# Order-related calculation

# CONTENTS

## 01 Sequence number access

1. Single sequence number access
2. Multiple sequence number access

## 02 Basic operations

1. Operation between two sequences: sequence comparison
2. Operation between two sequences: sum and difference
3. Operation between two sequences: union and intersection
4. Operation between two sequences: the four fundamental operations
5. Operation between two sequences: member comparison
6. Operation between sequence and single value

# CONTENTS

SPL

## 03 Aggregation

1. Aggregation of a single sequence: sum
2. Aggregation of a single sequence: maximum, minimum
3. Aggregation of a single sequence: average
4. Aggregation of a single sequence: count
5. Aggregation of a single sequence： logic and operation
6. Aggregation of a single sequence: logic or operation
7. Aggregation of a single sequence: non repeating count
8. Aggregation of a single sequence: median
9. Aggregation of a single sequence: ranking
10. Aggregation of sequences of sequences: sum sequence
11. Aggregation of sequences of sequences： union sequence and difference sequence
12. Aggregation of sequences of sequences： intersection sequence
13. Aggregation of sequences of sequences: XOR sequence

## 04 Loop calculation

1. Loop function
2. Symbol
3. Positioning calculation
4. Iteration calculation

# CONTENTS

## 05 Positioning

1. Locate a member's position in the sequence
2. Take the row number of the record corresponding to the maximum / minimum value
3. Get the member sequence number that meets the condition
4. Segment sequence number of member in sequence
5. Get the original sequence number of a sorted member
6. Overall positioning of sequence
7. Determine whether it is a sequence member
8. Find row number of primary key
9. Get the row numbers corresponding to TopN records

## 06 Select

1. Get the record corresponding to the minimum value
2. Get the record corresponding to the maximum value
3. Select member that meets the condition
4. Return the corresponding member in the sequence according to the section number
5. Sort
6. Get TopN records
7. Find the record where the primary key is located

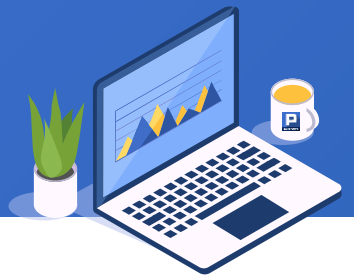
# CONTENTS



01

## Sequence number access

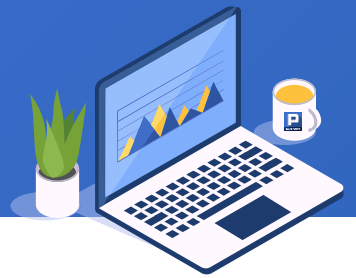
# ✦ 1. Single sequence number access



Find the trading information of the first trading day and the last trading day of the Shanghai stock index in 2019.

| 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. Single sequence number access

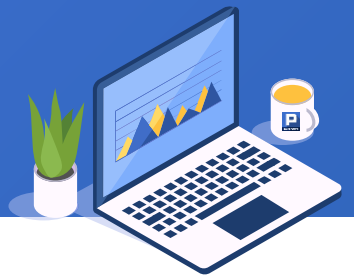


SPL is as follows, where A() and A.m() are used to get members:

| | A | B |
|---|---|---|
| 1 | =file("000001.csv").import@ct() | /Import data file |
| 2 | =A1.select(year(Date)==2019).sort(Date) | /Select records of 2019 and sort by date |
| 3 | =A2(1)\|A2.m(-1) | /Retrieve the information of the first and last trading days of the stock market |

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

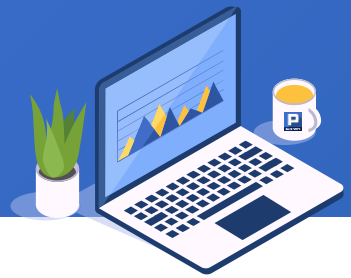
# ✦ 1. Single sequence number access



Take the employee table as an example to count the average salaries of state California, Texas, New York, and Florida. Employees in other regions are put into new group for statistics.

| 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. Single sequence number access



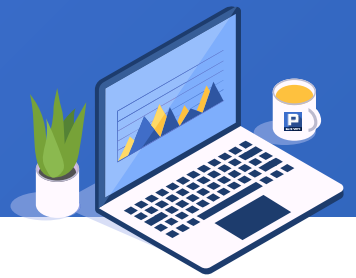
SPL is as follows, in which A.p(-1) is used to obtain the sequence number of the last member:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from EMPLOYEE") | /Query employee table |
| 3 | [California,Texas,New York,Florida] | /Create region sequence |
| 4 | =A2.align@an(A3,STATE) | /The employee table is grouped in alignment by region, the @a option returns all matching members in a group, and the @n option puts all mismatched members in the new group. |
| 5 | =A4.new(if (#>A3.p(-1),"Other",STATE):STATE,~.avg(SALARY):AvgSalary) | /Calculate the average salary of each group and generate a new sequence table. Use A.p(-1) to get the sequence number of the last member and rename the region of the last group to other. |

A5

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

# ✦ 2. Multiple sequence number access



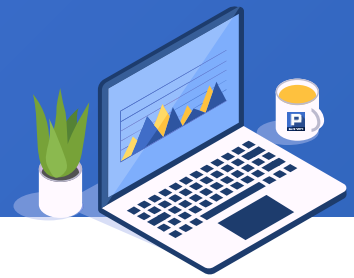
There is a table for recording daily attendance information, as shown below:

| 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 |

Every seven pieces of data are a group. We want to convert them into the following result:

| 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. Multiple sequence number access

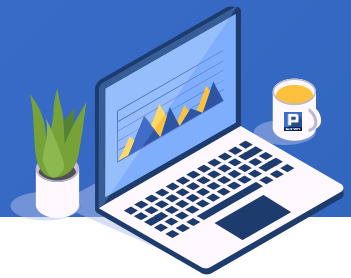


First create the target data structure, and then organize every seven records into the required structure to fill in the data. SPL is as follows, in which A() and A.m() multiple sequence number access members are used:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from DailyTime order by Per_Code,Date,Time") | /Query data and sort by Per_code, date and time |
| 3 | =A2.group(Per_Code,Date) | /Group by Per_code and date |
| 4 | =create(Per_Code,Date,In,Out,Break,Return) | /Create an empty sequence table to store the final result |
| 5 | =A3.(~([1,7,2,3,1,7,5,6])) | /For each group, use A([1,7,2,3,1,7,5,6]) to take out the records one by one, which is an orderly all day record. |
| 6 | =A5.conj([~.Per_Code,~.Date]|~.(Time).m([1,2,3,4])|[~.Per_Code,~.Date]|~.(Time).m([5,6,7,8])) | /Organize all the data of each record into a sequence. A.m() is used to access multiple members. |
| 7 | >A4.record(A6) | /Add records to the sequence table created by A4. |

| 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. Multiple sequence number access

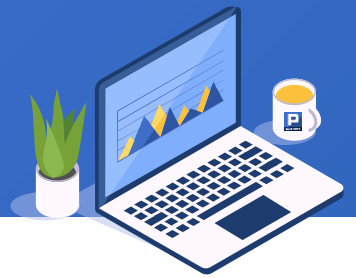


Find the prime number within 100. The SPL is as follows, in which the step() function is used to take members according to the fixed span:

| | A | B |
|---|---|---|
| 1 | =to(100) | /Create 1 to 100 sequence |
| 2 | =to(2,10) | /Create 2 to 10 sequence |
| 3 | =A2.(A1.step(~,~*2)) | /For each member in A2, the n-multiple within 100 is calculated (n > 1) |
| 4 | =A1.to(2,)\A3.conj() | Remove 1 and all composite numbers within 100, and get all prime numbers within 100. A3.conj() gets the composite numbers within 100. |

| A4 | Member |
|---|---|
| | 2 |
| | 3 |
| | 5 |
| | 7 |
| | 11 |
| | … |

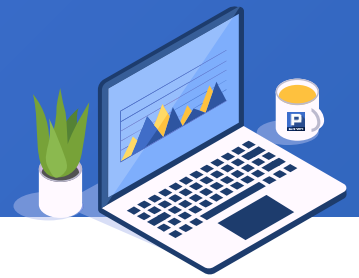
## ✦ 2. Multiple sequence number access



Find the increase rate of the closing price of Shanghai stock index in the last 10 trading days of 2019 compared with that of the previous day.

| 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. Multiple sequence number access

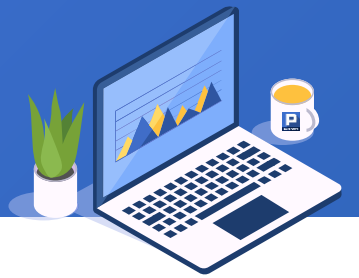


SPL is as follows, in which the A.p() function is used to return the sequence number of the last 10 members:

| | A | B |
|---|---|---|
| 1 | =file("000001.csv").import@ct() | /Import data file |
| 2 | =A1.select(year(Date)==2019).sort(Date) | /Select records of 2019 and sort by date |
| 3 | =A2.p(to(-10,-1)) | /Use the p() function to return the sequence number of the last 10 members |
| 4 | =A3.new(A2(~).Date:Date, string(A2(~).Close/A2(~-1).Close-1, "0.000%" ):Increase) | /Calculate the increase rate of the closing price of each trading day compared with the previous trading day in loop |

| 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% |
| | … | … |

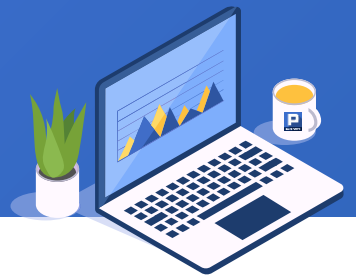
## ✦ 2. Multiple sequence number access



The following is a company's organizational structure table. Query all superior organizations of the Beijing market research team.

| 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. Multiple sequence number access



SPL is as follows, in which rvs() function is used to reverse the sequence:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Organization") | /Query organization table |
| 3 | >A2.switch(PARENT_ID,A2:ID) | /The foreign key PARENT_ID is mapped to the record where the ID is located to realize self join |
| 4 | =A2.select@1(ORG_NAME=="Beijing Market Research Team") | /Select the record of Beijing Market Research Team |
| 5 | =A4.prior(PARENT_ID) | /Use the prior function to find out the superior unit |
| 6 | =A5.rvs().(ORG_NAME).concat(" / ") | /Use rvs function to arrange superior units from superior to subordinate |

**A6**

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

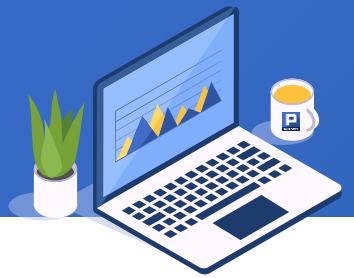
# CONTENTS



02

# Basic operations

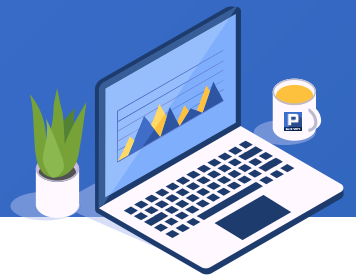
# ✦ 1. Operation between two sequences: sequence comparison



Take the Olympic medal table as an example to find out which Olympic Games China ranked higher than Russia in medal table.

| 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. Operation between two sequences: sequence comparison



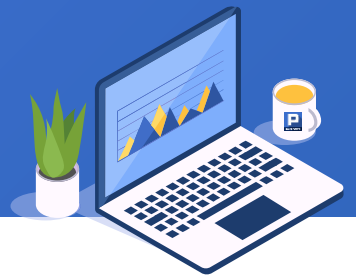
SPL is as follows, in which the ">" symbol is used to compare the sequence, and the corresponding members will be compared in sequence:

| | A | B |
|---|---|---|
| 1 | =file("Olympic.csv").import@cqt() | /Import the ranking of Olympic Games |
| 2 | =A1.run(Medal=Medal.split@c()) | /Medal field is split into sequence by commas |
| 3 | =A2.group(Game) | /Group by game |
| 4 | =A3.select(~.select(Nation=="China").Medal>~.select(Nation=="Russia").Medal) | /Using the ">" symbol to compare the medal sequence of China and Russia, the number of gold medals, silver medals and bronze medals will be compared in sequence, and the game number will be selected during which China ranked higher. |
| 5 | =A4.(Game) | /List the games |

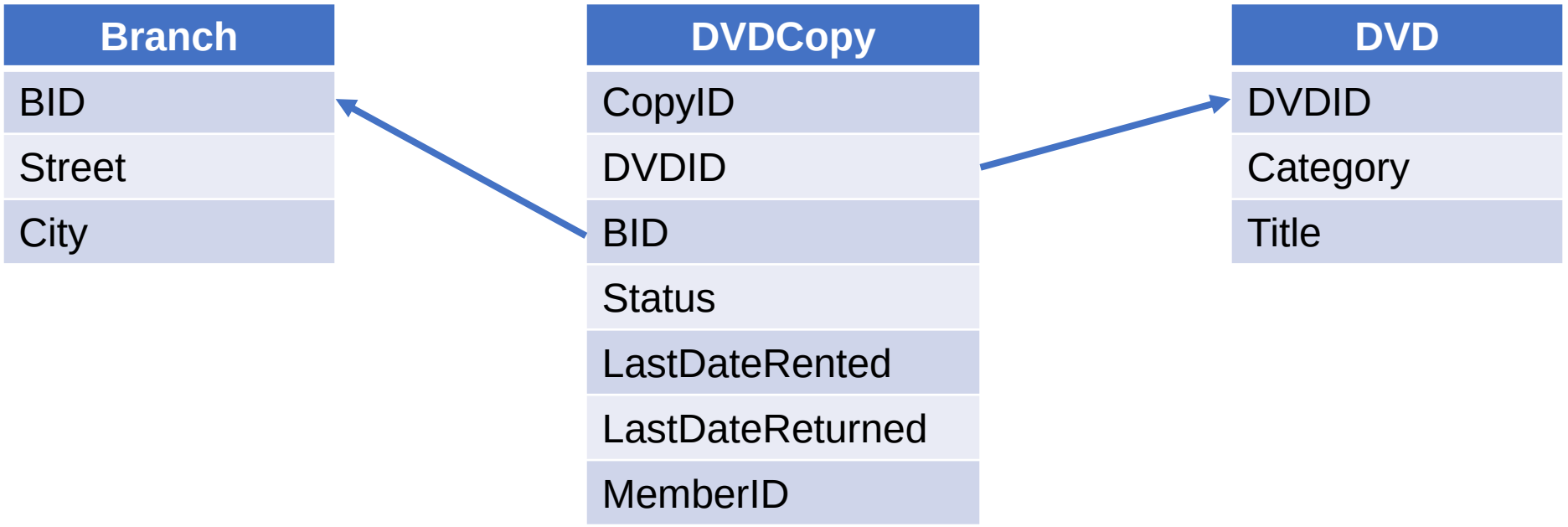
| A5 | Game |
|---|---|
| | 23 |
| | 25 |
| | 28 |
| | 29 |
| | 30 |

Similarly, we can use "<" and "==" to compare sequence.

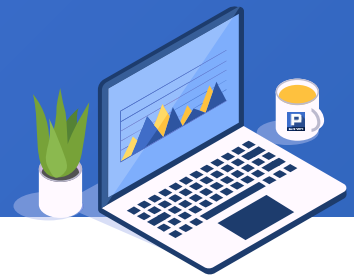
# ✦ 2. Operation between two sequences: sum and difference



Find out the out of stock branches, that is, the existing DVD copies less than 4 categories. The branch table stores the information of DVD branches; the DVD table stores the title and category information of DVD; the DVDCopy table stores multiple copies of DVD, and the DVD copy is the real disc, which is stored in each branch in the physical form.



# ✦ 2. Operation between two sequences: sum and difference



SPL is as follows, in which the sum of "|" and the difference of "\" are used:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =Branch=A1.query("select * from Branch") | /Read branch information and define it as branch variable |
| 3 | =DVD=A1.query("select * from DVD") | /Read DVD information and define it as DVD variable |
| 4 | =DVDCopy=A1.query("select * from DVDCopy") | /Read DVDCopy information and define it as DVDCopy variable |
| 5 | =DVDCopy.switch(DVDID,DVD:DVDID; BID,Branch:BID) | /Switch the DVDID field of DVDCopy to the corresponding record in DVD |
| 6 | =DVDCopy.select(STATUS!="Miss" && LASTDATERETURNED!=null) | /Filtering lost and unreturned DVD copies |
| 7 | =A6.group(BID) | /Group filtered data by BID |
| 8 | =A7.select(~.icount(DVDID.CATEGORY)<4) | /Select stores with DVD copy less than 4 categories |
| 9 | =A8.(BID) \| (Branch \ A7.(BID)) | /Out of stock branches. A8.(BID) refers to the branches with DVD copy less than 4 categories, Branch \ A7.(BID) refers to the branch without DVDCopy. |

| A9 | BID | STREET | CITY |
|---|---|---|---|
| | B002 | Street2 | Houston |
| | B003 | Street3 | LA |
| | B004 | Street4 | Lincoln |

# 3. Operation between two sequences: union and intersection

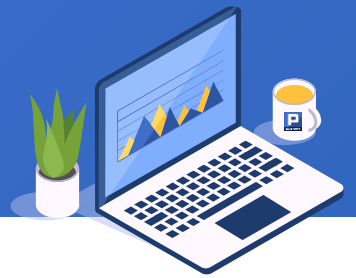
The municipalities directly under the central government of China are Beijing, Tianjin, Shanghai and Chongqing, and the first tier cities are Shanghai, Beijing, Shenzhen and Guangzhou. What are the municipalities directly under the central government and the first tier cities? Which cities are both directly under the central government and first tier cities?

| | A | B |
|---|---|---|
| 1 | [Beijing,Tianjin,Shanghai,Chongqing] | /Municipality directly under the Central Government |
| 2 | [Shanghai,Beijing,Shenzhen,Guangzhou] | /First-tier cities |
| 3 | =A1&A2 | /the union of municipalities directly under the central government and first tier cities |
| 4 | =A1^A2 | /intersection, i.e. the first tier city in the municipalities directly under the central government |

| A4 | Member |
|---|---|
| | Beijing |
| | Shanghai |

| A3 | Member |
|---|---|
| | Beijing |
| | Tianjin |
| | Shanghai |
| | Chongqing |
| | Shenzhen |
| | Guangzhou |

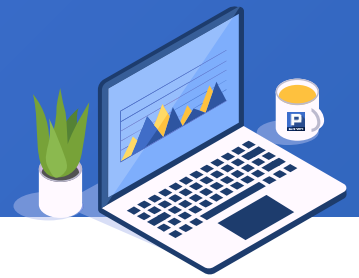
# 4. Operation between two sequences: the four fundamental operations



Calculate the daily relative yield of Shenzhen 300 (399007) to Shenzhen Composite Index (399001) from December 24 to 26, 2019.

| 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 |
| … | … | … | … | … | … |

# ✦ 4. Operation between two sequences: the four fundamental operations



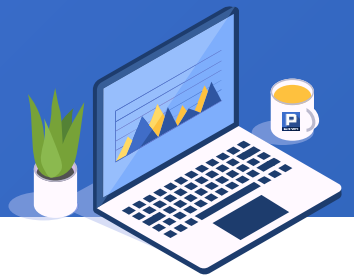
A ?? B is used to perform the "?" operation on the bitwise members of two sequences, where ?∈{+,-,*,/,%,\}.

SPL is as follows:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =["399007","399001"].(A1.query("select * from StockIndex where code=? and date between '2019-12-23' and '2019-12-26'",~)) | /Read the data of Shenzhen 300 and Shenzhen composite index from December 23 to 26, 2019, and take the 23rd to calculate the increase |
| 4 | =A2.(~.calc(to(2,4),Close/Close[-1])) | /Calculate the daily increase from the 24th to the 26th |
| 5 | =A3(1)--A3(2) | /The two sequence subtract by bit, i.e. relative yield |

| A5 | Member |
|---|---|
| | 0.0031349096521252617 |
| | 0.0011897141619391371 |
| | -4.4910504685946595E-4 |

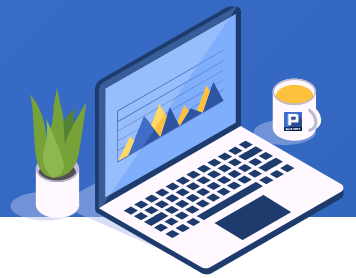
# 5. Operation between two sequences: member comparison



The following is the file generated after random sampling. Compare whether the same IDs are selected for the two random sampling. Part of the data in the file is as follows:

| 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 |
| … | … | … |

# ✦ 5. Operation between two sequences: member comparison



SPL is as follows, in which cmp() function is used to compare the members of two sequences bit by bit:

| | A | B |
|---|---|---|
| 1 | =file("p_old.csv").import@ct() | /Read the first output file |
| 2 | =file("p_new.csv").import@ct() | /Read the second output file |
| 3 | =cmp(A1.(ID),A2.(ID)) | /Compare whether the IDs generated twice are identical (the member values are equal and the order is the same) |

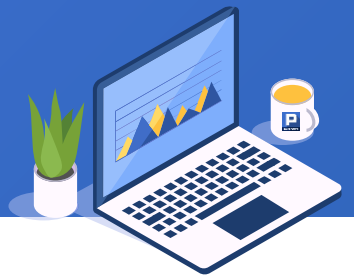
| A3 | Member |
|---|---|
| | 0 |

A result of 0 indicates that the IDs in the two files are exactly the same.

If the order of IDs may be different, you can use the eq() function to compare whether the members of the two sequences are the same:

| | A | B |
|---|---|---|
| 3 | =A1.(ID).eq(A2.(ID)) | /Compare the ID values generated twice to see if they are the same. The order is not required to be the same. |

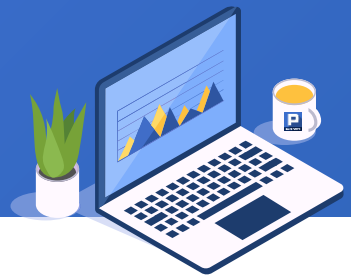
## 6. Operation between sequence and single value



In the same example, the order of IDs of two files is the same.  Compare how many pieces of data are exactly the same.

| 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 |
| … | … | … |

## 6. Operation between sequence and single value



SPL is as follows, where "|" is used to combine sequence and single value:

| | A | B | C |
|---|---|---|---|
| 1 | =file("p_old.csv").import@ct() | | /Read the first output file |
| 2 | =file("p_new.csv").import@ct() | | /Read the second output file |
| 3 | for A1.len() | =cmp(A1(A3),A2(A3)) | /Compare each record of two files in loop |
| 4 | | =@ \| B3 | /Combine the results of each comparison with the B4 value |
| 5 | =B4.count(~==0) | | /Count how many comparisons are equal |

| B4 | Member |
|---|---|
| | 0 |
| | 0 |
| | 0 |
| | … |

| A5 | Member |
|---|---|
| | 11302 |

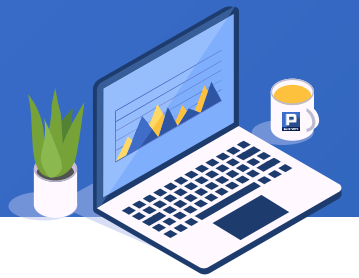
# CONTENTS



03

# Aggregation

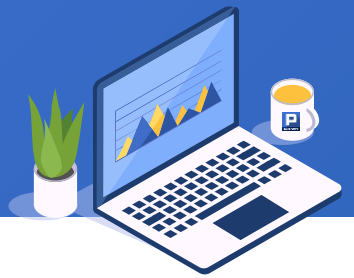
# ✦ 1. Aggregation of a single sequence: sum



The following is the urban GDP table. Calculate the per capita GDP of municipalities directly under the central government, first tier cities and second tier cities respectively.

| 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. Aggregation of a single sequence: sum

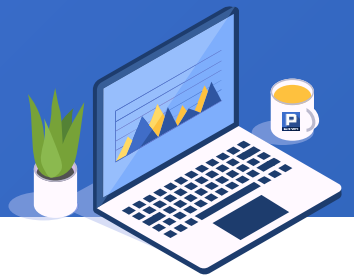


SPL is as follows, in which sum() function is used for summing.

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from GDP") | /Query city GDP table |
| 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] | /Enumerate municipalities directly under the central government, first tier cities and second tier cities |
| 4 | =A2.enum@r(A3,City) | /Enumeration grouping by city |
| 5 | =A4.new(A3(#):Area,~.sum(GDP)/~.sum(Population)*10000:CapitaGDP) | /Per capita GDP of each group |

| 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 |

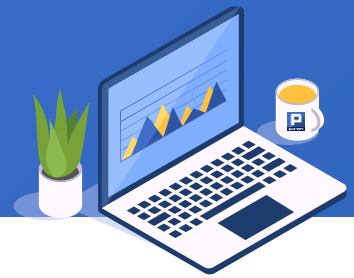
# ✦ 2. Aggregation of a single sequence: maximum, minimum



Take the order table as an example to merge the orders with repeated time periods of customer anatr.

| 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 |
| … | … | … | … | … |

# ✦ 2. Aggregation of a single sequence: maximum, minimum



SPL is as follows, where max() function is used to take the maximum value and min() function to take the minimum value:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Orders where Customer='ANATR' order by OrderDate") | /Select the order information of customer anatr and sort by order date |
| 3 | =A2.group@i(OrderDate>max(FinishDate[,-1])) | /Group orders with an order date less than the previous finish date |
| 4 | =A3.new(Customer,~.min(OrderDate):OrderDate,~.max(FinishDate):FinishDate) | /Take the earliest order date of each group as the order date and the latest finish date as the finish date |

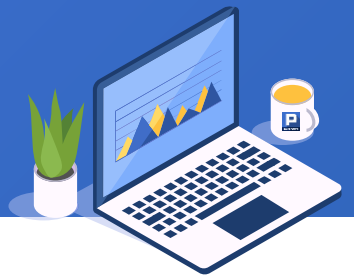
**A3**

| 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 |
| … | … | … |

# ✦ 3. Aggregation of a single sequence: average



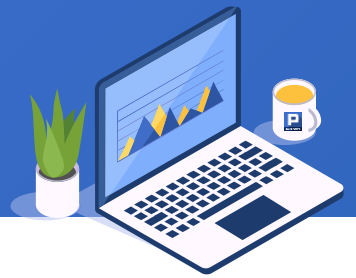
Some data in the employee table are as follows:

| 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 |
| … | … | … | … | … |

Calculate the average salary of each department in different regions, and want to convert it into the following result:

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

# 3. Aggregation of a single sequence: average



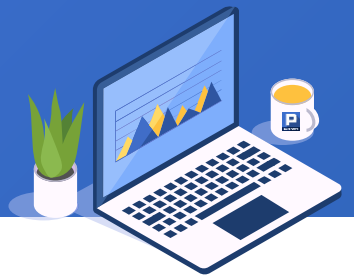
SPL is as follows, in which avg() function is used to calculate the average value:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from EMPLOYEE") | /Query employee table |
| 3 | =A2.groups(DEPT,STATE;avg(SALARY):AvgSalary) | /Group aggregation, use avg function to calculate the average salary of each department in each region |
| 4 | =A3.pivot(DEPT;STATE, AvgSalary) | /Transpose according to target table |

A4

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

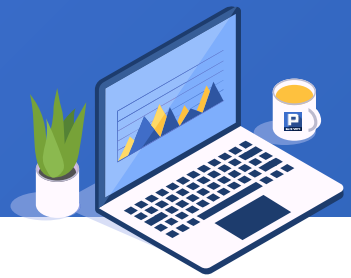
# 4. Aggregation of a single sequence: count



Take the score sheet as an example to find the number of students who failed in each subject in class one.

| 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. Aggregation of a single sequence: count

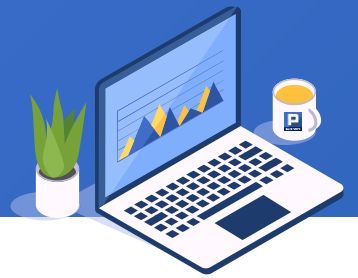


SPL is as follows, in which count() function is used to count:

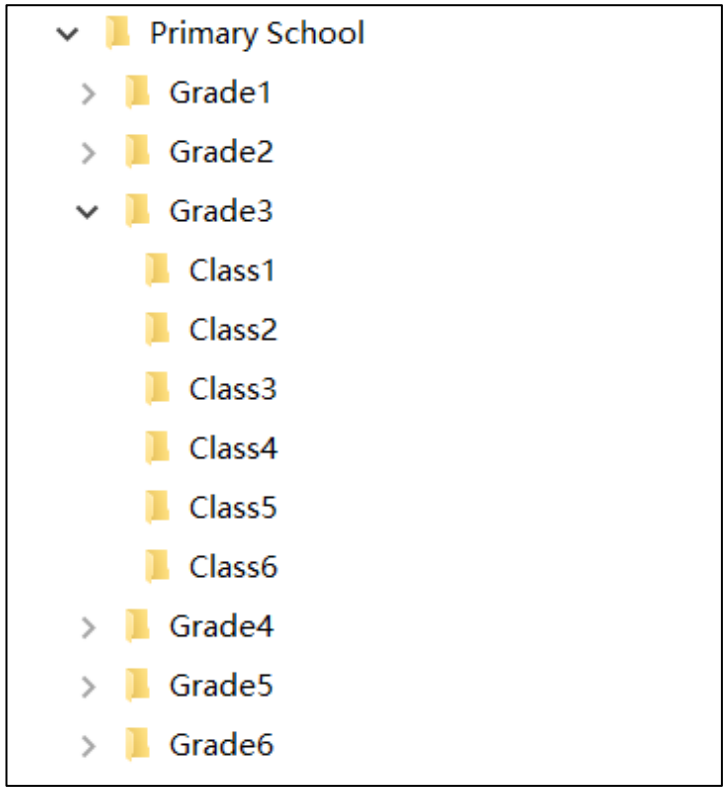
| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Scores where CLASS='Class one'") | /Query the scores of Class One students |
| 3 | =A2.groups(SUBJECT; count(SCORE<60):FailCount) | /Group aggregation, in which the count function is used to count the number of failures whose score is less than 60 |

| A3 | SUBJECT | FailCount |
|---|---|---|
| | English | 2 |
| | Math | 0 |
| | PE | 2 |

# 5. Aggregation of a single sequence：logic and operation

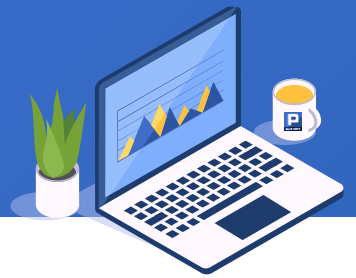


A primary school investigates terminals that students can use for online teaching to see if all students can use mobile phones. The questionnaire of each class and summary directory are as follows:




| 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 |
| … | … | … |

# ✦ 5. Aggregation of a single sequence: logic and operation



The A.ifn() function is used to get the first non null member, and the A.cand() function is used to perform and operation on the member. SPL is as follows:

| | A | B | |
|---|---|---|---|
| 1 | =directory@ps("D:/Primary School") | | /Recursively traversing directories, listing all files |
| 2 | for A1 | =file(A2).xlsimport@t() | /Import each class questionnaire excel file in loop |
| 3 | | =B2.([TERMINAL,"Phone"].ifn().split@c().pos("Phone") > 0) | /When the terminal in the questionnaire is not filled in, it is not considered that the mobile terminal is not supported. Use ifn() function to ensure this is true. |
| 4 | =B3.cand() | | /Use the cand() function to calculate whether the members of B3 are all true |

**A1**

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

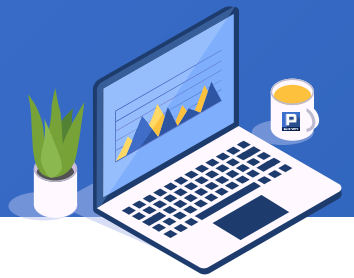
**B3**

| Member |
|---|
| true |
| true |
| true |
| … |

**A4**

| Member |
|---|
| false |

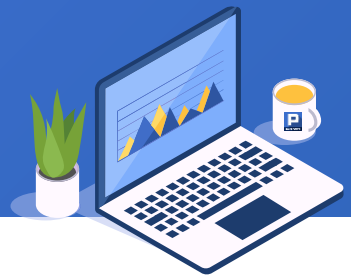
# 6. Aggregation of a single sequence: logic or operation



Part of the data in the sales table is as follows. Query whether the customer rattc has ranked among the top three monthly sales in 2014.

| 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 |
| … | … | … | … | … |

# ✦ 6. Aggregation of a single sequence: logic or operation



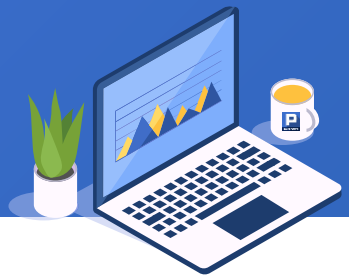
SPL is as follows, in which A.cor() function is used to perform or operate on members:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from sales") | /Connect to database and read the sales table |
| 2 | =A1.select(year(OrderDate)==2014) | /Select 2014 data |
| 3 | =A2.group(month(OrderDate)) | /Use the group function to group 2014 data by month |
| 4 | =A3.(~.groups(Customer; sum(Amount):Amount)) | /Group and aggregate sales amount by customer for each group |
| 5 | =A4.new(Customer, ~.top(-3;sum(Amount)):Top3) | /Cycle monthly data to calculate the top 3 customers of monthly sales |
| 6 | =A5.(Top3.(Customer).pos("RATTC")>0) | /Judge whether the top three of each month include customer rattc |
| 7 | =A6.cor() | /Use the cor() function to calculate whether the member of A6 exists true |

| A6 | Member |
|---|---|
| | false |
| | false |
| | true |
| | … |

| A7 | Member |
|---|---|
| | true |

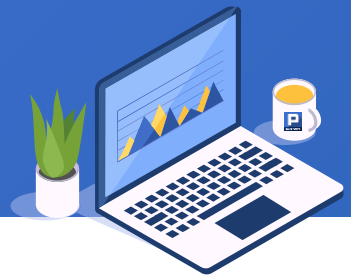
# ✦ 7. Aggregation of a single sequence: non repeating count



Analyze which field in the data file is the serial number. Some data are as follows:

| 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 |
| … | … | … | … | … | … |

# ✦ 7. Aggregation of a single sequence: non repeating count

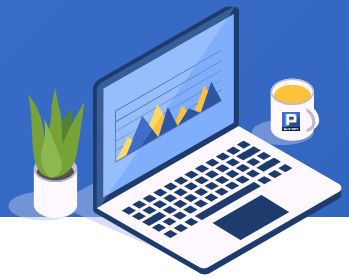


SPL is as follows, in which the icount() function is used for non repeated counting:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | =file("titanic_train.csv").import@cqt() | | | /Read data file |
| 2 | =A1.fno().new(A1.fname(~):Name,A1.field(~).icount():DCount) | | | /Use the icount() function to count the number of non repeating members for each field |
| 3 | =A2.select(DCount==A1.len()) | | | /Select the fields with the same length of non repeated count and all data |
| 4 | if (A3.len() > 1) | =A3.select(like@c(Name,"*id*")) | | /If there are more than one field satisfying the condition, select the one whose field name contains id. |
| 5 | | if (B4.len() > 0) | >A3=B4 | /If the field name contains id, assign the selected field to A3 |
| 6 | =A3.minp(len(Name)).Name | | | /Select the field name with the shortest name from the selected fields |

| A6 | Value |
|---|---|
| | PassengerId |

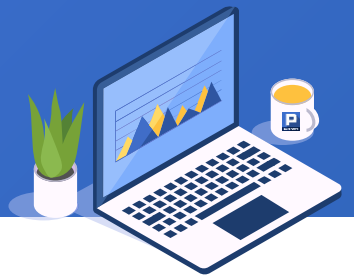
# ✦ 8. Aggregation of a single sequence: median



Part of the data in the employee table is as follows. Calculate the median salary of each department.

| 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 |
| … | … | … | … |

# 8. Aggregation of a single sequence: median

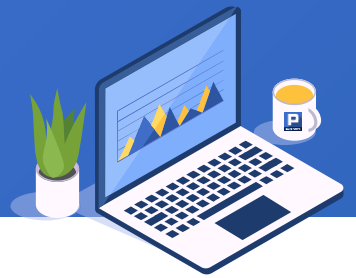


SPL is as follows, in which the median() function is used:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from employee") | /Connect to database, read employee table |
| 2 | =A1.groups(DEPT;median(;SALARY)) | /Group and aggregate by department to calculate the median salary of each department |

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

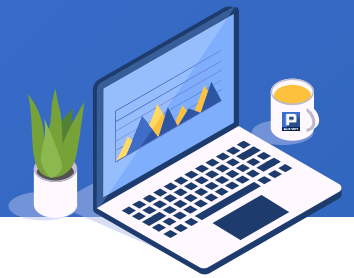
# 9. Aggregation of a single sequence: ranking



Take the score table as an example to find the ranking of the total score of the student whose student id is 8 in class one.

| 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 |
| … | … | … | … |

# 9. Aggregation of a single sequence: ranking

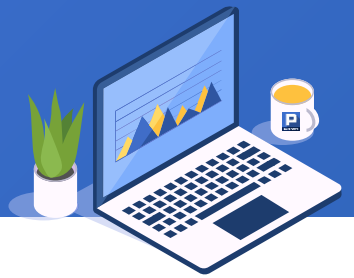


SPL is as follows, in which the A.rank() function is used to calculate the ranking:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from SCORES where CLASS='Class one'") | /Connect the data source and read the scores of the class one |
| 2 | =A1.groups(STUDENTID;sum(SCORE):TotalScore) | /Group and aggregate the total score of each student |
| 3 | =A2.select(STUDENTID==8).TotalScore | /Select the total score of the student whose id is 8 |
| 4 | =A2.rank@z(A3:TotalScore) | /Use the A.rank() function to calculate the ranking, and use the @z option to rank the total score from large to small |

| A4 | Value |
|---|---|
| | 13 |

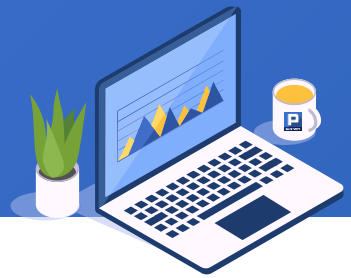
# 9. Aggregation of a single sequence: ranking



The prediction results of test data after intelligent modeling are as follows. Calculate the AUC index of the model:

| 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 |
| … | … | … |

# 9. Aggregation of a single sequence: ranking

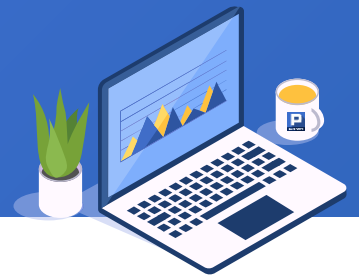


SPL is as follows, in which the ranks() function is used to calculate the ranking:

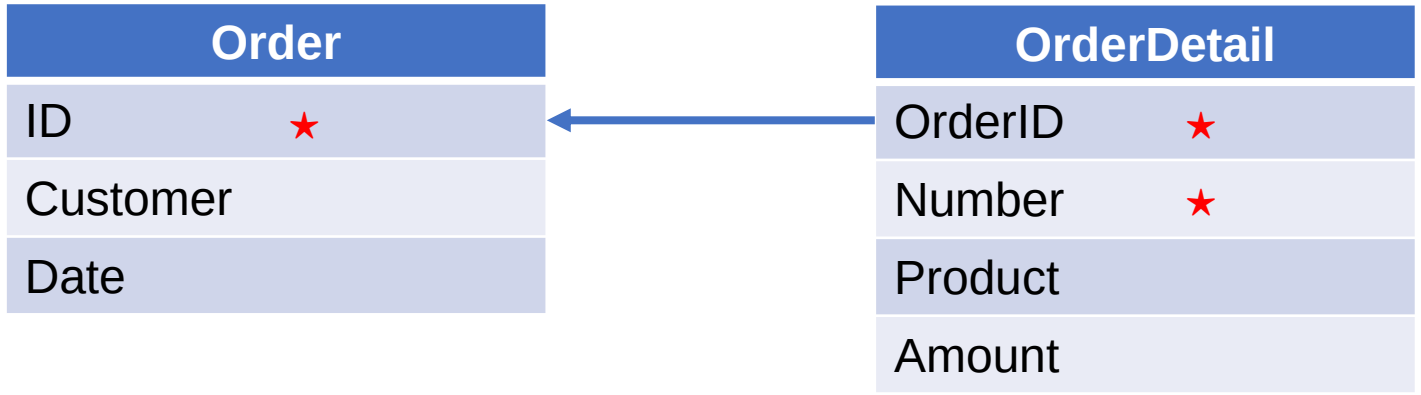
| | A | B |
|---|---|---|
| 1 | =file("p.csv").import@ct() | /read file |
| 2 | =P=A1.pselect@a(Original_Y==1),M=P.len() | /Select all the records whose real target is 1. Set the row number to P, and the quantity to M |
| 3 | =N=A1.len()-M | /For the records whose real target is not 1, set the quantity to N |
| 4 | =A1.(Predicted_Y).ranks@s() | /Use the ranks() function to calculate the rank of the prediction value. The @s option is used here. When the values are equal, the average value of the ranking will be used |
| 5 | =(A4(P).sum()-M*(1+M)/2)/(M*N) | /Calculate AUC value according to formula |

| A5 | Member |
|---|---|
| | 0.9055583178459794 |

# 10. Aggregation of sequences of sequences: sum sequence



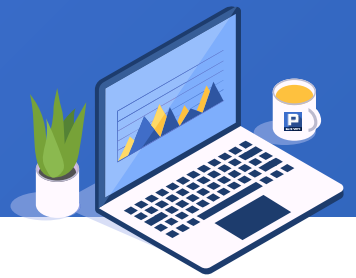
The order table and order detail table are the main sub table relationships. Each order has multiple details. As shown in the following figure:




The detail data of each order in the order detail table is variable in length. To find out the following table:

| 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 | | | | |

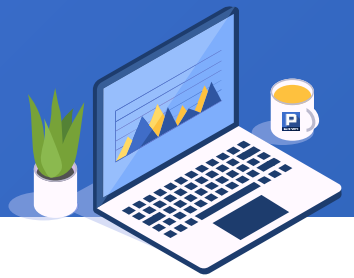
# ✦ 10. Aggregation of sequences of sequences: sum sequence



SPL is as follows, in which the A.conj() function is used to merge sequence members:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from OrderDetail left join Order on Order.ID=OrderDetail.OrderID") | /Import order detail table and order table, and join by order ID |
| 3 | =A2.group(ID) | /Group by order ID |
| 4 | =A3.max(~.count()).("Product"+string(~)+","+"Amount"+string(~)).concat@c() | /Find the group with the most members to determine the target table data structure |
| 5 | =create(ID,Customer,Date,${A4}) | /Create sequence table according to data structure determined by A4 |
| 6 | >A3.run(A5.record([ID,Customer,Date]\|~.([Product,Amount]).conj())) | /Loop the grouped data, and put members into a sequence in each group. Here, the conj function is used to get the sum sequence of each group's products and quantities. Finally, the generated records are inserted into the sequence table created by A5. |

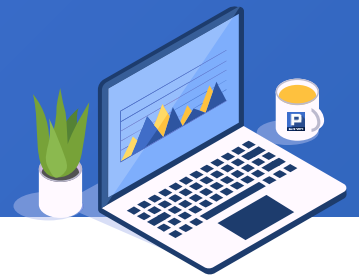
# ✦ 10. Aggregation of sequences of sequences: sum sequence



Here is the JSON data of the number of confirmed cases of new coronavirus around the world at a certain time. Calculate the number of confirmed cases in the world.

```
[
{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},
...]
```

# 10. Aggregation of sequences of sequences: sum sequence



SPL is as follows, in which A.conj@r() function is used to recursively merge sequence members:

| | A | B |
|---|---|---|
| 1 | =json(file("COVID-19.json").read()) | /Import JSON data file |
| 2 | =A1.field@r("Confirmed") | /Use the @r option of the A.field() function to recursively get all the confirmed fields |
| 3 | =A2.conj@r() | /Use the @r option of the A.conj() function to merge recursively |
| 4 | =A3.sum() | /Sum |

**A2**

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

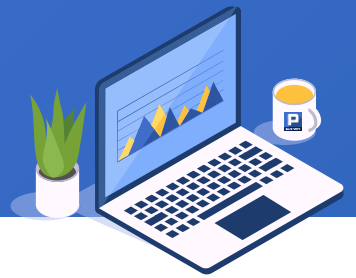
**A3**

| Member |
|---|
| 51986 |
| 3009 |
| 3791 |
| 1447 |
| 1206 |
| 1125 |
| … |

**A4**

| Member |
|---|
| 64438 |

# ✦ 11. Aggregation of sequences of sequences：union sequence and difference sequence

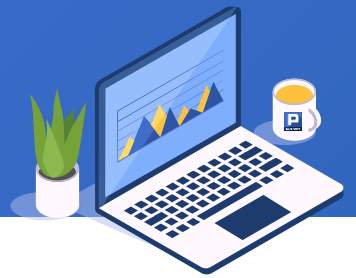


There are course table and course selection table. Query which courses are not selected by students. The course selection table can be multiple choices, separated by commas. Some data are as follows:

| 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 |
| … | … | … |

# 11. Aggregation of sequences of sequences: union sequence and difference sequence



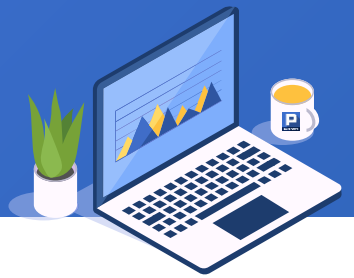
The A.union() function is used to find the union of sequence members, and the A.diff() function is used to find the difference of sequence members. The SPL is as follows:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Course") | /Read course table |
| 3 | =A1.query("select * from SelectCourse") | /Read SelectCourse table |
| 4 | =A3.union(COURSE.split@cp()) | /After the courses in the SelectCourse table are split by commas, union the course sequence using the union() function |
| 5 | =A2.(ID) | /IDs of all courses |
| 6 | =A2(A5.pos([A5,A4].diff())) | /Use the diff() function to find the difference sequence between the IDs of the course table and the SelectCourse table, that is, the course that no students selected. After positioning in A5, select from A2. |

A6

| ID | NAME | TEACHERID |
|---|---|---|
| 1 | Fundamentals of economic management | 21 |

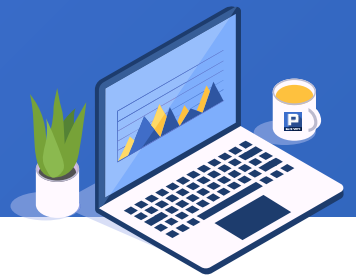
# 12. Aggregation of sequences of sequences：intersection sequence



Part of the data in the sales table is as follows. Calculate the names of customers whose monthly sales amounts are in the top 20 for every month in 2014.

| 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 |
| … | … | … | … | … |

# 12. Aggregation of sequences of sequences: intersection sequence

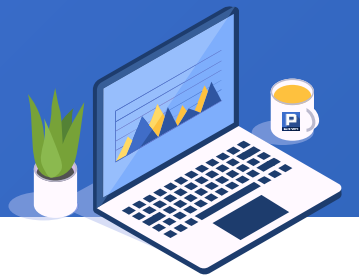


SPL is as follows, in which isect() function is used to find member intersection:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from sales") | /Connect the data source and read the sales table |
| 2 | =A1.select(year(OrderDate)==2014) | /Select 2014 data |
| 3 | =A2.group(month(OrderDate)) | /Use the group function to group 2014 data by month |
| 4 | =A3.(~.group(Customer)) | /Grouped members are grouped by customer |
| 5 | =A4.(~.top(-20;sum(Amount))) | /Loop monthly data to calculate the top 20 customers of monthly sales |
| 6 | =A5.(~.(Customer)) | /List the top 20 customers |
| 7 | =A6.isect() | /Use isect() function to find the intersection between each group |

| A7 | Member |
|---|---|
| | HANAR |
| | SAVEA |

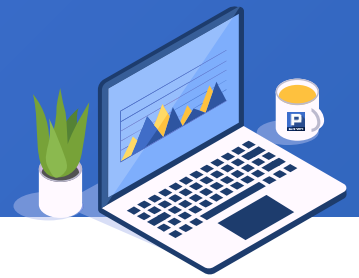
# 13. Aggregation of sequences of sequences: XOR sequence



The score sheet is saved in different files according to the semester. Query the ID of the students who have only entered the top ten of the total score once in the first and second semester.

| 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 |
| … | … | … | … |

# ✦ 13. Aggregation of sequences of sequences: XOR sequence



SPL is as follows, in which the A.xunion() function is used to select the records that are not repeated among the sequence members:

| | A | B |
|---|---|---|
| 1 | =file("Scores1.csv").import@ct() | /Import scores of first semester |
| 2 | =file("Scores2.csv").import@ct() | /Import scores of second semester |
| 3 | =A1.groups(STUDENTID;sum(SCORE):Score) | /Group and summarize student’ s total score of first semester |
| 4 | =A2.groups(STUDENTID;sum(SCORE):Score) | /Group and summarize student’ s total score of second semester |
| 5 | =A3.top(-10;Score).(STUDENTID) | /Select the top ten student ID of first semester's total score |
| 6 | =A4.top(-10;Score).(STUDENTID) | /Select the top ten student ID of second semester's total score |
| 7 | =[A5,A6].xunion() | /Using the xunion function, select the records that the student ID of the first and second semester does not repeat. |

| A5 | Member |
|---|---|
| | 2 |
| | 9 |
| | 4 |
| | 10 |
| | … |

| A6 | Member |
|---|---|
| | 12 |
| | 1 |
| | 8 |
| | 4 |
| | … |

| A7 | Member |
|---|---|
| | 2 |
| | 9 |
| | 10 |
| | 7 |
| | … |

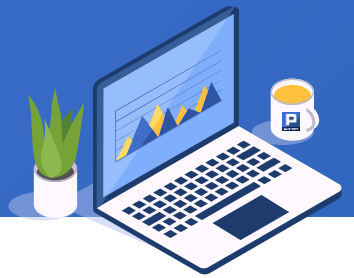
# CONTENTS



04

# Loop calculation

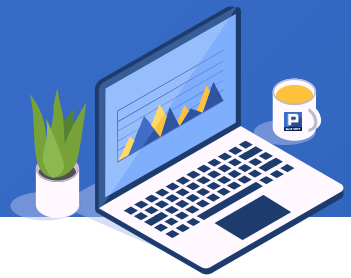
# ✦ 1. Loop function



Below is the JSON format sales data grouped by month and year. Calculate the total sales in 2016.

```
[
{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. Loop function



SPL is as follows, in which A.() is used for loop calculation:

| | A | B |
|---|---|---|
| 1 | =json(file("sales.json").read()) | /Import JSON data file |
| 2 | =A1.select(YEAR=2016) | /Select sales data of 2016 |
| 3 | =A2.field@r("SALES") | /Recursively get sales field value |
| 4 | =A3.(~.sum()).sum() | /Use A.() to calculate the total sales of each group in loop, and then calculate the total sales |

**A3**

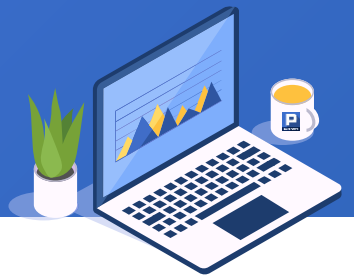
| Member |
|---|
| [3782,3773.38,1404, …] |
| [7208,8690.36,4566.05,…] |
| [5265.15,6130.35,3485.82…] |
| [2793.86,9264.86,2082.49,…] |

| Member |
|---|
| 3782 |
| 3773.38 |
| 1404 |
| … |

**A4**

| Member |
|---|
| 1252700.43 |

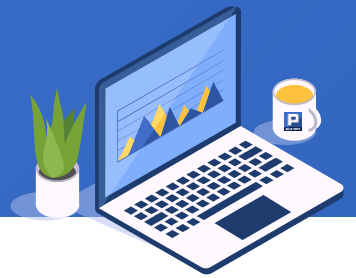
# 1. Loop function



Part of the data in the sales table is as follows. Another 5% performance award will be given to the top 10% salesmen in 2014.

| 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. Loop function

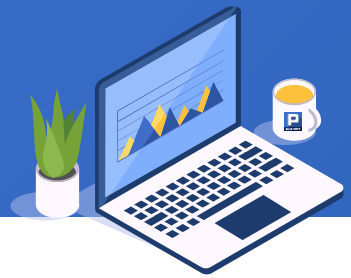


SPL is as follows, in which the A.run() function is used to loop the sequence members:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from sales") | /Connect the data source and read the sales table |
| 2 | =A1.select(year(OrderDate)==2014) | /Select 2014 data |
| 3 | =A2.groups(SellerId;sum(Amount):Amount) | /Use the groups function to summarize the total sales of the current year in groups by salesperson |
| 4 | =A3.sort@z(Amount).to(A3.len()*0.1) | /In descending order of the sales amount, take the top 10% |
| 5 | =A4.run(Amount*=1.05) | /Calculate sales amount of the top 10% in loop, give 5% reward and assign value |

| A5 | SellerId | Amount |
|---|---|---|
| | 4 | 150433.185 |
| | 3 | 127878.04 |
| | 1 | 102756.759 |
| | 8 | 87965.346 |

# ✦ 1. Loop function



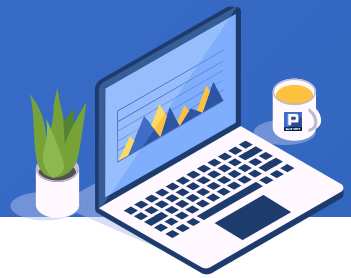
Some data of the payment table is as follows:

| 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 |

It is required to output the monthly payable amount according to the specified year (such as 2014). If there is no data of the current month, the payable amount of the current month is the value of the previous month:

| 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. Loop function



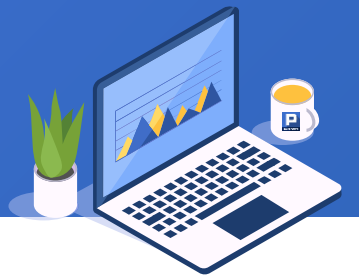
SPL is as follows. In the loop, the current member is calculated in loop:

| | A | B |
|---|---|---|
| 1 | $select * from Payment.txt  where year(due_date)=2014 | /Import data of payment table in 2014 |
| 2 | =create(name,${12.string()}) | /Create sequence table by target structure |
| 3 | =A1.group(customID).((m12=12.(null),~.(m12(month(due_date))=amount_payable), m12.(~=ifn(~,~[-1])),A2.record(name\|m12))) | /Group by customer ID, loop each group of data, and then loop the members, calculate the payable amount of each month, and add it to the sequence table created by A2 together with the customer name |

**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 |
| … | | | | | | | | | | | | |

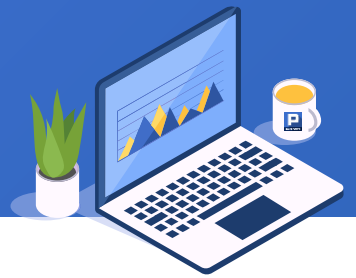
# ✦ 2. Symbol



Find the maximum number of consecutive days for the closing price of Shanghai stock index to increase in 2019.

| 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. Symbol

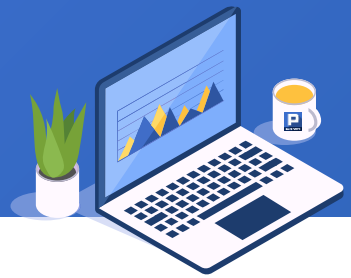


SPL is as follows, where ~ is used to represent the current member in the loop:

| | A | B |
|---|---|---|
| 1 | =file("000001.csv").import@ct() | /Import data file |
| 2 | =A1.select(year(Date)==2019).sort(Date) | /Select records of 2019 and sort by date |
| 3 | =last=0,count=0,A2.(Close).((if(~>last,count+=1,count=0),last=~,count)).max() | /Loop the closing price, compare the closing price of each day with the closing price of the previous day. If the closing price of the current day is higher, add 1 to the count, and finally select the maximum count value. |

| A3 | Value |
|---|---|
| | 6 |

# ✦ 2. Symbol



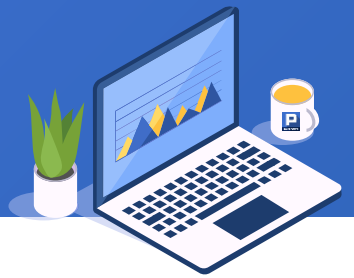
One cock is worth five, one hen is worth three, and three chicks are worth one. If you bought a total of 100 of mixed cocks, hens and chicks for 100 yuan, how many  are the cocks, hens and chicks respectively?

SPL is as follows, where ~ is used to represent the current member in the loop:

| | A | B |
|---|---|---|
| 1 | =to(100/5) | /Quantity of cocks that may be purchased |
| 2 | =to(100/3) | /Quantity of hens that may be purchased |
| 3 | =33.(~*3) | /Quantity of chicks that may be purchased |
| 4 | =create(Cock,Hen,Chick) | /Create a sequence table to store the number of cock, hen and chick |
| 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.~))))) | /Loop cock, hen, chick respectively, when the condition is met, insert the result to the sequence table created in A4. The ~ symbol is used to represent the current member of the sequence loop |

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

## ✦ 2. Symbol



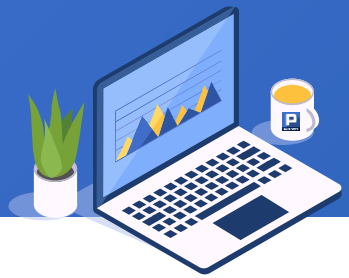
Here are two texts. Find the string of file 1 in file 2. We want to return it in the following form:

| file1 |
| --- |
| like parks |
| went out |
| go out |

| file2 |
| --- |
| I like to go out because I like parks. |
| Ben does not go out much. |
| Shelly went out often but does not like parks. |
| Harry does not go out neither does he like parks. |

| 输出 |
| --- |
| Q1. like parks |
| I |
| Shelly |
| Harry |
| Q2. went out |
| Shelly |
| Q3. go out |
| I |
| Ben |
| Harry |

# ✦ 2. Symbol

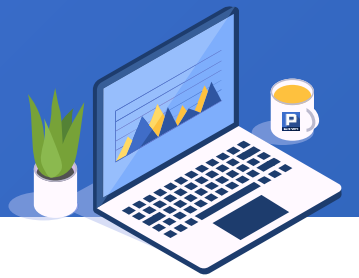


SPL is as follows. In the loop, the # symbol is used to represent the current sequence number:

| | A | B |
|---|---|---|
| 1 | =file("file1.txt").read@n() | /Read file1 |
| 2 | =file("file2.txt").read@n() | /Read file2 |
| 3 | =A1.conj(("Q"+string(#)+". "+~)\|A2.select(pos(~,A1.~)).(~.words()(1))) | /Loop through the string in file 1, find it in file 2, and take the first word. In A2.select, ~ represents the current member of A2, and A1.~ represents the current member of A1. Before each group of searching result, spell Q+A1 sequence number+A1 current member, where A1 sequence number is got from #. |

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

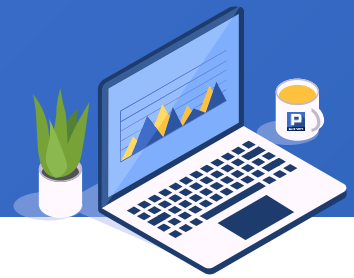
# ✦ 3. Positioning calculation



List the 20 day closing average price of China Merchants Bank (600036) from January 1 to 10, 2020.

| 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. Positioning calculation



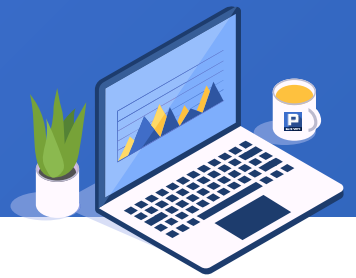
SPL is as follows, where the A.calc() function is used to calculate and return at the specified position, and the [a:b] is used to access members in loop:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect data source |
| 2 | =A1.query("select Date, Close from Stock where Code='600036' order by Date") | /Select data of China Merchants Bank and sort by date |
| 3 | =A2.pselect@a(Date>=date("2020/01/01") && Date<=date("2020/01/10")) | /Use the pselect() function to get the row number of the record from January 1 to 10, 2020 |
| 4 | =A2(A3).derive(A2.calc(A3(#),avg(Close[-19:0])):ma20) | /The calc() function is used to calculate in loop the 20-day average of the data of the first ten days and return it. The close[- 19:0] is used to represent the closing price from today to the past 19 days. |

| 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 |

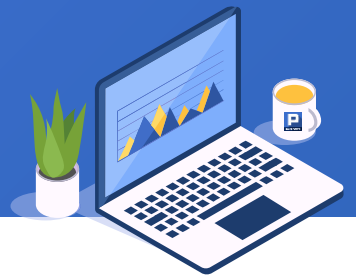
# 4. Iteration calculation



Part of the data in the sales table is as follows. How many days does it take to reach 20 orders per month in 2014?

| 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. Iteration calculation

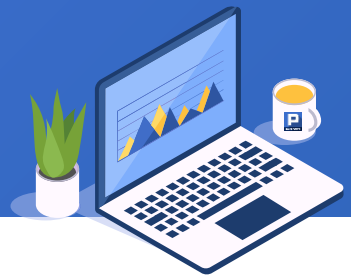


SPL is as follows, in which seq() function is used to generate sequence number:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from sales") | /Connect the data source and read the sales table |
| 2 | =A1.select(year(OrderDate)==2014) | /Select 2014 data |
| 3 | =A2.sort(OrderDate) | /Sort by order date |
| 4 | =A3.select(seq(month(OrderDate))==20) | /Use the seq() function to generate the order sequence number of each month and select the record with the sequence number of 20 for each month |
| 5 | =A4.new(month(OrderDate):Month,day(OrderDate):Day) | /List the number of days needed to reach 20 sales per month |

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

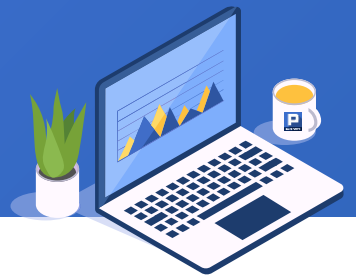
# 4. Iteration calculation



Query the ranking of employee income by department. Some data in the employee table is as follows:

| 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. Iteration calculation

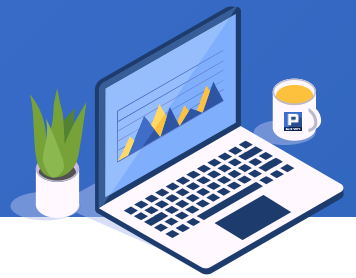


SPL is as follows, in which rank() function is used to uniformly number members with the same field value:

| | A | B |
|---|---|---|
| 1 | =connect("db") .query("select * from Employee order by DEPT, SALARY DESC") | /Connect data source, read employee table and sort by department and salary |
| 2 | =A1.derive(rank(SALARY;DEPT):DeptRank) | /Use rank() function to number members for ordered department and salary data, calculate the ranking of each department |

| 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 |
| | … | … | … | … | … |

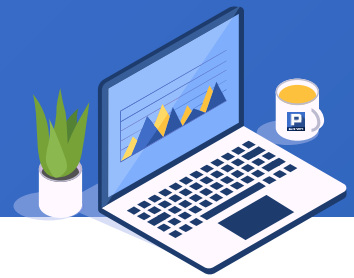
# 4. Iteration calculation



Take the score table as an example to find the ranking of student with ID 8 in each subject in class one.

| 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. Iteration calculation

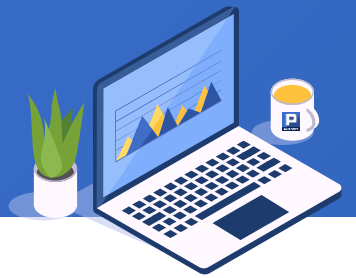


The ranki() function is used to uniformly number members with the same field value. The difference between ranki and rank functions is that the number returned is consecutive, similar to DENSE_RANK. SPL is as follows:

| | A | B |
|---|---|---|
| 1 | =connect("db") .query("select * from SCORES where CLASS='Class one' order by SUBJECT, SCORE DESC") | /Connect data source, read scores table and sort by subject and score |
| 2 | =A1.derive(ranki(SCORE;SUBJECT):Rank) | /use ranki () function to number for orderly subjects and scores, calculate the ranking of each subject |
| 3 | =A2.select(STUDENTID==8) | /Select student information with student ID 8 |
| 4 | =create(${A3.(SUBJECT).concat@c()}).record(A3.(Rank)) | /Using the result of A3, sort out the ranking of each subject |

| A4 | English | Math | PE |
|---|---|---|---|
| | 10 | 4 | 14 |

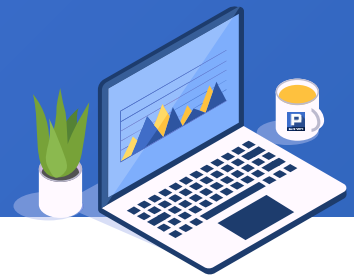
# ✦ 4. Iteration calculation



Calculate the total annual turnover of Shanghai stock index on each trading day in 2019.

| 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. Iteration calculation

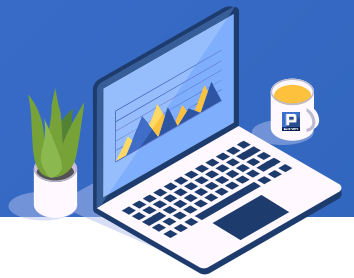


SPL is as follows, in which the cum() function is used to calculate the cumulative transaction amount:

| | A | B |
|---|---|---|
| 1 | =file("000001.csv").import@ct() | /Import data file |
| 2 | =A1.select(year(Date)==2019).sort(Date) | /Select records of 2019 and sort by date |
| 3 | =A2.derive(cum('Amount'):CUM) | /Use the cum() function to calculate the cumulative transaction amount |

| 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 |
| | … | … | … | … | … |

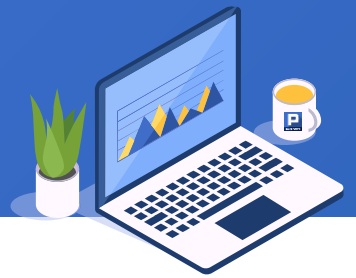
# 4. Iteration calculation



At the beginning of 2014, the sales target for the first quarter was set at 150000 yuan. Query on which day the sales target was achieved. Some data in the sales table is as follows:

| 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. Iteration calculation



SPL is as follows, in which the A.iterate() function is used to iteratively calculate the sequence members:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from sales") | /Connect the data source and read the sales table |
| 2 | =A1.select(year(OrderDate)==2014) | /Select 2014 data |
| 3 | =A2.iterate((@+=Amount,<br>~~=OrderDate),0,@>150000) | /Iterate using the iterate() function with an initial value of 0. Add the sales amount to the current grid until it is over 150000. Function returns the order date. |

| A3 | Member |
|---|---|
| | 2014/3/25 |

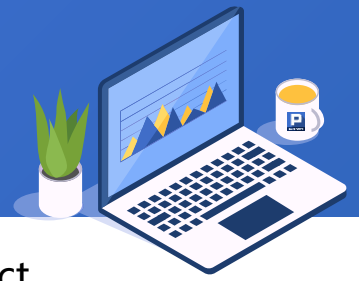
# CONTENTS



05

## Positioning

# ✦ 1. Locate a member's position in the sequence



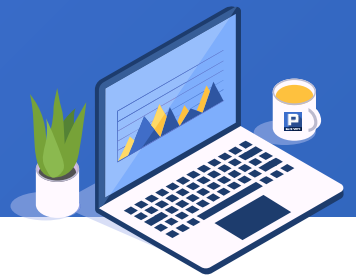
In the following teachers table, the first column is the name of the teacher, the second column is the subject, followed by the course code (null is empty).

| 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 | … |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … | … |

List the appropriate teachers for each course according to the teachers table and the courses table on the right.

| 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. Locate a member's position in the sequence

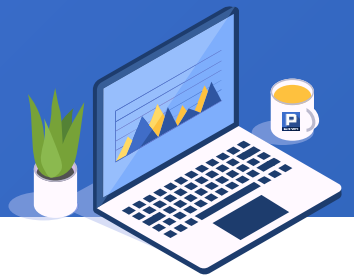


SPL is as follows, in which the pos function is used to obtain the member’ s position.

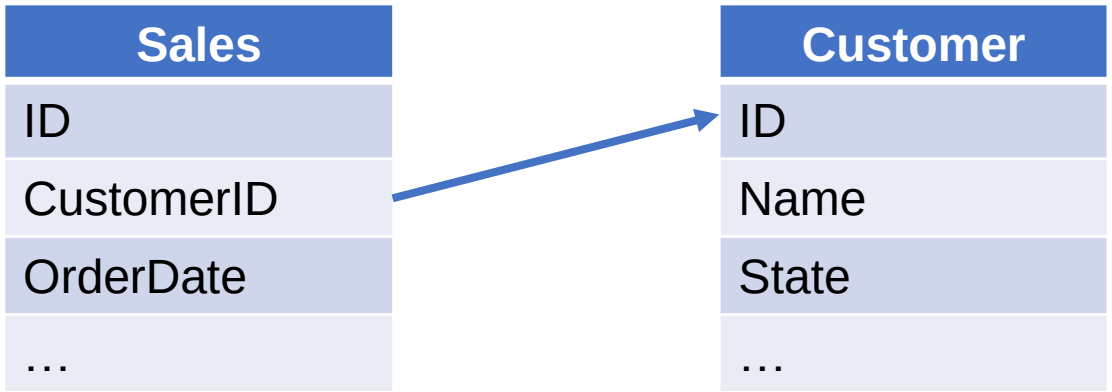
| | A | B |
|---|---|---|
| 1 | =file("Teachers.txt").import() | /Import data file |
| 2 | =A1.new(#1:professor,~.array().to(3,A1.fno()).select(~!=null):codeArray) | /Generate a new sequence table with two columns. The first column is the teacher name and the second column is the course list. |
| 3 | =file("Courses.txt").import@t().conj(~.array()) | /List all courses according to the courses table and merge them into one sequence |
| 4 | =A3.(A2.select(codeArray.pos(A3.~)).(professor)) | /Loop the courses table, use pos function to find the course in the teachers table and select the appropriate teachers for each class. |
| 5 | =create(Monday,Tuesday,Wednesday,Thursday,Friday).record(A4.(~.concat@c())) | /Create a curriculum from Monday to Friday and fill in the teachers accordingly. |

| 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… |
| | … | … | … | … | … |

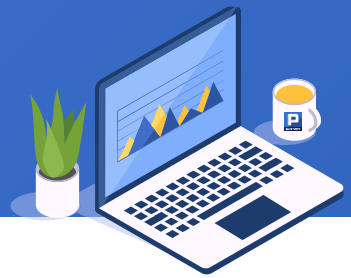
# ✦ 1. Locate a member's position in the sequence



There are sales table and customer table. Query customers without sales records in 2014.



# ✦ 1. Locate a member's position in the sequence

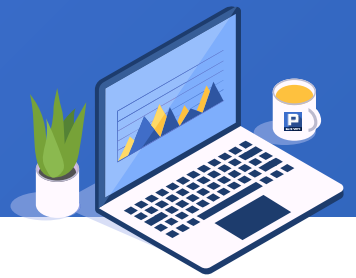


SPL is as follows, in which A.pos() function is used to locate the position of members in the sequence:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Sales where year(OrderDate)=2014") | /Query 2014 sales records |
| 3 | =A1.query("select * from Customer") | /Query customer table |
| 4 | =A3.(ID).sort() | /Select customer ID from customer table |
| 5 | =A2.align(A4.len(), A4.pos@b(CustomerID)) | /The sales table is grouped in alignment to the customer ID, and the pos function is used to locate the customer ID. Because the customer ID is in order, the @b option is used here for binary search, which can locate more quickly. |
| 6 | =A3(A5.pos@a(null)) | /Use the @a option of the pos() function to select all customer information without sales records (the value is null), otherwise only one will be selected. |

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

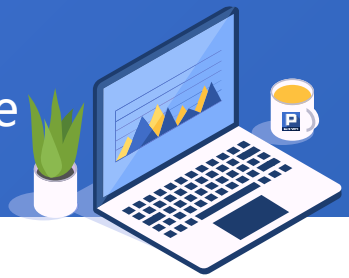
## ✦ 2. Take the row number of the record corresponding to the maximum / minimum value



On the day when the closing price of Shanghai stock index is at its highest point, find the increase rate compared to the previous day.

| 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. Take the row number of the record corresponding to the maximum / minimum value



We need to know the record row number of the stock market's highest point, and then compare it with the previous trading day to get the result.

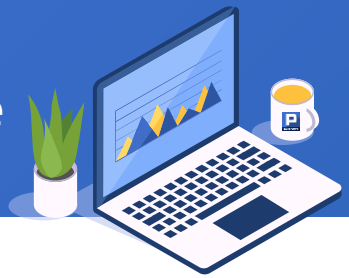
SPL is as follows, where pmax function is used to get the row number of the maximum value:

| | A | B |
|---|---|---|
| 1 | =file("000001.csv").import@ct() | /Import data file |
| 2 | =A1.sort(Date) | /Sort by date |
| 3 | =A2.pmax(Close) | /Take the row number of the highest point of the stock market |
| 4 | =A2(A3).Close/A2.m(A3-1).Close-1 | /Use the closing price of the day and the closing price of the previous day to calculate the increase rate |

Similarly, pmin function can be used to get the row number of the minimum value:

| | A | B |
|---|---|---|
| 3 | =A3.pmin(Close) | /Take the row number of the lowest point of the stock market |

## 2. Take the row number of the record corresponding to the maximum / minimum value



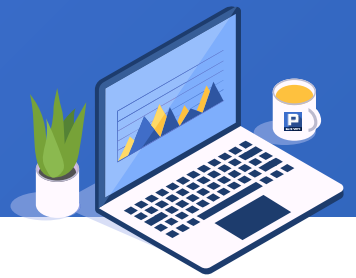
The records of maximum value are not necessarily unique. If you want to return all row numbers, you can use pmax@a option:

| | A | B |
|---|---|---|
| **3** | =A2.pmax@a(Close) | /Take the row numbers of all the records of the highest point of the stock market |
| **4** | =A3.(A2(A3.~).Close/A2.m(A3.~-1).Close-1) | /Use the closing price of the day and the closing price of the previous day to calculate the increase rate in loop |

If you want to locate from back to front, you can use the pmax@z option:

| | A | B |
|---|---|---|
| **3** | =A2.pmax@z(Close) | /Take the row numbers of the records of the highest point of the stock market from back to front |

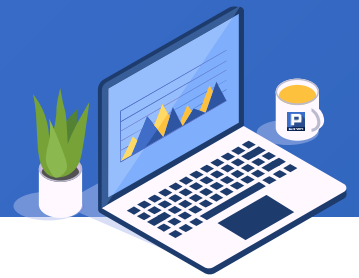
# ✦ 3. Get the member sequence number that meets the condition



For the trading days when the closing price of Shanghai stock index rises by more than 3% in 2019, find the trading amount increase.

| 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. Get the member sequence number that meets the condition



We need to find the row number where the closing price of the stock market has risen by more than 3%, and then compare it with the previous trading day to get the result. SPL is as follows, in which pselect() function is used to locate the row number of the member:

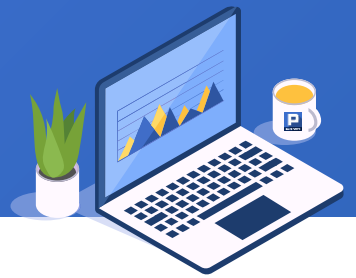
| | A | B |
|---|---|---|
| 1 | =file("000001.csv").import@ct() | /Import data file |
| 2 | =A1.select(year(Date)==2019).sort(Date) | /Select the stock market records for 2019 |
| 3 | =A2.pselect@a(Close/Close[-1]>1.03) | /Take the row number of the record where the closing price of the stock market has increased by more than 3%, and the @a option will return all the row numbers that meet the conditions |
| 4 | =A3.new(A2(~).Date:Date, A2(~).Amount/A2(~-1).Amount:'Amount increase') | /Use daily trading amount and trading amount of the previous day to calculate the increase in loop |

| A3 | Member |
|---|---|
| | 161 |
| | 187 |
| | 211 |

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

We can see that the trading volume of the three days when the closing price rose by more than 3% was significantly higher than the previous day.

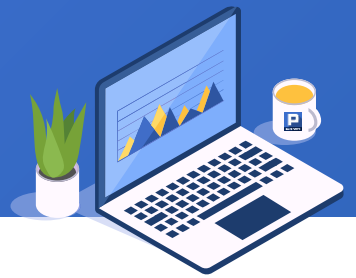
# ✦ 4. Segment sequence number of member in sequence



The following is an employee table. According to the salary, the employees are divided into below 8000, 8000-12000 and above 12000, and the number of each group is counted.

| 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. Segment sequence number of member in sequence



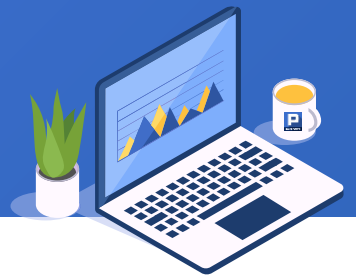
SPL is as follows, in which the overall positioning function pseg(x) is used:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from EMPLOYEE") | /Query employee table |
| 3 | [0,8000,12000] | /Define salary interval |
| 4 | =A2.align@a(A3.len(),A3.pseg(SALARY)) | /Use pseg function to get the salary interval |
| 5 | =A4.new(A3 (#):SALARY,~.count():COUNT) | /Count the number of each group |

A5

| SALARY | COUNT |
|---|---|
| 0 | 308 |
| 8000 | 153 |
| 12000 | 39 |

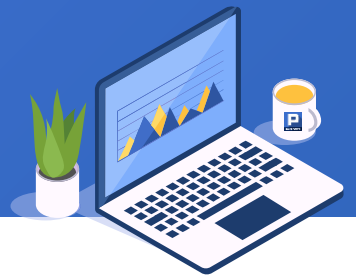
# ✦ 4. Segment sequence number of member in sequence



The following is an employee table. Employees are divided into less than 10 years, 10-20 years and more than 20 years according to the date of entry, and the average salary of each group is calculated.

| 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. Segment sequence number of member in sequence



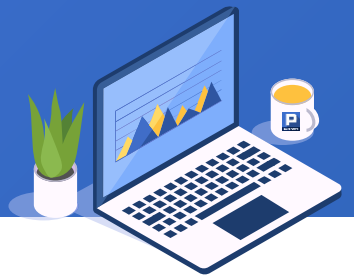
SPL is as follows, in which the overall positioning function pseg(x,y) is used:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from EMPLOYEE") | /Query employee table |
| 3 | [0,10,20] | /Define interval |
| 4 | =A2.align@a(A3.len(),A3.pseg(year(now())-~,year(HIREDATE))) | /Use pseg function to get the interval each hire date belongs to |
| 5 | =A4.new(A3(#):EntryYears,~.avg(SALARY):AvgSalary) | /Calculate the average salary of each group |

**A5**

| EntryYears | AvgSalary |
|---|---|
| 0 | 6807.69 |
| 10 | 7417.78 |
| 20 | 7324.32 |

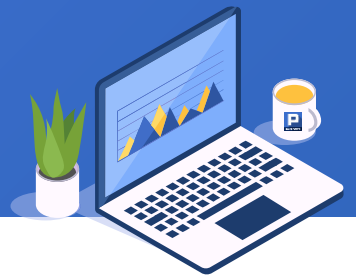
# ✦ 5. Get the original sequence number of a sorted member



The following is an employee table. Find the entry order of the three oldest employees.

| 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. Get the original sequence number of a sorted member

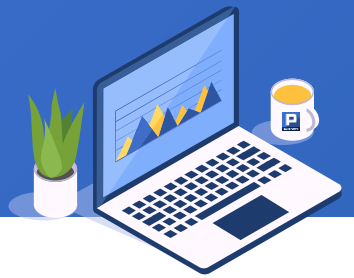


A.psort() function is used here to get the original sequence number of a sorted member. It is worth noting that the psort() function does not change the original sequence order. SPL is as follows:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from EMPLOYEE order by HIREDATE") | /Query employee table and sort by hire date |
| 3 | =A2.psort(BIRTHDAY) | /Get the sequence number of employee's birthday |
| 4 | =A2(A3.to(3).sort()) | /Select the three oldest employees in the employee table |

| 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 |

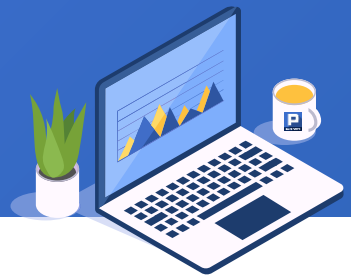
# ✦ 6. Overall positioning of sequence



The following is the posting records. Group posts by labels and count the frequency of each label.

| 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. Overall positioning of sequence

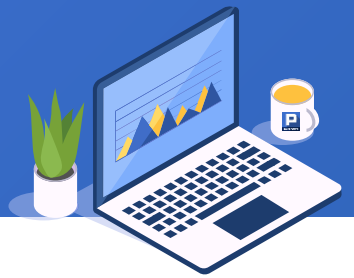


SPL is as follows, and A.pos() function is used for overall positioning:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from PostRecord") | /Query PostRecord |
| 3 | =A2.conj(Label.split(",")).id() | /Separate the labels by commas and merge them into a sequence to get all the labels without duplicate values. |
| 4 | =A2.align@ar(A3.len(),A3.pos(Label.split(","))) | /Use the pos() function to overall locate the position of the post’ s labels in all labels. Then use the @r option of the align() function to group by positioning. |
| 5 | =A4.new(A3(#):Label,~.count():Count).sort@z(Count) | /Count the number of posts per label, in descending order |

| A5 | Label | Count |
|---|---|---|
| | SPL | 7 |
| | SQL | 6 |
| | Basics | 5 |
| | … | … |

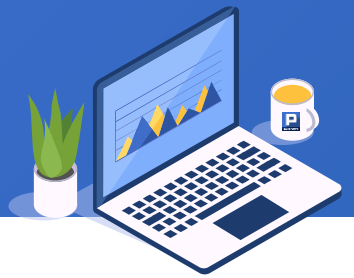
# ✦ 7. Determine whether it is a sequence member



The following is the official language table of all countries in the world. Check the countries whose official languages include Chinese and English.

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

# ✦ 7. Determine whether it is a sequence member

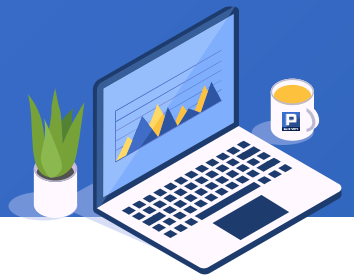


SPL is as follows. The A.contain() function is used to determine whether it is a sequence member:

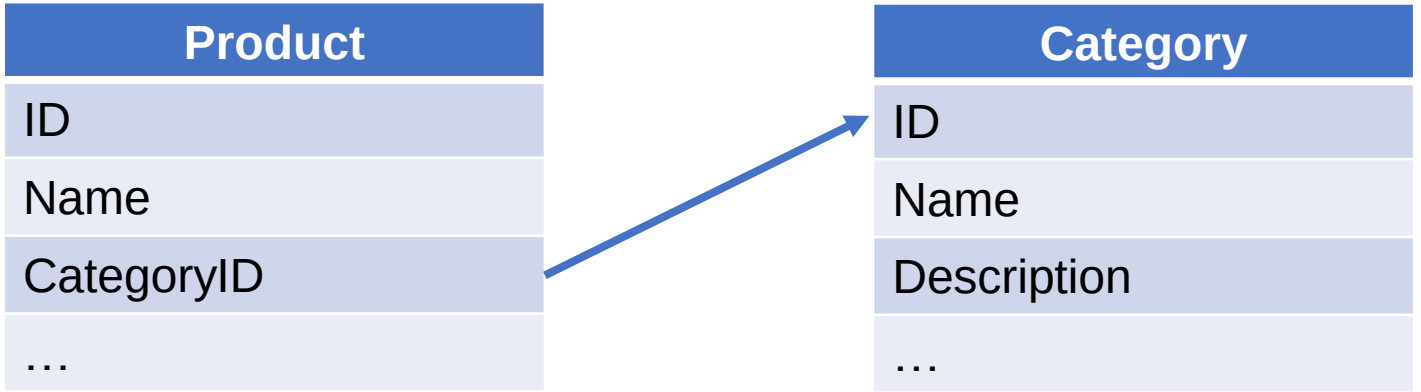
| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Language") | /Query official language table |
| 3 | =A2.group(Country) | /Group by country |
| 4 | =A3.select(~.(Language).contain("Chinese","English")) | /Use the contain() function to determine whether the current country's languages include Chinese and English. |
| 5 | =A4.(Country) | /Get country list |

| A5 | Member |
|---|---|
| | Singapore |

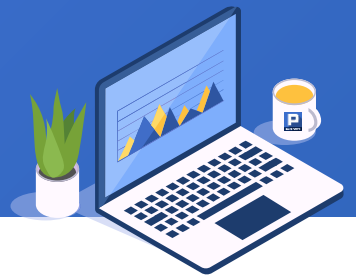
# 8. Find row number of primary key



There are product table and category table. Query the product records whose product categories are not registered in the category table. As the following figure:



# 8. Find row number of primary key

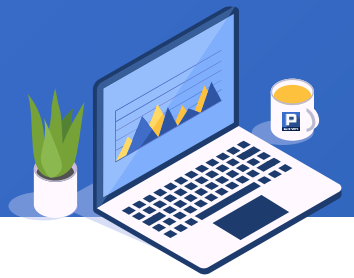


SPL is as follows, and the A.pfind() function is used to find the row number of the primary key:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Category").keys(ID) | /Query category table, and set ID as primary key |
| 3 | =A1.query("select * from Product") | /Query product table |
| 4 | =A3.select(A2.pfind(CategoryID)==0) | /Use the pfind function to find the row number with the primary key equal to the category ID in the category table, and return 0 to indicate that it does not exist. In the product table, select the record whose category ID does not exist. |

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

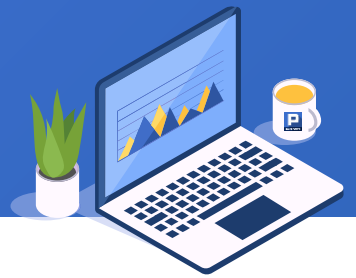
# ✦ 9. Get the row numbers corresponding to TopN records



For the three days with the highest closing price of Shanghai stock index in 2019, find the increase rate of the transaction amount compared with the previous day.

| 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. Get the row numbers corresponding to TopN records



We need to know the row number of the highest three-day closing price records of the stock market, and then compare it with the previous trading day to get the result. SPL is as follows, where the ptop() function is used to get the row number of the highest three days:

| | A | B |
|---|---|---|
| 1 | =file("000001.csv").import@ct() | /Import data file |
| 2 | =A1.select(year(Date)==2019) | /Select records of 2019 |
| 3 | =A2.ptop(-3, Close) | /Use ptop function to get the row number of the highest three-day closing price records.-3 indicates to get top 3 from large to small. If it is a positive integer, it means getting from small to large. |
| 4 | =A3.run(~=A2(~).Amount/A2(~+1).Amount-1) | /Use daily trading amount and trading amount of the previous day to calculate the increase in loop. |

| A3 | VALUE |
|---|---|
| | 154 |
| | 156 |
| | 157 |

| A4 | VALUE |
|---|---|
| | -0.0278 |
| | -0.0139 |
| | 0.0112 |

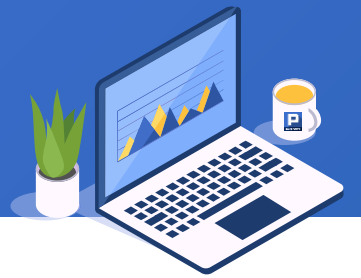
# CONTENTS



06

## Select

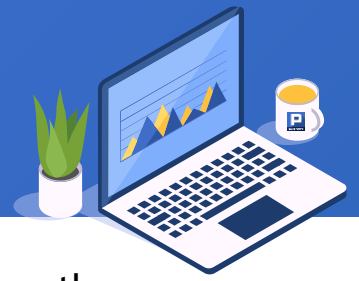
# 1. Get the record corresponding to the minimum value



Take the score table as an example to find the student ID with the lowest score in mathematics in class one.

| 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. Get the record corresponding to the minimum value



SPL is as follows, in which minp() function is used to take the record where the minimum value is, and then the student ID is taken from it:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Scores where SUBJECT='Math' and CLASS='Class one'") | /Query the math scores of class one |
| 3 | =A2.minp(SCORE) | /Use minp function to get the record with lowest score |
| 4 | =A3.STUDENTID | /Take student ID from record |

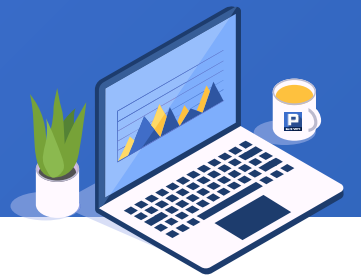
The record of the minimum value is not necessarily unique. If you want to return all records, you can use the minp@a option:

| | A | B |
|---|---|---|
| 3 | =A2.minp@a(SCORE) | /Get all the records with lowest score |
| 4 | =A3.(STUDENTID) | /Take student ID sequence from multiple records |

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

| A4 | STUDENTID |
|---|---|
| | 5 |
| | 14 |

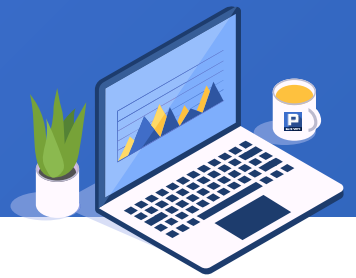
# 2. Get the record corresponding to the maximum value



Take the medal table of the Olympic Games as an example to seek the national information of the longest number of Olympic Games with the total result ranking first in a row.

| 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. Get the record corresponding to the maximum value

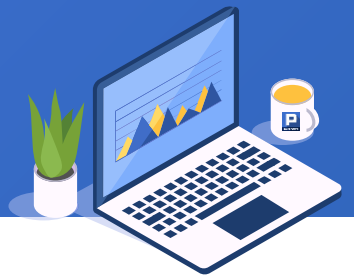


SPL is as follows, in which maxp() function is used to take the record corresponding to the maximum value:

| | A | B |
|---|---|---|
| 1 | =file("Olympic.csv").import@cqt() | /Import the medal table of Olympic Games |
| 2 | =A1.sort@z(Game, 1000000*Gold+1000*Silver+Copper) | /Sort by game number and total score in descending order |
| 3 | =A2.group@o1(Game) | /Take one from each game, i.e. the first because the data is in order |
| 4 | =A3.group@o(Nation) | /Group neighboring countries in the same order |
| 5 | =A4.maxp(~.len()) | /Take the longest group |

| 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. Select member that meets the condition



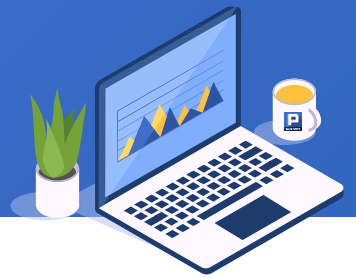
The world urban population table is as follows:

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

List the names and population of cities with a population of more than 2 million in Europe and Africa in columns (each column is sorted from most to least). The expected result is as follows:

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

# ✦ 3. Select member that meets the condition

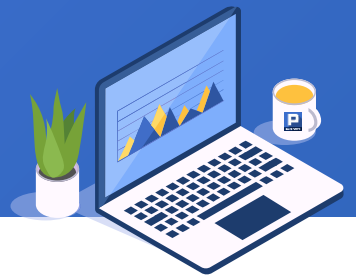


SPL is as follows, in which A.select() function is used to select qualified members:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from World where Continent in('Europe', 'Africa') and Population >= 2000000") | /Connect to the database and retrieve records of more than 2 million people in Europe and Africa |
| 2 | =A1.select(Continent:"Europe") | /Using the select() function to retrieve European data |
| 3 | =A1.select(Continent:"Africa") | /Using the select() function to retrieve African data |
| 4 | =create('Europe City',Population,'Africa City', Population) | /Create an empty sequence table according to target structure |
| 5 | =A4.paste(A2.(City),A2.(Population),A3.(City),A3.(Population)) | /Paste the sequence of values into the corresponding column using the paste() function |

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

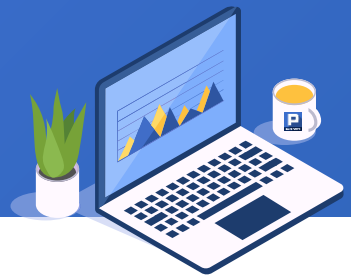
## 4. Return the corresponding member in the sequence according to the section number



Take the score sheet as an example to count the number of excellent, pass and fail in subject English.

| 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. Return the corresponding member in the sequence according to the section number

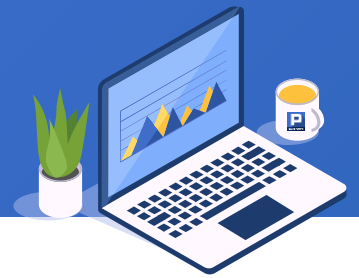


SPL is as follows, in which the A.segp() function is used to get the members in the sequence corresponding to the section number:

| | A | B |
|---|---|---|
| 1 | =connect("db").query("select * from Scores where SUBJECT='English'") | /Connect to the database and query English scores |
| 2 | =create(Assessment,Score).record(["fail",0,"pass",60,"excellent",90]) | /Create score and assessment comparison table |
| 3 | =A1.derive(A2.segp(Score,SCORE).Assessment:Assessment) | /Using segp function to return assessment corresponding to score |
| 4 | =A3.groups(Assessment;count(1):Count) | /Group and count according to assessment |

| A4 | Assessment | Count |
|---|---|---|
| | excellent | 6 |
| | fail | 4 |
| | pass | 18 |

# ✦ 5. Sort

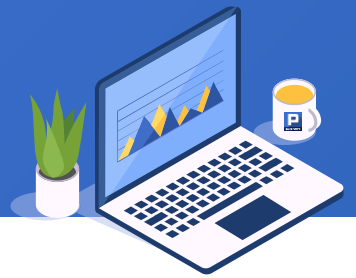


The economic indicators of each country in the world in a certain year are as follows, and the comparison of economic indicators between China and the United States with a large gap is required.

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

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

# ✦ 5. Sort

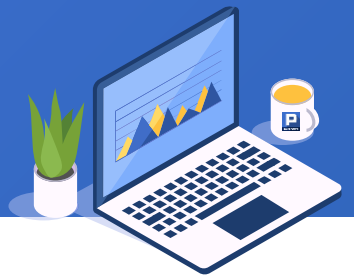


SPL is as follows, in which the A.sort() function is used to sort the data:

| | A | B |
|---|---|---|
| 1 | =file("Indicators.xlsx").xlsopen() | /Open excel file |
| 2 | =A1.xlsimport@t(Indicator,Last).select(Indicator!=null) | /Import the first sheet (U.S.), select the record whose indicator is not null |
| 3 | =A1.xlsimport@t(Indicator,Last;"China").select(Indicator!=null) | /Import China sheet，select the record whose indicator is not null |
| 4 | =A2.join(Indicator,A3:Indicator,Last:'China').rename(Last:'United States') | /Join US data according to indicator name, add China indicator field |
| 5 | =A4.sort@z(abs('United States'-'China')) | /Sort in descending order by the absolute value of the difference between China and US indicators. The sort function is used for sorting, and the @z option indicates descending order. |

| 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 |
| | … | … | … |

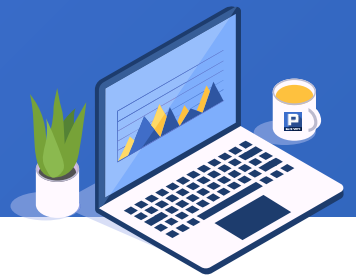
# ✦ 6. Get TopN records



Take the score sheet as an example to find the top two students in each subject in each class.

| 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. Get TopN records

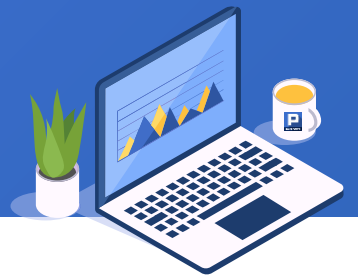


SPL is as follows, in which the A.top() function is used to get the first n / last n members:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Scores") | /Query student scores |
| 3 | =A2.group(CLASS,SUBJECT;~.top(-2;SCORE):TOP2) | /Group by class and subject and take out the top two scores of each group |
| 4 | =A3. conj(TOP2) | /Combine the top two records of all classes and subjects |

| 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. Find the record where the primary key is located



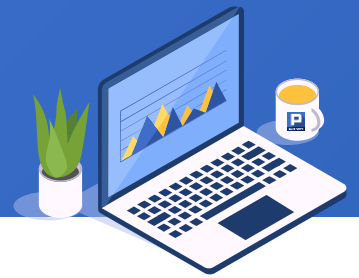
There are course table and course selection table. Check the courses selected by students:

| 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. Find the record where the primary key is located



SPL is as follows, where find() function is used to find the record where the primary key is located:

| | A | B |
|---|---|---|
| 1 | =connect("db") | /Connect to database |
| 2 | =A1.query("select * from Course").keys(ID) | /Read course table, and set primary key ID |
| 3 | =A1.query("select * from SelectCourse") | /Read SelectCourse table |
| 4 | =A3.run(COURSE=COURSE.split@cp()) | /Split the courses in the SelectCourse table by commas and assign them to the course field |
| 5 | =A4.max(COURSE.len()) | /Find the maximum number of selected courses |
| 6 | =create(ID,NAME, ${A5.("COURSE"+string(~)).concat@c()}) | /Create an empty table, and the course column is created according to the maximum number |
| 7 | >A4.run(A6.record([ID,STUDENT_NAME]\|COURSE.(A2.find(~).Name))) | /Loop the SelectCourse table, use the find() function to find the record where the course primary key is located, and find the course name. And insert the combined records into the A6 sequence table. |

| 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))) | /In A7, row() function can also be used to find the row number in the sequence. A.find(~) is equivalent to ~.row(A) |

# THANKS