York	Texas	…
Finance	8000	10000	7500	8166.67	…
HR	10000	7000	5000	6500	…
…	…	…	…	…	…

1. 单个序列的聚合：计数



以成绩表为例，求一班各科不及格人数。

CLASS	STUDENTID	SUBJECT	SCORE
Class one	1	English	84
Class one	1	Math	77
Class one	1	PE	69
Class one	2	English	81
Class one	2	Math	80
…	…	…	…

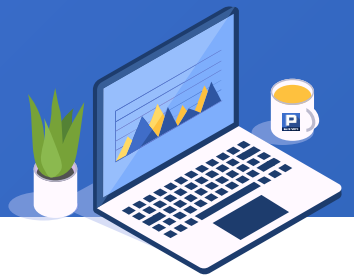
1. 单个序列的聚合：计数

SPL如下，其中用到了count() 函数计数：

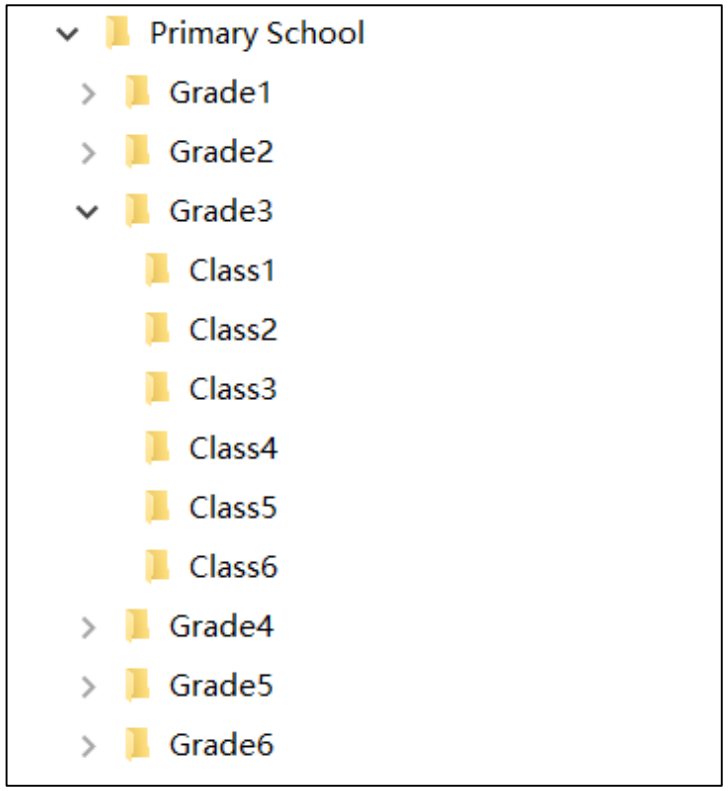
	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Scores where CLASS='Class one'")	/查询一班学生成绩
3	=A2.groups(SUBJECT; count(SCORE<60):FailCount)	/分组汇总，其中用到了count函数统计分数小于60的不及格人数

A3	SUBJECT	FailCount
	English	2
	Math	0
	PE	2

1. 单个序列的聚合：逻辑与运算

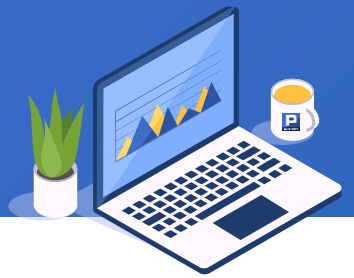


某小学调查学生在线教学可以使用的终端，查看是否所有学生都可以使用手机。各班问卷及汇总目录如下：



ID	STUDENT_NAME	TERMINAL
1	Rebecca Moore	Phone
2	Ashley Wilson	Phone,PC,Pad
3	Rachel Johnson	Phone,PC,Pad
4	Emily Smith	Phone,Pad
5	Ashley Smith	Phone,PC
6	Matthew Johnson	Phone
7	Alexis Smith	Phone,PC
8	Megan Wilson	Phone,PC,Pad
…	…	…

1. 单个序列的聚合：逻辑与运算



使用了A.ifn()函数取第一个非null成员，使用了A.cand()函数对成员进行与运算。

SPL如下：

	A	B	
1	=directory@ps("D:/Primary School")		/递归遍历目录，列出所有文件
2	for A1	=file(A2).xlsimport@t()	/循环导入各班级问卷excel文件
3		=B2.([TERMINAL,"Phone"].ifn().split@c().pos("Phone") > 0)	/当问卷中终端未填写时不认为不支持手机终端，使用ifn()函数保证此项为true。
4	=B3.cand()		/使用cand()函数计算B3的成员是否都是true

A1

Member
D:\Primary School\Grade1\Class1\Questionnaire.xlsx
D:\Primary School\Grade1\Class2\Questionnaire.xlsx
D:\Primary School\Grade1\Class3\Questionnaire.xlsx
…

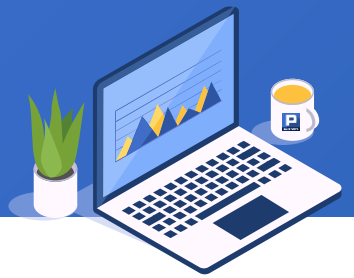
B3

Member
true
true
true
…

A4

Member
false

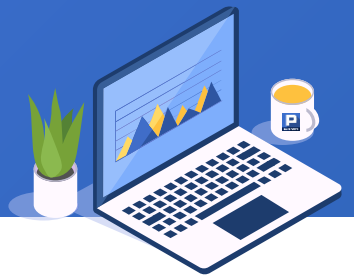
1. 单个序列的聚合：逻辑或运算



销售表部分数据如下，查询客户RATTC，在2014年是否排进过某月销售额前三名。

OrderID	Customer	SellerId	OrderDate	Amount
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
10404	MAGAA	2	2014/01/03	1675.0
…	…	…	…	…

1. 单个序列的聚合：逻辑或运算



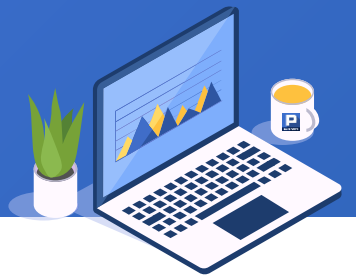
SPL如下，其中用到了A.cor()函数对成员进行或运算：

	A	B
1	=connect("db").query("select * from sales")	/连接数据源，读取销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年数据
3	=A2.group(month(OrderDate))	/使用group函数，将2014年的数据按照月份分组
4	=A3.(~.groups(Customer; sum(Amount):Amount))	/分组后的成员按照客户分组汇总销售额
5	=A4.new(Customer, ~.top(-3;sum(Amount)):Top3)	/循环每个月的数据，计算每月销售额前3的客户
6	=A5.(Top3.(Customer).pos("RATTC")>0)	/判断每个月前三名是否包含客户RATTC
7	=A6.cor()	/使用cor()函数计算A6的成员是否存在true

A6	Member
	false
	false
	true
	…

A7	Member
	true

1. 单个序列的聚合：非重复计数



分析数据文件中哪个字段是序号，部分数据如下：

PassengerId	Survived	Pclass	Name	Sex	Age
1	0	3	"Braund, Mr. Owen Harris"	male	22
2	1	1	"Cumings, Mrs. John Bradley"	female	38
3	1	3	"Heikkinen, Miss. Laina"	female	26
4	1	1	"Futrelle, Mrs. Jacques Heath"	female	35
5	0	3	"Allen, Mr. William Henry"	male	35
6	0	3	"Moran, Mr. James"	male	
7	0	1	"McCarthy, Mr. Timothy J"	male	54
…	…	…	…	…	…

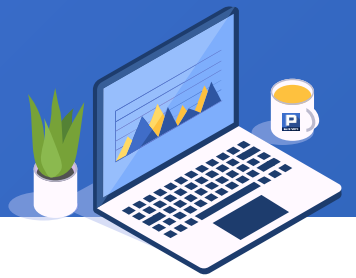
1. 单个序列的聚合：非重复计数

SPL如下，其中用到了icount() 函数非重复计数：

	A	B	C	D
1	=file("titanic_train.csv").import@cqt()			/读取数据文件
2	=A1.fno().new(A1.fname(~):Name,A1.field(~).icount():DCount)			/使用icount()函数计算每个字段的非重复成员的数量
3	=A2.select(DCount==A1.len())			/选出非重复计数与全部数据长度相同的字段
4	if (A3.len() > 1)	=A3.select(like@c(Name,"*id*"))		/如果满足条件的字段有多个，选出字段名包含id的。
5		if (B4.len() > 0)	>A3=B4	/如果有字段名包含id的，将选出字段赋值给A3
6	=A3.minp(len(Name)).Name			/从选出字段中，选出名称最短的字段名

A6	Value
	PassengerId

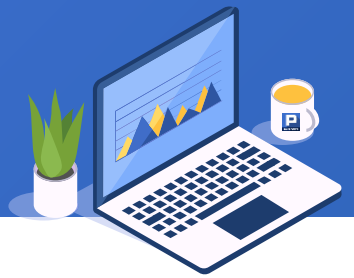
1. 单个序列的聚合：中位数



员工表部分数据如下，要统计每个部门工资的中位数。

ID	NAME	DEPT	SALARY
1	Rebecca	R&D	7000
2	Ashley	Finance	11000
3	Rachel	Sales	9000
4	Emily	HR	7000
5	Ashley	R&D	16000
…	…	…	…

1. 单个序列的聚合：中位数

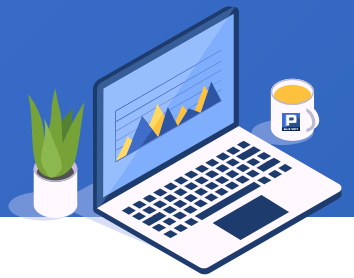


SPL如下，其中用到了median()函数求中位数：

	A	B
1	=connect("db").query("select * from employee")	/连接数据源，读取员工表
2	=A1.groups(DEPT;median(;SALARY))	/按部门分组汇总，计算每个部门的工资中位数

A3	DEPT	MedianSalary
	Administration	9500.0
	Finance	7000.0
	HR	7000
	Marketing	7000
	Production	6500
	…	…

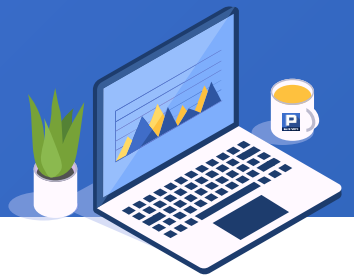
1. 单个序列的聚合：排名



以成绩表为例，求一班学号为8的学生的总成绩在班级的排名。

CLASS	STUDENTID	SUBJECT	SCORE
Class one	1	English	84
Class one	1	Math	77
Class one	1	PE	69
Class one	2	English	81
Class one	2	Math	80
…	…	…	…

1. 单个序列的聚合：排名

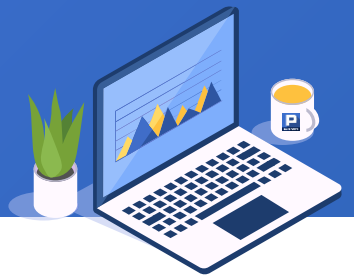


SPL如下，其中用到了A.rank()函数计算排名：

	A	B
1	=connect("db").query("select * from SCORES where CLASS='Class one'")	/连接数据源，读取一班成绩
2	=A1.groups(STUDENTID;sum(SCORE):TotalScore)	/分组汇总每位学生的总成绩
3	=A2.select(STUDENTID==8).TotalScore	/取出学号为8的学生的总成绩
4	=A2.rank@z(A3:TotalScore)	/使用A.rank()函数计算排名，用到@z选项使总分从大到小排名

A4	Value
	13

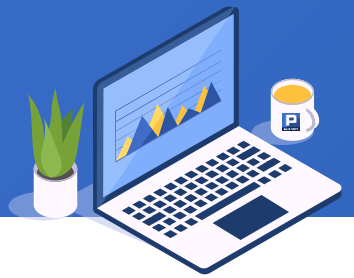
1. 单个序列的聚合：排名



智能建模后测试数据的预测结果如下，计算模型的AUC指标：

ID	Predicted_Y	Original_Y
10	0.012388464367608093	0.0
11	0.01519899123978988	0.0
13	0.0007920238885061248	0.0
19	0.0012656367468159102	0.0
21	0.00946054599747337 9	0.0
…	…	…

1. 单个序列的聚合：排名

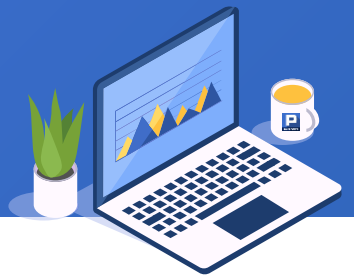


SPL如下，其中用到了ranks() 函数计算排名：

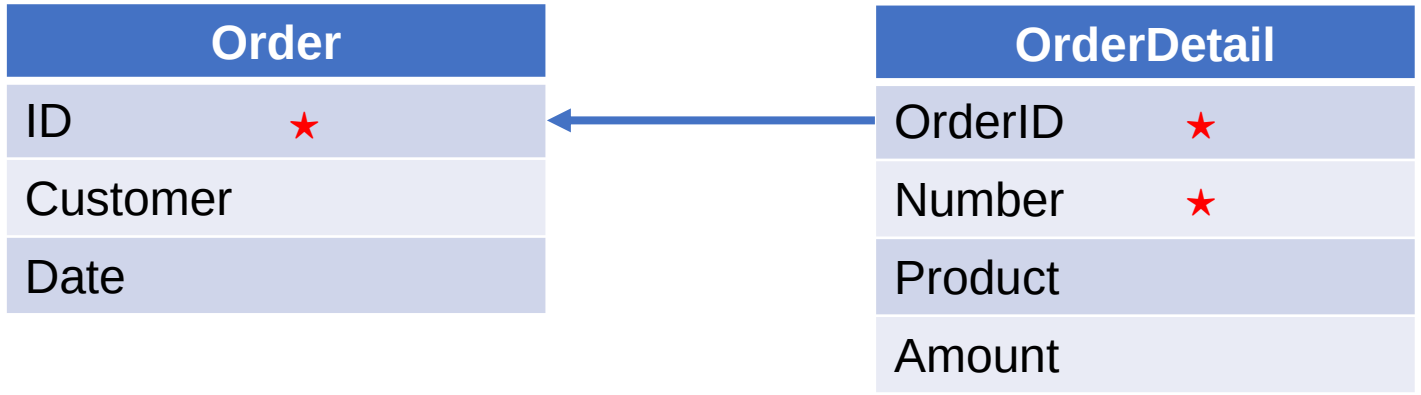
	A	B
1	=file("p.csv").import@ct()	/读取文件
2	=P=A1.pselect@a(Original_Y==1),M=P.len()	/选出所有真实目标是1的记录行号设为P，其数量设为M
3	=N=A1.len()-M	/真实目标不是1的数量设为N
4	=A1.(Predicted_Y).ranks@s()	/使用ranks()函数计算预测值的排名。这里使用了@s选项，值相等时会使用排名的均值
5	=(A4(P).sum()-M*(1+M)/2)/(M*N)	/根据公式计算AUC值

A5	Member
	0.9055583178459794

2. 序列的序列的聚合：和列



订单表和订单明细表是主子表关系，每个订单有多条明细数据。如下图：



订单明细表中每个订单的明细数据是不定长的。想要查询出如下表格：

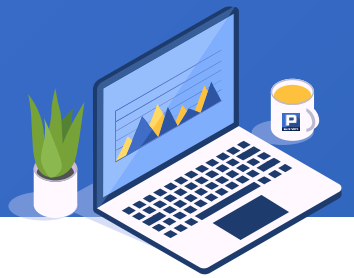
ID	Customer	Date	Product1	Amount1	Product2	Amount2	Product3	Amount3
1	3	20190101	Apple	5	Milk	3	Salt	1
2	5	20190102	Beef	2	Pork	4		
3	2	20190102	Pizza	3				

2. 序列的序列的聚合：和列

SPL如下，其中用到了A.conj()函数合并序列成员：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from OrderDetail left join Order on Order.ID=OrderDetail.OrderID")	/导入订单明细表和订单表，并按订单ID连接订单表
3	=A2.group(ID)	/将取出的数据按订单ID分组
4	=A3.max(~.count()).("Product"+string(~)+","+"Amount"+string(~)).concat@c()	/找到分组后成员最多的一组确定目标表格数据结构
5	=create(ID,Customer,Date,${A4})	/根据A4确定的数据结构创建序表
6	>A3.run(A5.record([ID,Customer,Date]|~.([Product,Amount]).conj()))	/循环分组数据，每个分组内将成员拼到一个序列，这里用到了conj函数取每组各个产品和数量的和列。最后把生成的记录插入到A5创建的序表中。

2. 序列的序列的聚合：和列



下面是某时刻，新冠状病毒世界各地确诊人数的JSON数据，要统计世界确诊人数。

```
[
{Region:"China",Confirmed:[
        {Region:"Hubei",Confirmed:[
                {Region:"Wuhan",Confirmed:51986},
                {Region:"Xiaogan",Confirmed:3009},
                {Region:"Huanggang",Confirmed:3791},
                ...]
        },
        {Region:"Taiwan",Confirmed:18},
        ...]
},
{Region:"Thailand",Confirmed:33},
...]
```

2. 序列的序列的聚合：和列

SPL如下，其中用到了A.conj@r()函数来递归合并序列成员：

	A	B
1	=json(file("COVID-19.json").read())	/导入JSON数据文件
2	=A1.field@r("Confirmed")	/使用A.field()函数的@r选项递归获取所有确诊字段
3	=A2.conj@r()	/使用A.conj()函数的@r选项递归合并
4	=A3.sum()	/确诊人数求和

A2

Member
[[51986,3009,3791],[58,…]
251
33
28
19
16
…

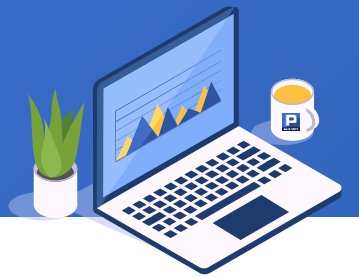
A3

Member
51986
3009
3791
1447
1206
1125
…

A4

Member
64438

2. 序列的序列的聚合：并列和差列



有课程表和选课表，查询有哪些课没有学生选修。其中选课表可以多选，用逗号分隔，部分数据如下：

Course		
ID	NAME	TEACHERID
1	Environmental protection and …	5
2	Mental health of College Students	1
3	Computer language Matlab	8
4	Electromechanical basic practice	7
5	Introduction to modern life science	3
6	Modern wireless communication system	14
…	…	…

SelectCourse		
ID	STUDENTID	COURSE
1	59	2,7
2	43	1,8
3	52	2,7,10
4	44	1,10
5	37	5,6
6	57	3
…	…	…

2. 序列的序列的聚合：并列和差列

使用了A.union()函数求序列的序列成员的交列，使用了A.diff()函数求序列的序列成员的差列， SPL如下：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Course")	/读取课程表
3	=A1.query("select * from SelectCourse")	/读取学生选课表
4	=A3.union(COURSE.split@cp())	/将选课表中的课程按逗号拆分后，使用union()函数将课程序列求交列
5	=A2.(ID)	/所有课程的序号
6	=A2(A5.pos([A5,A4].diff()))	/使用diff()函数求课程表和选课表的课程序号的差列，即没有学生选择的课程，在A5中定位后，从A2中选出。

A6

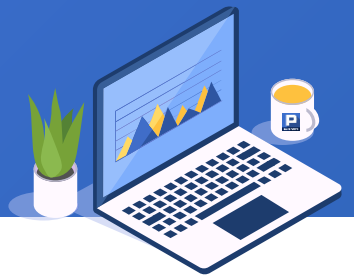
ID	NAME	TEACHERID
1	Fundamentals of economic management	21

2. 序列的序列的聚合：交列

销售表部分数据如下，统计出2014年每个月销售金额均排在前20名的客户名称。

OrderID	Customer	SellerId	OrderDate	Amount
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
10404	MAGAA	2	2014/01/03	1675.0
…	…	…	…	…

2. 序列的序列的聚合：交列

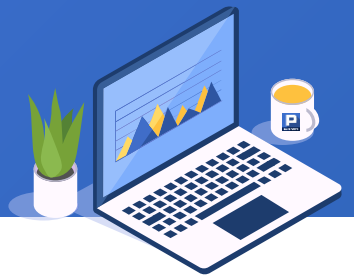


SPL如下，其中用到了isect()函数求成员交集：

	A	B
1	=connect("db").query("select * from sales")	/连接数据源，读取销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年数据
3	=A2.group(month(OrderDate))	/使用group函数，将2014年的数据按照月份分组
4	=A3.(~.group(Customer))	/分组后的成员按照客户分组
5	=A4.(~.top(-20;sum(Amount)))	/循环每个月的数据，计算每月销售额前20的客户
6	=A5.(~.(Customer))	/列出了销售额前20名客户名称
7	=A6.isect()	/使用isect()函数求每组之间的交集

A7	Member
	HANAR
	SAVEA

2. 序列的序列的聚合：异或列



成绩表按学期保存在不同的文件中，要查询上下学期只有一次进入总分前十名的学生ID。

CLASS	STUDENTID	SUBJECT	SCORE
Class one	1	English	84
Class one	1	Math	77
Class one	1	PE	69
Class one	2	English	81
Class one	2	Math	80
…	…	…	…

2. 序列的序列的聚合：异或列

SPL如下，其中用到了A.xunion() 函数将序列中的序列成员之间不重复的记录选出：

	A	B
1	=file("Scores1.csv").import@ct()	/导入学生上学期成绩
2	=file("Scores2.csv").import@ct()	/导入学生下学期成绩
3	=A1.groups(STUDENTID;sum(SCORE):Score)	/分组汇总上学期学生总成绩
4	=A2.groups(STUDENTID;sum(SCORE):Score)	/分组汇总下学期学生总成绩
5	=A3.top(-10;Score).(STUDENTID)	/选出上学期总分前十名的学生ID
6	=A4.top(-10;Score).(STUDENTID)	/选出下学期总分前十名的学生ID
7	=[A5,A6].xunion()	/使用xunion函数，选出上下学期的学生ID不重复的记录。

A5	Member
	2
	9
	4
	10
	…

A6	Member
	12
	1
	8
	4
	…

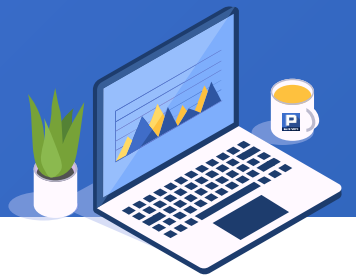
A7	Member
	2
	9
	10
	7
	…

CONTENTS

04

循环计算

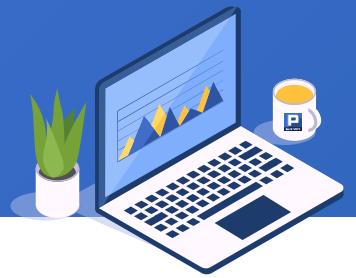
1. 循环函数



下面是按年月分组的JSON格式销售数据，求2016年的总销售额。

```
[
{YEAR:2016,MONTH:1,SALES:[
                {ORDERNUMBER:10101, ORDERLINENUMBER:4, SALES:3782, ...},
                {ORDERNUMBER:10102, ORDERLINENUMBER:1, SALES:3773.38, ...},
               ...]
},
{YEAR:2016,MONTH:2,SALES:[
               {ORDERNUMBER:10105, ORDERLINENUMBER:2, SALES:7208 ...},
               {ORDERNUMBER:10106, ORDERLINENUMBER:15, SALES:8690.36, ...},
               ...]
},
...]
```

1. 循环函数



SPL如下，其中用到了A.() 进行循环计算：

	A	B
1	=json(file("sales.json").read())	/导入JSON数据文件
2	=A1.select(YEAR=2016)	/选出2016年的销售数据
3	=A2.field@r("SALES")	/递归获取销售额字段值
4	=A3.(~.sum()).sum()	/使用A.()循环计算每组的销售额总和，再求总销售额

A3

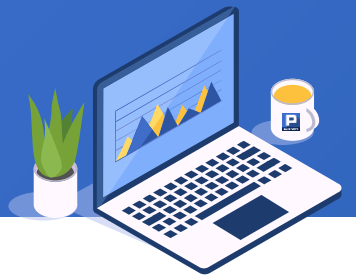
Member
[3782,3773.38,1404, …]
[7208,8690.36,4566.05,…]
[5265.15,6130.35,3485.82…]
[2793.86,9264.86,2082.49,…]

Member
3782
3773.38
1404
…

A4

Member
1252700.43

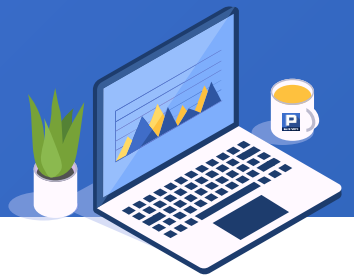
1. 循环函数



销售表部分数据如下，对2014年业绩在前 10% 销售员再给予 5% 的业绩奖励。

OrderID	Customer	SellerId	OrderDate	Amount
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
10404	MAGAA	2	2014/01/03	1675.0
…	…	…	…	…

1. 循环函数

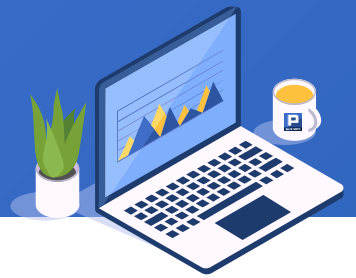


SPL如下，其中用到了A.run()函数对序列成员进行循环计算：

	A	B
1	=connect("db").query("select * from sales")	/连接数据源，读取销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年数据
3	=A2.groups(SellerId;sum(Amount):Amount)	/使用groups函数，按销售员分组汇总当年销售总额
4	=A3.sort@z(Amount).to(A3.len()*0.1)	/按销售额降序排列，取前百分之十
5	=A4.run(Amount*=1.05)	/对前百分之十循环计算销售额，给予5%的奖励并赋值

A5	SellerId	Amount
	4	150433.185
	3	127878.04
	1	102756.759
	8	87965.346

1. 循环函数



支付表部分数据如下：

ID	customID	name	amount_payable	due_date	amount_paid	pay_date
112101	C013	CA	12800	2014-02-21	12800	2014-12-19
112102	C013	CA	3500	2014-06-15	3500	2014-12-15
112103	C013	CA	2600	2015-03-21	6900	2015-10-17

要求根据指定年份（如2014），输出每月应付金额，若无当月数据，则当月应付金额为上月该值：

name	1	2	3	4	5	6	7	8	9	10	11	12
CA		12800	12800	12800	12800	3500	3500	3500	3500	3500	3500	3500
…												

1. 循环函数

SPL如下，在循环中对当前成员再进行循环计算：

	A	B
1	$select * from Payment.txt where year(due_date)=2014	/导入支付表2014年的数据
2	=create(name,${12.string()})	/按目标结构创建序表
3	=A1.group(customID).((m12=12.(null),~.(m12(month(due_date))=amount_payable), m12.(~=ifn(~,~[-1])),A2.record(name|m12)))	/按客户ID分组，循环每组数据，对成员再进行循环，计算每个月的应付金额，最后和客户名称一起添加到A2创建的序表中

A3

name	1	2	3	4	5	6	7	8	9	10	11	12
CA		12800	12800	12800	12800	3500	3500	3500	3500	3500	3500	3500
NK			5800	5800	5800	9600	9600	9600	9600	3100	3100	3100
…												

2. 符号

求上证指数2019年收盘价连续增长的最大天数。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

2. 符号

SPL如下，其中用到了~代表循环中的当前成员：

	A	B
1	=file("000001.csv").import@ct()	/导入数据文件
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年的记录并按日期排序
3	=last=0,count=0,A2.(Close).((if(~>last,count+=1,count=0),last=~,count)).max()	/循环收盘价，比较每天的收盘价和前日收盘价，如果当日收盘价更高，则计数加1，最后选出计数最大值。

A3	Value
	6

2. 符号

百鸡问题，鸡翁一值钱五，鸡母一值钱三，鸡雏三值钱一。百钱买百鸡，问鸡翁、母、雏各几？SPL如下，其中用到了~符号代表当前成员：

	A	B
1	=to(100/5)	/可能购买的鸡翁数量
2	=to(100/3)	/可能购买的鸡母数量
3	=33.(~*3)	/可能购买的鸡雏数量
4	=create(Cock,Hen,Chick)	/创建序表用于存放三种鸡的数量
5	>A1.run(A2.run(A3.run(if(A1.~+A2.~+A3.~==100 && A1.~*5+A2.~*3+A3.~/3==100,A4.insert(0,A1.~,A2.~,A3.~)))))	/分别循环鸡翁、鸡母、鸡雏，当满足百钱买百鸡时，将结果插入到A4创建的序表中。其中用到了~符号代表序列循环的当前成员

A5	Cock	Hen	Chick
	4	18	78
	8	11	81
	12	4	84

2. 符号

下面是两个文本，在文本2中查找文本1的字符串，希望按下面形式返回:

file1	file2	输出
like parks	I like to go out because I like parks.	Q1. like parks
went out	Ben does not go out much.	I
go out	Shelly went out often but does not like parks.	Shelly
	Harry does not go out neither does he like parks.	Harry
		Q2. went out
		Shelly
		Q3. go out
		I
		Ben
		Harry

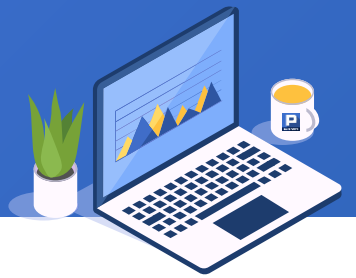
2. 符号

SPL如下，在循环中用到了#符号代表当前序号：

	A	B
1	=file("file1.txt").read@n()	/读取文本1
2	=file("file2.txt").read@n()	/读取文本2
3	=A1.conj(("Q"+string(#)+". "+~)|A2.select(pos(~,A1.~)).(~.words()(1)))	/循环文本1中的字符串，在文本2中查找，并取第一个英文单词。A2.select中用到了~代表A2当前成员，A1.~代表A1的当前成员。每组搜索结果前拼上Q+A1的序号+A1当前成员，其中序号通过#取得。

A3	Member
	Q1. like parks
	I
	Shelly
	Harry
	Q2. went out
	Shelly
	Q3. go out
	I
	Ben
	Harry

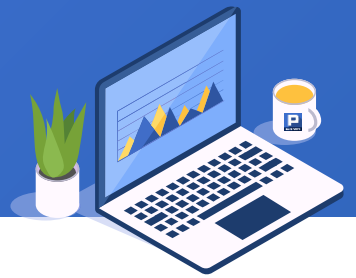
3. 定位计算



列出招商银行(600036)2020年1月1日到10日每天的20日收盘均价。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

3. 定位计算



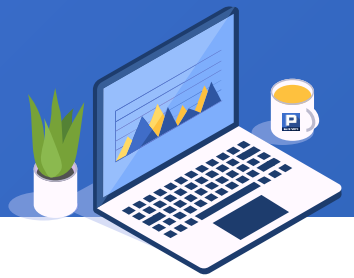
SPL如下，其中使用了A.calc()函数在指定位置上计算并返回，使用了 [a:b]循环访问成员：

	A	B
1	=connect("db")	/连接数据源
2	=A1.query("select Date, Close from Stock where Code='600036' order by Date")	/选出招商银行数据并按日期排序
3	=A2.pselect@a(Date>=date("2020/01/01") && Date<=date("2020/01/10"))	/使用pselect()函数获取2020年1月1日到10日的记录所在的行号
4	=A2(A3).derive(A2.calc(A3(#),avg(Close[-19:0])):ma20)	/使用calc()函数循环计算前十日数据的20日均值并返回，计算时使用了Close[-19:0] 代表从今天到过去19天的收盘价。

A3	Member
	4311
	4312
	4313
	4314
	4315
	4316
	4317

A4	Date	Close	ma20
	2020/01/02	38.88	37.35
	2020/01/03	39.4	37.50
	2020/01/06	39.24	37.64
	2020/01/07	39.15	37.79
	2020/01/08	38.41	37.90
	2020/01/09	38.9	38.03
	2020/01/10	39.04	38.16

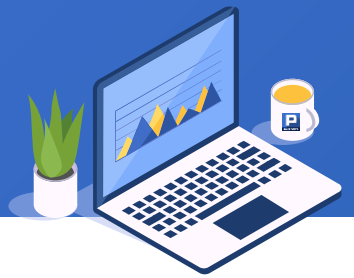
4. 迭代计算



销售表部分数据如下，统计出2014年每个月达到20笔订单需要多少天。

OrderID	Customer	SellerId	OrderDate	Amount
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
10404	MAGAA	2	2014/01/03	1675.0
…	…	…	…	…

4. 迭代计算

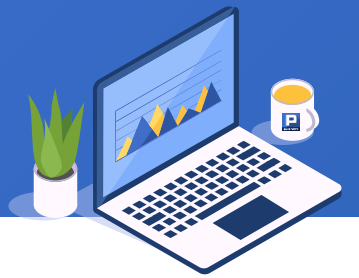


SPL如下，其中用到了seq()函数生成序号：

	A	B
1	=connect("db").query("select * from sales")	/连接数据源，读取销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年数据
3	=A2.sort(OrderDate)	/按照订单日期排序
4	=A3.select(seq(month(OrderDate))==20)	/使用seq()函数，生成每个月份的订单序号，并选出每个月序号为20的记录
5	=A4.new(month(OrderDate):Month,day(OrderDate):Day)	/列出了每月达到20笔销售需要的天数

A5	Month	Day
	1	20
	2	20
	3	20
	4	18
	…	…

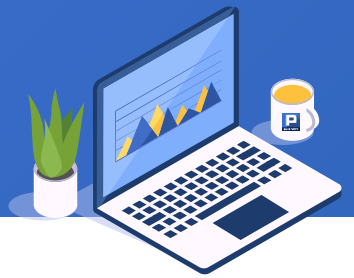
4. 迭代计算



求员工收入按部门的排名。员工表部分数据如下：

ID	NAME	DEPT	SALARY
1	Rebecca	R&D	7000
2	Ashley	Finance	11000
3	Rachel	Sales	9000
4	Emily	HR	7000
5	Ashley	R&D	16000
…	…	…	…

4. 迭代计算

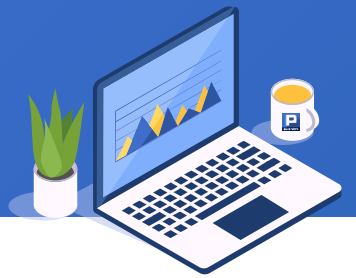


SPL如下，其中用到了rank()函数对有相同字段值的成员进行统一编号：

	A	B
1	=connect("db") .query("select * from Employee order by DEPT, SALARY DESC")	/连接数据源，读取员工表并按部门和收入排序
2	=A1.derive(rank(SALARY;DEPT):DeptRank)	/对有序的部门和收入利用rank()函数编号，计算出各部门排名

A2	ID	NAME	DEPT	SALARY	DeptRank
	2	Ashley	Finance	11000	1
	32	Andrew	Finance	11000	1
	230	Hannah	Finance	10000	3
	24	Chloe	Finance	10000	3
	…	…	…	…	…

4. 迭代计算



以成绩表为例，求一班学生ID为8的学生各科成绩的排名。

CLASS	STUDENTID	SUBJECT	SCORE
Class one	1	English	84
Class one	1	Math	77
Class one	1	PE	69
Class one	2	English	81
Class one	2	Math	80
…	…	…	…

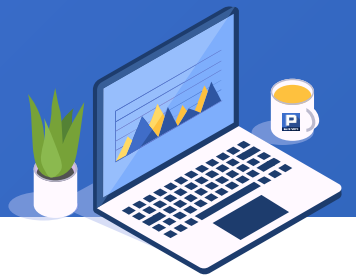
4. 迭代计算

使用了ranki()函数对有相同字段值的成员进行统一编号。ranki与rank函数的区别是，返回的编号是连续的，类似DENSE_RANK。SPL如下：

	A	B
1	=connect("db") .query("select * from SCORES where CLASS='Class one' order by SUBJECT, SCORE DESC")	/连接数据源，读取学生成绩表并按学科和成绩排序
2	=A1.derive(ranki(SCORE;SUBJECT):Rank)	/对有序的学科和成绩利用ranki()函数编号，计算出各学科排名
3	=A2.select(STUDENTID==8)	/选出学生ID是8的学生信息
4	=create(${A3.(SUBJECT).concat@c()}).record(A3.(Rank))	/利用A3选出的结果，整理出各学科排名

A4	English	Math	PE
	10	4	14

4. 迭代计算



求上证指数2019年每个交易日当天的全年累计成交金额。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

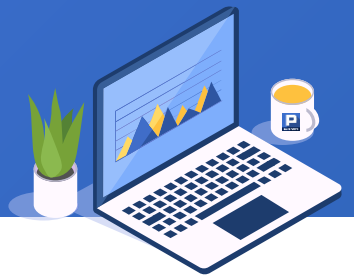
4. 迭代计算

SPL如下，其中用到了cum()函数计算累计成交金额：

	A	B
1	=file("000001.csv").import@ct()	/导入数据文件
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年的记录并按日期排序
3	=A2.derive(cum('Amount'):CUM)	/使用cum()函数计算累计成交金额

A3	Date	Open	Close	Amount	CUM
	2019/01/02	2497.8805	2465.291	9.759E10	9.759E10
	2019/01/03	2461.7829	2464.3628	1.07E11	2.046E11
	2019/01/04	2446.0193	2514.8682	1.39E11	3.436E11
	2019/01/07	2528.6987	2533.0887	1.46E11	4.896E11
	2019/01/08	2530.3001	2526.4622	1.23E11	6.126E11
	…	…	…	…	…

4. 迭代计算

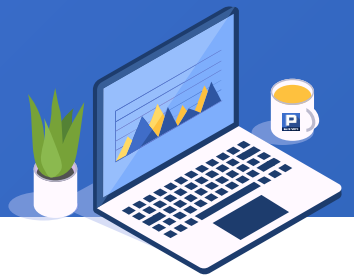


2014年初制定了第一季度销售目标为15万，求哪一天完成的销售目标?

销售表部分数据如下:

OrderID	Customer	SellerId	OrderDate	Amount
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
10404	MAGAA	2	2014/01/03	1675.0
…	…	…	…	…

4. 迭代计算



SPL如下，其中用到了A.iterate()函数对序列成员进行迭代计算：

	A	B
1	=connect("db").query("select * from sales")	/连接数据源，读取销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年数据
3	=A2.iterate((@+=Amount, ~~=OrderDate),0,@>150000)	/使用iterate()函数迭代计算，初始值为0。将销售额累加到当前格，直到超过15万终止。函数返回订单日期。

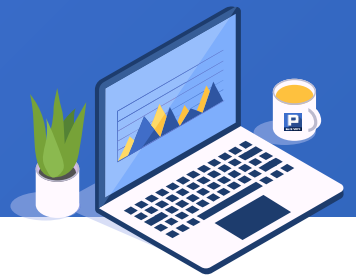
A3	Member
	2014/3/25

CONTENTS

05

定位

1. 定位成员在序列中的位置



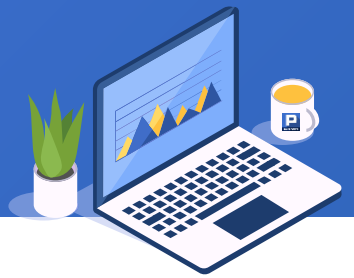
下面任课教师表中，第一列是教师姓名，第二列是学科，后面是课程代码（null是空）。

Teachers.txt														
Petitti	Matematica	mif	mig	vif	vig	null	null	null	null	null	null	null	null	…
Canales	Apesca	luc	lud	mac	mad	mic	mid	juc	jud	null	null	null	null	…
Lucero	NavegacionI	lub	luc	lud	lue	mab	mac	mad	mae	mib	mic	mid	mie	…
Bergamaschi	TecPesc	lua	luf	maa	maf	mia	mif	jua	juf	via	vif	null	null	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

根据任课教师表和右侧课程表(Courses.txt)列出每个课程适合的老师：

Monday	Tuesday	Wednesday	Thursday	Friday
lua	maa	mia	jua	via
lub	mab	mib	jub	vib
luc	mac	mic	juc	vic
lud	mad	mid	jud	vid
lue	mae	mie	jue	vie
luf	maf	mif	juf	vif
lug	mag	mig	jug	vig

1. 定位成员在序列中的位置

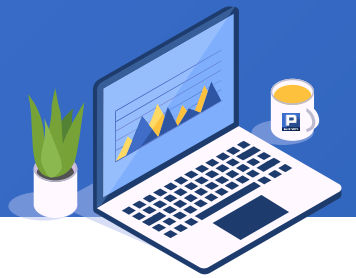


SPL如下，其中用到了pos函数获取成员位置序号：

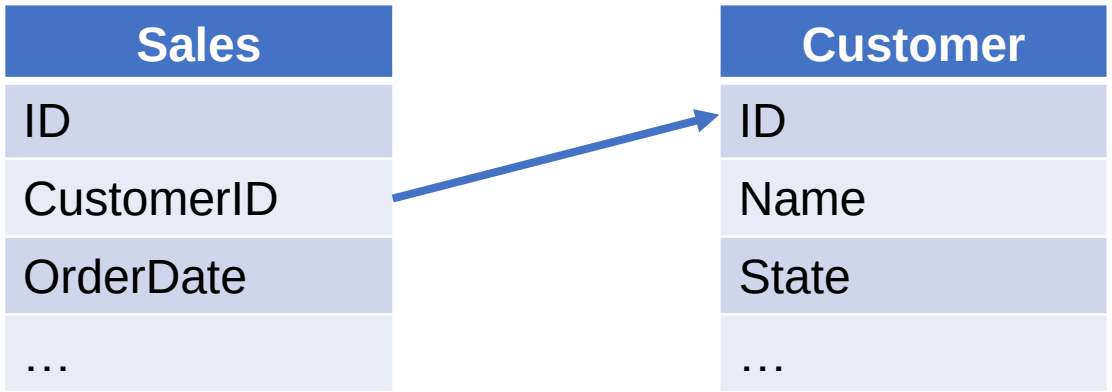
	A	B
1	=file("Teachers.txt").import()	/导入数据文件
2	=A1.new(#1:professor,~.array().to(3,A1.fno()).select(~!=null):codeArray)	/产生两个列的序表，其中第一列是教师名，第二列是课程列表。
3	=file("Courses.txt").import@t().conj(~.array())	/根据课程表列出所有课程，合并到一个序列
4	=A3.(A2.select(codeArray.pos(A3.~)).(professor))	/循环课程表，使用pos函数在教师的课程列表中查找该课程，选出每节课适合的教师。
5	=create(Monday,Tuesday,Wednesday,Thursday,Friday).record(A4.(~.concat@c()))	/创建周一到周五的课程表，并依此填入教师。

A5	Monday	Tuesday	Wednesday	Thursday	Friday
	Bergamaschi,Puebla	Bergamaschi,Pue…	Bergamaschi,Puebla	Bergamaschi,Pue…	Bergamaschi,Puebla
	Lucero,Puebla,Lu…	Lucero,Mazza,Pu…	Lucero,Puebla,Chi…	Lucero,Mazza,Pe…	Lucero,Puebla,Vel…
	Canales,Lucero,P…	Canales,Lucero,M…	Canales,Lucero,P…	Canales,Lucero,M…	Lucero,Velasco,Lu…
	…	…	…	…	…

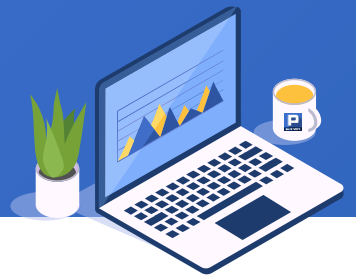
1. 定位成员在序列中的位置



有销售表和客户表，查询2014年没有销售记录的客户。



1. 定位成员在序列中的位置

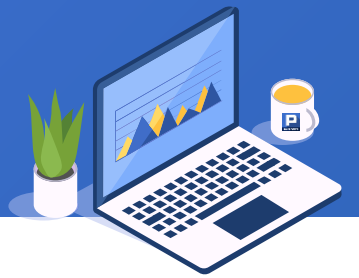


SPL如下，其中用到了A.pos()函数定位成员在序列中的位置：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Sales where year(OrderDate)=2014")	/查询2014年的销售记录
3	=A1.query("select * from Customer")	/查询客户表
4	=A3.(ID).sort()	/从客户表中选出客户序号
5	=A2.align(A4.len(), A4.pos@b(CustomerID))	/销售表按照客户序号对位分组，使用pos函数定位客户序号。由于客户序号是有序的，这里使用@b选项进行二分法查找，可以更快速定位。
6	=A3(A5.pos@a(null))	/使用pos()函数的@a选项，选出所有没有销售记录（值为null）的客户信息，否则只会选出一个。

A6	ID	Name	State	…
	ALFKI	CMA-CGM	Texas	…
	CENTC	Nedlloyd	Florida	…

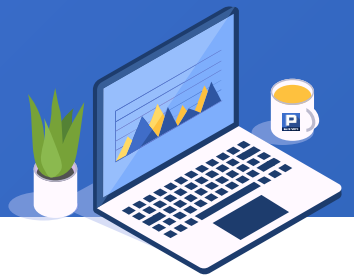
2. 取最大值/最小值对应记录的行号



求上证指数收盘价最高点那天，相对于前一日的涨幅。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

2. 取最大值/最小值对应记录的行号



我们需要知道股市最高点的记录所在行号，再与上一个交易日比较得出结果。

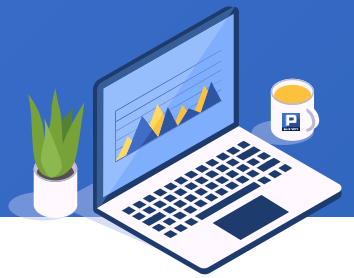
SPL如下，其中用到了pmax函数来取最大值所在行号：

	A	B
1	=file("000001.csv").import@ct()	/导入数据文件
2	=A1.sort(Date)	/按日期排序
3	=A2.pmax(Close)	/取出股市最高点所在行号
4	=A2(A3).Close/A2.m(A3-1).Close-1	/使用当天收盘价和前日收盘价计算涨幅

同样可以使用pmin函数来取最小值所在行号：

	A	B
3	=A3.pmin(Close)	/取出股市最低点所在行号

2. 取最大值/最小值对应记录的行号



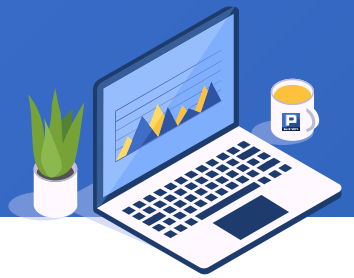
最大值所在记录不一定是唯一的，如果想返回所有的行号，可以使用pmax@a选项：

	A	B
3	=A2.pmax@a(Close)	/取出所有股市最高点记录所在行号
4	=A3.(A2(A3.~).Close/A2.m(A3.~-1).Close-1)	/循环使用当天收盘价和前日收盘价计算涨幅

如果希望从后向前定位，可以使用pmax@z选项：

	A	B
3	=A2.pmax@z(Close)	/从后向前取出股市最高点记录所在行号

3. 获取满足条件的成员序号



求2019年上证指数收盘价涨幅超过3%的交易日，当天的交易量涨幅。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

3. 获取满足条件的成员序号

我们需要找到股市收盘价涨幅超过3%的记录所在行号，再与上一个交易日比较得出结果。SPL如下，其中用到了pselect()函数来定位成员所在行号：

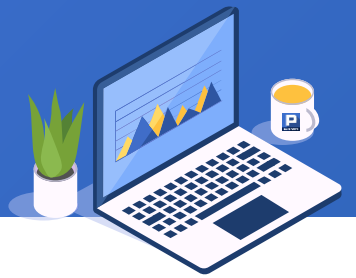
	A	B
1	=file("000001.csv").import@ct()	/导入数据文件
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年的股市记录
3	=A2.pselect@a(Close/Close[-1]>1.03)	/取出股市收盘价涨幅超过3%的记录所在行号，@a选项会返回所有满足条件的行号
4	=A3.new(A2(~).Date:Date, A2(~).Amount/A2(~-1).Amount:'Amount increase')	/循环用每天交易量和前日交易量计算涨幅

A3	Member
	161
	187
	211

A4	Date	Amount increase
	2019/02/25	1.758490566037736
	2019/03/29	1.3344827586206895
	2019/05/10	1.3908629441624365

我们可以看到，收盘价涨幅超过3%的这三天，交易量较前日大幅提升。

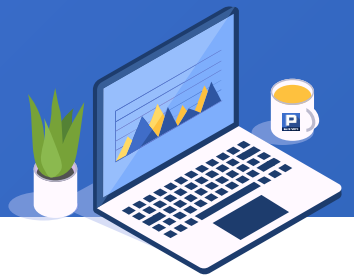
4. 成员在序列中的区段序号



下面是员工表，根据工资将员工分为8000以下、8000-12000和12000以上，并统计每组人数。

ID	NAME	BIRTHDAY	SALARY
1	Rebecca	1974-11-20	7000
2	Ashley	1980-07-19	11000
3	Rachel	1970-12-17	9000
4	Emily	1985-03-07	7000
5	Ashley	1975-05-13	16000
…	…	…	…

4. 成员在序列中的区段序号



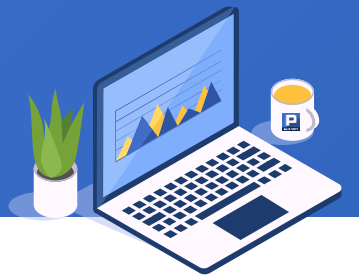
SPL如下，其中用到了整体定位函数pseg(x)：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from EMPLOYEE")	/查询员工表
3	[0,8000,12000]	/定义工资区间
4	=A2.align@a(A3.len(),A3.pseg(SALARY))	/使用pseg函数获取工资所在区间
5	=A4.new(A3 (#):SALARY,~.count():COUNT)	/统计每组的人数

A5

SALARY	COUNT
0	308
8000	153
12000	39

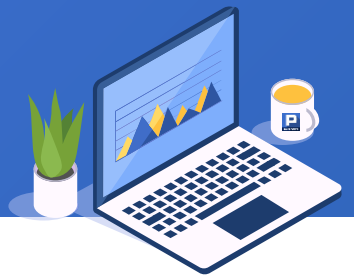
4. 成员在序列中的区段序号



下面是员工表，根据入职时间将员工分为10年以下、10-20年和20年以上，并统计每组的平均工资。

ID	NAME	HIREDATE	SALARY
1	Rebecca	2005-03-11	7000
2	Ashley	2008-03-16	11000
3	Rachel	2010-12-01	9000
4	Emily	2006-08-15	7000
5	Ashley	2004-07-30	16000
…	…	…	…

4. 成员在序列中的区段序号



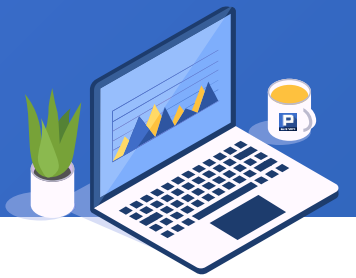
SPL如下，其中用到了整体定位函数pseg(x,y)：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from EMPLOYEE")	/查询员工表
3	[0,10,20]	/定义入职年限区间
4	=A2.align@a(A3.len(),A3.pseg(year(now())-~,year(HIREDATE)))	/使用pseg函数获取入职时间所在区间
5	=A4.new(A3(#):EntryYears,~.avg(SALARY):AvgSalary)	/统计每组的平均工资

A5

EntryYears	AvgSalary
0	6807.69
10	7417.78
20	7324.32

5. 获取排序后成员在排序前的序号



下面是员工表，求年龄最大的三名员工的入职顺序。

ID	NAME	BIRTHDAY	HIREDATE
1	Rebecca	1974-11-20	2005-03-11
2	Ashley	1980-07-19	2008-03-16
3	Rachel	1970-12-17	2010-12-01
4	Emily	1985-03-07	2006-08-15
5	Ashley	1975-05-13	2004-07-30
…	…	…	…

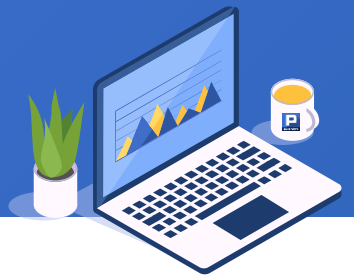
5. 获取排序后成员在排序前的序号

这里用到了A.psort() 函数获得排序后成员在排序前的序号，值得注意的是psort()函数不会改变原序列顺序。SPL如下：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from EMPLOYEE order by HIREDATE")	/查询员工表并按入职时间排序
3	=A2.psort(BIRTHDAY)	/获取员工生日的顺序号
4	=A2(A3.to(3).sort())	/在员工表中按生日前三的员工序号选出

A5	ID	NAME	BIRTHDAY	HIREDATE
	296	Olivia	1968-11-05	2006-11-01
	440	Nicholas	1968-11-24	2008-07-01
	444	Alexis	1968-11-12	2010-12-01

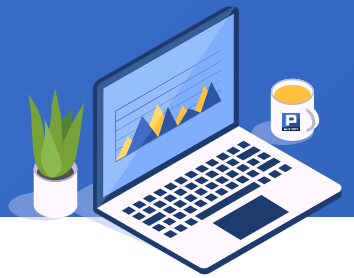
6. 序列的整体定位



下面是发帖记录表。按标签将帖子分组，并统计各标签出现频率。

ID	TITLE	Author	Label
1	Easy analysis of Excel	2	Excel,ETL,Import,Export
2	Early commute: Easy to pivot excel	3	Excel,Pivot,Python
3	Initial experience of SPL	1	Basics,Introduction
4	Talking about set and reference	4	Set,Reference,Dispersed,SQL
5	Early commute: Better weapon than Python	4	Python,Contrast,Install
…	…	…	…

6. 序列的整体定位

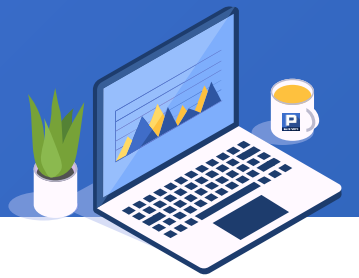


SPL如下，使用了A.pos()函数进行整体定位：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from PostRecord")	/查询发帖记录表
3	=A2.conj(Label.split(",")).id()	/将标签按逗号分隔后合并到一个序列，获得没有重复值的全部标签。
4	=A2.align@ar(A3.len(),A3.pos(Label.split(",")))	/使用pos()函数整体定位帖子的标签在全部标签中的位置。再使用align()函数的@r选项，按照定位分组。
5	=A4.new(A3(#):Label,~.count():Count).sort@z(Count)	/统计每个标签的帖子数量，按降序排列

A5	Label	Count
	SPL	7
	SQL	6
	Basics	5
	…	…

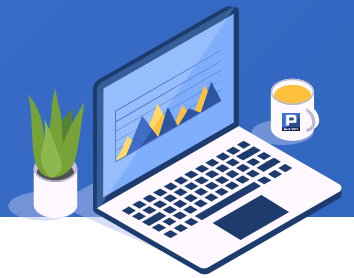
7. 判断是否序列成员



下面是世界各国官方语言表，查询官方语言包括中文和英文的国家。

Country	Language
China	Chinese
UK	English
Singapore	English
Singapore	Malay
Singapore	Chinese
Singapore	Tamil
Malaysia	Malay
…	…

7. 判断是否序列成员

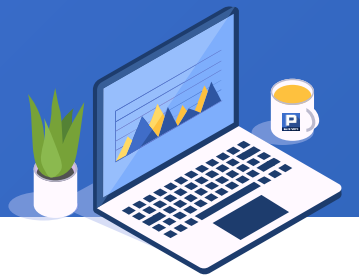


SPL如下，使用了A.contain()函数判断是否序列成员：

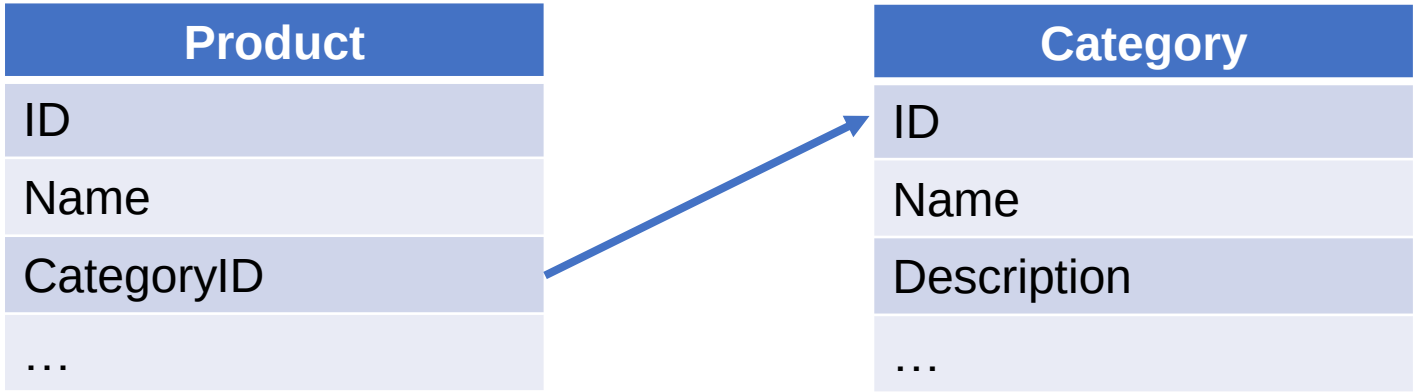
	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Language")	/查询官方语言表
3	=A2.group(Country)	/按国家分组
4	=A3.select(~.(Language).contain("Chinese","English"))	/使用contrain()函数判断当前国家的语言是否包含中文和英文。
5	=A4.(Country)	/取出国家列表

A5	Member
	Singapore

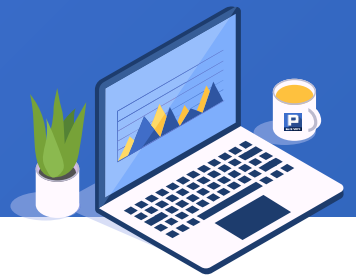
8. 查找主键所在行号



有产品表和类别表，要查询产品类别没有登记在类别表的数据，如下图：



8. 查找主键所在行号

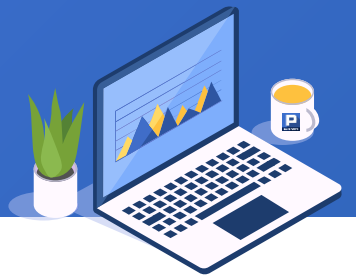


SPL如下，使用了A.pfind()函数查找主键所在行号：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Category").keys(ID)	/查询类别表，并设置主键为ID
3	=A1.query("select * from Product")	/查询产品表
4	=A3.select(A2.pfind(CategoryID)==0)	/使用pfind函数在类别表中查找主键等于类别ID的行号，返回0说明不存在。在产品表中选出类别ID不存在的记录。

A4	ID	Name	CategoryID	…
	12	German cheese		…
	26	Spun sugar	9	…

9. 取前N个/后N个记录对应的行号



求2019年上证指数收盘价最高的三天，交易额相对于前一日的涨幅。

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
2019/12/25	2980.4276	2981.8805	1.9E11
…	…	…	…

9. 取前N个/后N个记录对应的行号

我们需要知道股市收盘价最高的三天记录所在行号，再与上一个交易日比较得出结果。SPL如下，其中用到了ptop()函数来取最高三天所在行号：

	A	B
1	=file("000001.csv").import@ct()	/导入数据文件
2	=A1.select(year(Date)==2019)	/选出2019年的记录
3	=A2.ptop(-3, Close)	/使用ptop函数取出股市收盘价最高三天所在行号，-3表示从大到小取前三。如果是正整数表示从小到大取。
4	=A3.run(~=A2(~).Amount/A2(~+1).Amount-1)	/循环用每天交易量和前日交易量计算涨幅

A3	VALUE
	154
	156
	157

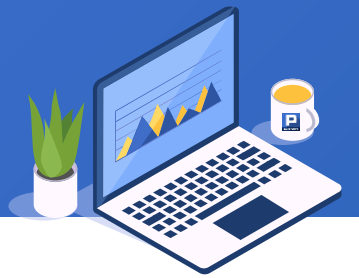
A4	VALUE
	-0.0278
	-0.0139
	0.0112

CONTENTS

06

选出

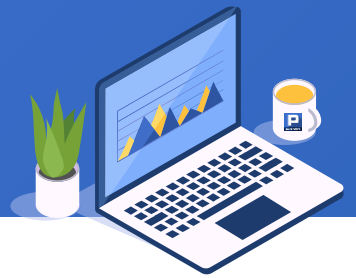
1. 取最小值对应记录



以成绩表为例，求一班数学最低分的学生ID。

CLASS	STUDENTID	SUBJECT	SCORE
Class one	1	English	84
Class one	1	Math	77
Class one	1	PE	69
Class one	2	English	81
Class one	2	Math	80
…	…	…	…

1. 取最小值对应记录



SPL如下，其中用到了minp()函数来取最小值所在记录，再从中取学生ID：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Scores where SUBJECT='Math' and CLASS='Class one'")	/查询一班的数学成绩
3	=A2.minp(SCORE)	/使用minp函数取出最低分所在记录
4	=A3.STUDENTID	/从记录中取学生ID

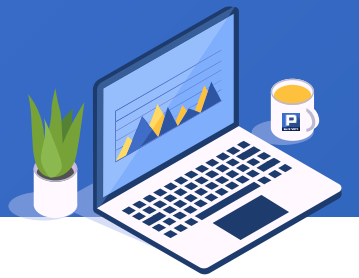
最小值所在记录不一定是唯一的，如果想返回所有记录，可以使用minp@a选项：

	A	B
3	=A2.minp@a(SCORE)	/取出所有最高分的记录
4	=A3.(STUDENTID)	/从多条记录中取学生ID序列

A3	CLASS	STUDENTID	SUBJECT	SCORE
	Class one	5	Math	60
	Class one	14	Math	60

A4	STUDENTID
	5
	14

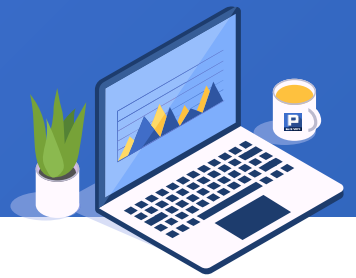
2. 取最大值对应记录



以奥运会奖牌榜为例，求奥运会总成绩蝉联第一名的最长届数的国家信息。

Game	Nation	Gold	Silver	Copper
30	USA	46	29	29
30	China	38	27	23
30	UK	29	17	19
30	Russia	24	26	32
30	Korea	13	8	7
…	…	…	…	…

2. 取最大值对应记录

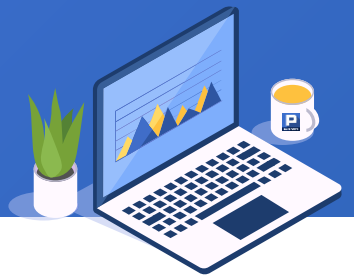


SPL如下，其中用到了maxp()函数取最大值对应记录：

	A	B
1	=file("Olympic.csv").import@cqt()	/导入奥运会历届排名
2	=A1.sort@z(Game, 1000000*Gold+1000*Silver+Copper)	/按第几届和总成绩降序排列
3	=A2.group@o1(Game)	/每届取一名，因为有序也就是第一名
4	=A3.group@o(Nation)	/将相邻的国家按原序分组
5	=A4.maxp(~.len())	/取最长的一组，也就是蝉联次数最多的

A5	Game	Nation	Gold	Silver	Copper
	10	USA	41	32	30
	9	USA	22	18	16
	8	USA	45	27	27
	7	USA	41	27	28

3. 选出符合条件的成员



世界城市人口表，如下图：

Continent	Country	City	Population
Africa	Egypt	Cairo	6789479
Asia	China	Shanghai	24240000
Europe	Britain	London	7285000

分栏列出欧洲和非洲人口超 200 万的城市名称及人口（每栏按从多到少排序），期望结果如下图：

Europe City	Population	Africa City	Population
Moscow	8389200	Cairo	6789479
London	7285000	Kinshasa	5064000
St Petersburg	4694000	Alexandria	3328196

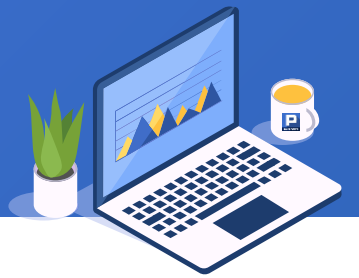
3. 选出符合条件的成员

SPL如下，其中用到了A.select()函数选出符合条件的成员：

	A	B
1	=connect("db").query("select * from World where Continent in('Europe', 'Africa') and Population >= 2000000")	/连接数据库，取出欧洲和非洲超过200万人口的记录
2	=A1.select(Continent:"Europe")	/使用select()函数取出欧洲数据
3	=A1.select(Continent:"Africa")	/使用select()函数取出非洲数据
4	=create('Europe City',Population,'Africa City', Population)	/按目标结构创建一个空序表
5	=A4.paste(A2.(City),A2.(Population),A3.(City),A3.(Population))	/使用paste()函数将值序列粘贴到对应列

A5	Europe City	Population	Africa City	Population
	Moscow	8389200	Cairo	6789479
	London	7285000	Kinshasa	5064000
	St Petersburg	4694000	Alexandria	3328196
	…	…	…	…

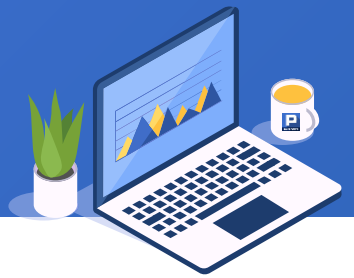
4. 根据区段序号返回序列中对应成员



以成绩表为例，统计英语科目的优秀、及格和不及格的人数。

CLASS	STUDENTID	SUBJECT	SCORE
Class one	1	English	84
Class one	1	Math	77
Class one	1	PE	69
Class one	2	English	81
Class one	2	Math	80
…	…	…	…

4. 根据区段序号返回序列中对应成员



SPL如下，其中用到了A.segp()函数取区段序号对应的序列中的成员：

	A	B
1	=connect("db").query("select * from Scores where SUBJECT='English'")	/连接数据库，查询英语成绩
2	=create(Assessment,Score).record(["fail",0,"pass",60,"excellent",90])	/创建分数与评价对照表
3	=A1.derive(A2.segp(Score,SCORE).Assessment:Assessment)	/使用segp函数返回分数对应的评价
4	=A3.groups(Assessment;count(1):Count)	/按评价分组统计人数

A4	Assessment	Count
	excellent	6
	fail	4
	pass	18

5. 排序

世界各国某年的经济指标如下，求中美两国差距较大的经济指标对比。

	A	B	C	D	E	F	G	H
1	Indicator	Last	Previous	Min	Max	Unit	Frequency	Range
2	Gover--nment Debt: % of GDP	105.7	107.3	31.8	107.3	%	Yearly	1969 - 2017
3		2017	2016	1974	2016			Updated on 2018---03-28
4	Business Confi--dence: Net Balance	18.6	21.6	-41.2	55	% Point	Monthly	Jan 1948 - Mar 2018
5		Mar-18	Feb-18	May-80	Jul-50			Updated on 2018---04-02
6	Foreign Direct Inves--tment	49,814.00	105,653.00	-75,269.00	246,224.00	USD mn	Quarterly	Mar 1960 - Dec 2017
7		Dec-17	Sep-17	Mar-14	Mar-15			Updated on 2018---03-21
8	Total Imports Growth	10.8	7.3	-35.4	33.2	%	Monthly	Jan 1990 - Feb 2018
9		Feb-18	Jan-18	May-09	May-10			Updated on Mar 2018
10	Money Supply M2	13,858,400.00	13837800.00	286,600.00	13,858,400.00	USD mn	Monthly	Jan 1959 - Feb 2018
11		Feb-18	Jan-18	Jan-59	Feb-18			Updated on 2018---03-29
12	Forecast: Gover--nment Revenue	7,612.00	7,322.72	3,270.25	7,612.00	USD bn	Yearly	2001 - 2022
13		2022	2021	2002	2022			Updated on 2017---10-10
14	Motor Vehicle Produ--ction	11,189,985.00	12,180,301.00	5,377,687.00	12,279,582.00	Unit	Semia--nnual	Dec 2001 - Dec 2017
15		Dec-17	Dec-16	Jun-12	Dec-02			Updated on 2018---03-08
16	Teled--ensity: Fixed Line	37.09	38.4	26.44	67.64	Number	Yearly	1960 - 2016
17		2016	2015	1960	2000			Updated on 2017
18	Natural Gas: Exports	65,542.00	50,502.00	4,000.00	65,542.00	Cub m mn	Yearly	1995 - 2016
19		2016	2015	1996	2016			Updated on 2017---06-13
20	Real GDP Growth	2.6	2.3	-4.1	13.4	%	Quarterly	Mar 1948 - Dec 2017
21		Dec-17	Sep-17	Jun-09	Dec-50			Updated on Mar 2018
22	Labour Produ--ctivity Growth	1.36	0.9	-2.65	9.72	%	Quarterly	Mar 1949 - Dec 2017
23		Dec-17	Sep-17	Mar-74	Dec-50			Updated on 2018---03-28
24	Domestic Credit Growth	5.1	4.9	-2.4	10	%	Quarterly	Dec 2002 - Dec 2016
25		Dec-16	Sep-16	Jun-10	Mar-08			Updated on 2018---03-19

United States | China | Japan | Germany | France | United Kingdom

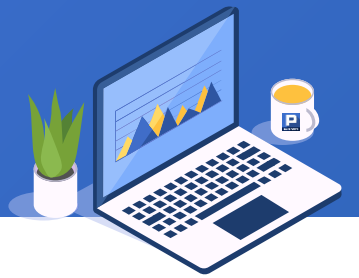
5. 排序

SPL如下，其中用到了A.sort()函数对数据排序：

	A	B
1	=file("Indicators.xlsx").xlsopen()	/打开excel文件
2	=A1.xlsimport@t(Indicator,Last).select(Indicator!=null)	/导入第一页（美国），选出指标不为空的记录
3	=A1.xlsimport@t(Indicator,Last;"China").select(Indicator!=null)	/导入中国页，选出指标不为空的记录
4	=A2.join(Indicator,A3:Indicator,Last:'China').rename(Last:'United States')	/美国数据按照指标名称连接增加中国指标字段
5	=A4.sort@z(abs('United States'-'China'))	/按中美指标的差值的绝对值降序排列。其中用到了sort函数进行排序，@z选项表示降序

A5	Indicator	United States	China
	Minerals Production	2118592432	4358945768
	Number of Subscriber Mobile	416684000	1364934000
	Employed Persons	155215000	776400000
	Exports: ICT Goods	1.4175230767E8	6.0755925913E8
	…	…	…

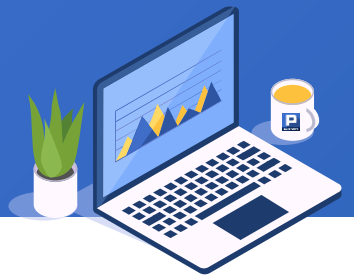
6. 取前N名/后N名记录



以成绩表为例，求每班各科前两名。

CLASS	STUDENTID	SUBJECT	SCORE
Class one	1	English	84
Class one	1	Math	77
Class one	1	PE	69
Class one	2	English	81
Class one	2	Math	80
…	…	…	…

6. 取前N名/后N名记录

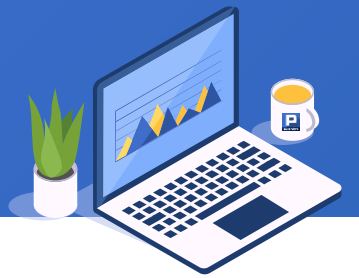


SPL如下，其中用到了A.top()函数取前N名/后N名：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Scores")	/查询学生成绩
3	=A2.group(CLASS,SUBJECT;~.top(-2;SCORE):TOP2)	/按班级和学科分组并取出每组分数前两名
4	=A3. conj(TOP2)	/将所有班级各科的前两名记录合并

A4	CLASS	STUDENTID	SUBJECT	SCORE
	Class one	4	English	96
	Class one	9	English	93
	Class one	13	Math	97
	Class one	10	Math	97
	…	…	…	…

7. 查找主键所在记录



有课程表和选课表，想查看学生所选课程：

Course		
ID	NAME	TEACHERID
1	Environmental protection and …	5
2	Mental health of College Students	1
3	Computer language Matlab	8
…	…	…

SelectCourse		
ID	STUDENTID	COURSE
1	59	2,7
2	43	1,8
3	52	2,7,10
…	…	…

ID	STUDENTID	COURSE1	COURSE2	COURSE3	…
1	59	Mental health of College Students	Into Shakespeare		…
2	43	Environmental protection and …	Modern economics		…
3	52	Mental health of College Students	Into Shakespeare	Music appreciation	…
…	…	…	…	…	…

7. 查找主键所在记录

SPL如下，其中用到了find()函数查找主键所在的记录：

	A	B
1	=connect("db")	/连接数据库
2	=A1.query("select * from Course").keys(ID)	/读取课程表，并设置主键ID
3	=A1.query("select * from SelectCourse")	/读取学生选课表
4	=A3.run(COURSE=COURSE.split@cp())	/将选课表中的课程按逗号拆分后赋值给课程字段
5	=A4.max(COURSE.len())	/找到选课最多的数量
6	=create(ID,NAME, ${A5.("COURSE"+string(~)).concat@c()})	/创建空表，课程列按照最多的数量创建
7	>A4.run(A6.record([ID,STUDENT_NAME]|COURSE.(A2.find(~).Name)))	/循环选课表，使用find()函数查找课程主键所在记录，找到课程名称。并把合并起来的记录插入到A6序表。

A6	ID	STUDENTID	COURSE1	COURSE2	COURSE3
	1	59	Mental health of College Students	Into Shakespeare	
	2	43	Environmental protection and …	Modern economics	
	3	52	Mental health of College Students	Into Shakespeare	Music appreciation
	…	…	…	…	…

	A	B
7	>A4.run(A6.record([ID,STUDENT_NAME]|COURSE.(~.row(A2).Name)))	/A7也可以用row()函数来查找在序列中的行号。A.find(~)等价于~.row(A)

THANKS

感谢观看