SPL Base

Handling of strings and date time

CONTENTS

01 Strings

1. Concatenation
2. Split
3. Parse
4. Complex processing

02 Date time

1. Calculate the single value of date and time
2. Calculate date time sequence

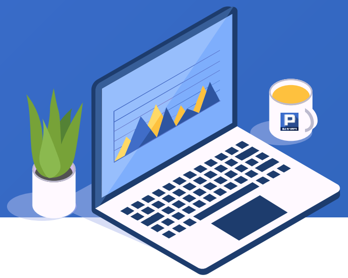
CONTENTS

1. Concatenation
2. Split
3. Parse
4. Complex processing

01

Strings

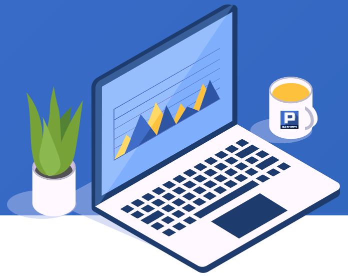
1. Concatenation：Concatenate two strings



Check the full name and salary of R&D employees in New York. The employee table is as follows:

ID	NAME	SURNAME	STATE	DEPT	SALARY
1	Rebecca	Moore	California	R&D	7000
2	Ashley	Wilson	New York	Finance	11000
3	Rachel	Johnson	New Mexico	Sales	9000
4	Emily	Smith	Texas	HR	7000
5	Ashley	Smith	Texas	R&D	16000
…	…	…	…	…	…

✦ 1. Concatenation：Concatenate two strings

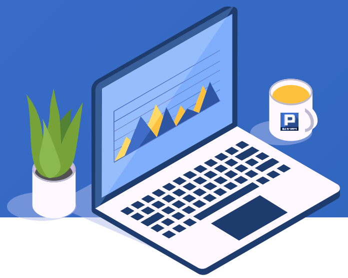


SPL is as follows, in which the "+" is used to concatenate strings:

	A	B
1	=connect("db")	/Connect to data source
2	=A1.query("select * from Employee")	/Import employee table
3	=A2.select(STATE=="New York"&&DEPT=="R&D")	/Select employee records of R&D department in New York
4	=A3.new(NAME+" "+SURNAME:FULLNAME, SALARY)	/Use sign + to concatenate strings to form full name

A4	FULLNAME	SALARY
	Matthew Johnson	6000
	Lauren Thomas	12000
	Brooke Williams	12000

✦ 1. Concatenation：Concatenate string and any type



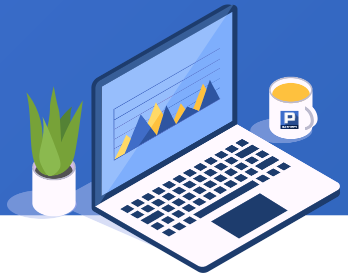
Here are two texts. Look for the string of text 1 in text 2, and we want to return the following form:

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.

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

✦ 1. Concatenation：Concatenate string and any type



SPL is as follows, where the symbol '/' is used to concatenate string and other type of value:

	A	B
1	=file("file1.txt").read@n()	/Read text1
2	=file("file2.txt").read@n()	/Read text2
3	=A1.conj(("Q"/#+". "+~)|A2.select(pos(~,A1.~)).(~.words()(1)))	/Loop the strings in text 1, find it in text 2, and take the first word. Each group of search result is spelled with Q and the ordinal numbers of A1 and the current member, where the ordinal number is an integer type, and symbol '/' is used to concatenate.

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

✦ 1. Concatenation：Concatenate sequence members

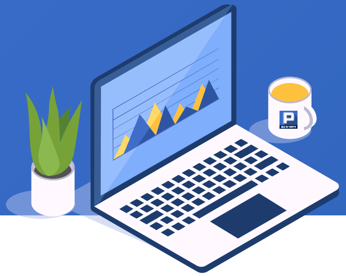
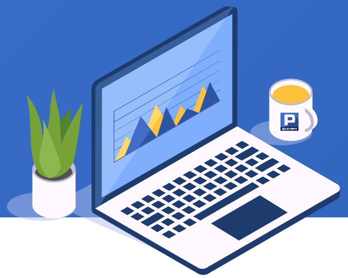


Table A and table B have the same structure. Use table B to update table A. When the primary key of table B exists in table A, update the record; otherwise, add a new row.

Table A		
ID	Amount	…
1	3063.0	…
2	3868.6	…
4	2713.5	…
…	…	…

Table B		
ID	Amount	…
1	3063.0	…
2	4507.0	…
3	2713.5	…
…	…	…

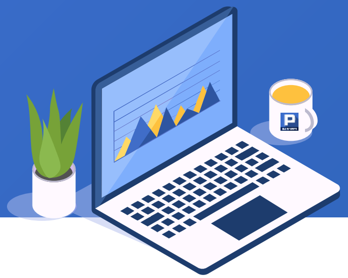
✦ 1. Concatenation：Concatenate sequence members



The function A.concat(d) concatenates the sequence members with the separator d and returns a string. The @c option indicates that comma is used to connect. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select COLUMN_NAME from INFORMATION_SCHEMA.KEY_COLUMN_USAGE k where k.TABLE_NAME='B'")	/Get the primary key of the table from the system table. Each database gets the primary key differently. Take MSSQL as an example.
3	=pks=A2.(COLUMN_NAME)	/Define variable pks，which is a sequence of primary key column names
4	=A1.query("select COLUMN_NAME from INFORMATION_SCHEMA.COLUMNS c where c.TABLE_NAME='B'")	/Retrieve all columns of the table
5	=columns=A4.(COLUMN_NAME)	/Define variable columns, which is a sequence of column names
6	="MERGE INTO A as t USING B as s ON "+pks.("t."+~+"=s."+~).concat(" and ")+" WHEN MATCHED THEN UPDATE SET "+(columns\pks).("t." + ~ +"=s." + ~).concat@c()+" WHEN NOT MATCHED THEN INSERT VALUES("+columns.("s."+ ~).concat@c()+")"	/Dynamically concatenate the “Merge into” statement. The A.concat function is used to concatenate the sequence members and return a string.
7	=A1.excute(A6)	/Execute the “merge into” statement in A6

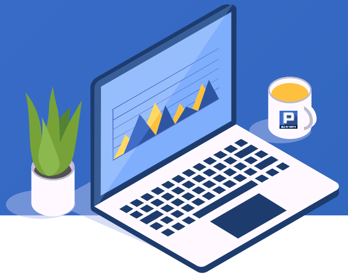
✦ 1. Concatenation：Concatenate sequence members



Query the states in which employees in each department are distributed, and state names are separated by spaces. Because some states have space in the name, they should be distinguished by quotation marks. The employee table is as follows:

ID	NAME	SURNAME	STATE	DEPT	SALARY
1	Rebecca	Moore	California	R&D	7000
2	Ashley	Wilson	New York	Finance	11000
3	Rachel	Johnson	New Mexico	Sales	9000
4	Emily	Smith	Texas	HR	7000
5	Ashley	Smith	Texas	R&D	16000
…	…	…	…	…	…

✦ 1. Concatenation：Concatenate sequence members

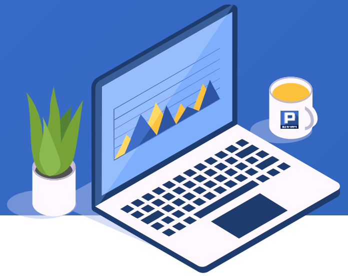


Use the @q option of the function A.concat() to concatenate strings with double quotation marks. Similarly, the @i option of A.concat() uses single quotation marks when concatenating strings. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Employee")	/Import employee table
3	=A2.group(DEPT; ~.id(STATE):STATES)	/Group by department, calculate the unique value of state name in each group
4	=A3.new(DEPT, STATES.concat@q(" "):STATES)	/Each group's state name is concatenated into a string using the A.concat() function, with the option @q member in double quotation marks.

A4	DEPT	STATES
	Administration	"Florida" "Pennsylvania"
	Finance	"California" "Colorado" "Florida" "Georgia" "Illinois" "Michigan" "New Jersey" "New York" "North Carolina"
	…	…

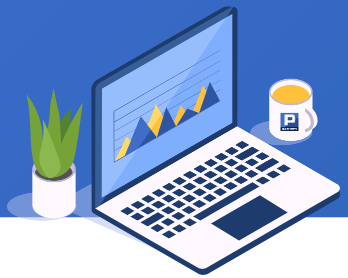
1. Concatenation：Concatenate sequence table members



Convert the Department table to CSV format and copy it to the system clipboard. The Department table is as follows:

ID	Name	Manager
1	Administration	1
2	Finance	4
3	HR	5
4	Marketing	6
5	Production	7
…	…	…

✦ 1. Concatenation：Concatenate sequence table members



The function A.export() concatenates the sequence table members and returns a string. The @t option is used to generate the header line, and the @c option is used to generate the CSV format. SPL is as follows:

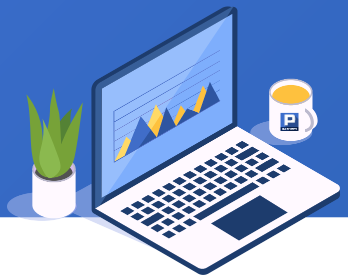
	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Department")	/Import Department table
3	=A2.export@ct()	/Use A.export() function to concatenate sequence members into a string. The @c option is used to generate the CSV format, and the @t option is used to generate the header line.
4	=clipboard(A3)	/Copy the string of A3 to the system clipboard

A2	ID	Name	Manager
	1	Administration	1
	2	Finance	4
	3	HR	5
	…	…	…

A3

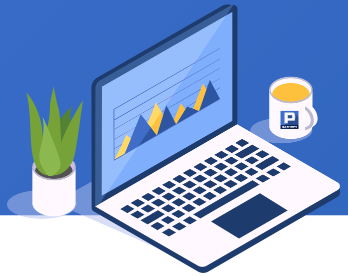
```
ID,Name,Manager
1,Administration,1
2,Finance,4
3,HR,5
…
```

✦ 2. Split: split into character sequence



Count the number of commas outside the bracket in the source code of a web page. The source code of the website is as follows:

```
<html>
<b></b>
<table cellpadding="2.5px" rules="all" style=";background-color: rgb(255,255,255);border: 1px
solid;border-collapse: collapse; border-color: rgb(187,187,187)">
<colgroup><col width="25px" style="background-color: rgb(218,231,245)" /><col /><col /><col
/><col /><col /><col /></colgroup>
<thead><tr style=" background-color: rgb(218,231,245);text-align: center;color:
rgb(22,17,32)"><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</t
h></tr></thead>
...
```



✦ 2. Split: split into character sequence

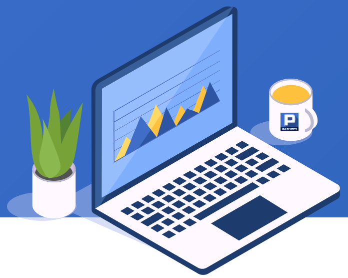
SPL is as follows, in which the function A.split() is used to split the strings into character sequences:

	A	B	C	B
1	=file("code.html").read()			/Read strings from file
2	=A1.split()	0	0	/Use function A.split() to split the strings into character sequences
3	for A2	if A3=="["	=B2+=1	/If the left bracket appears, B2 plus 1. It is used to match the bracket.
4		else if A3=="]"	=B2-=1	/If the right bracket appears, B2 minus 1. It is used to match the bracket.
5		else if A3==","&&B2==0	>C2+=1	/If a comma appears and the brackets match, the C2 count is added by 1. C2 is the number of commas outside brackets.

A2	Members
	<
	h
	t
	m
	…

C2	Value
	27

2. Split: split into word sequence



Count the top three words in an article. Part of the article is as follows:

How to Call an SPL Script in Java

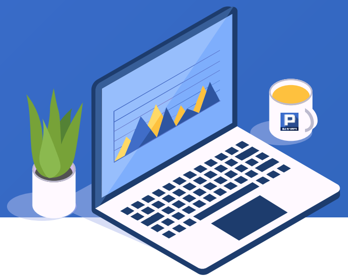
esProc provides its own JDBC driver to become integration-friendly with a Java application. The method of calling an esProc SPL script is similar to the execution of SQL queries and stored procedures in Java.

Deploying esProc JDBC in a Java application

Simply put, to deploy JDBC in a Java application is to put in place the necessary jars and configuration files for loading esProc when starting the application. esProc JDBC requires JDK 1.6 or a higher version.

...

✦ 2. Split: split into word sequence

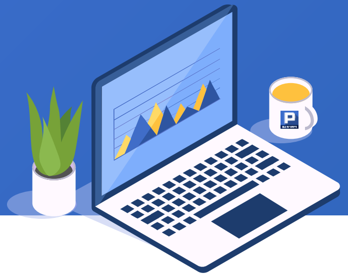


Use the function A.words() to split the strings into words. SPL is as follows:

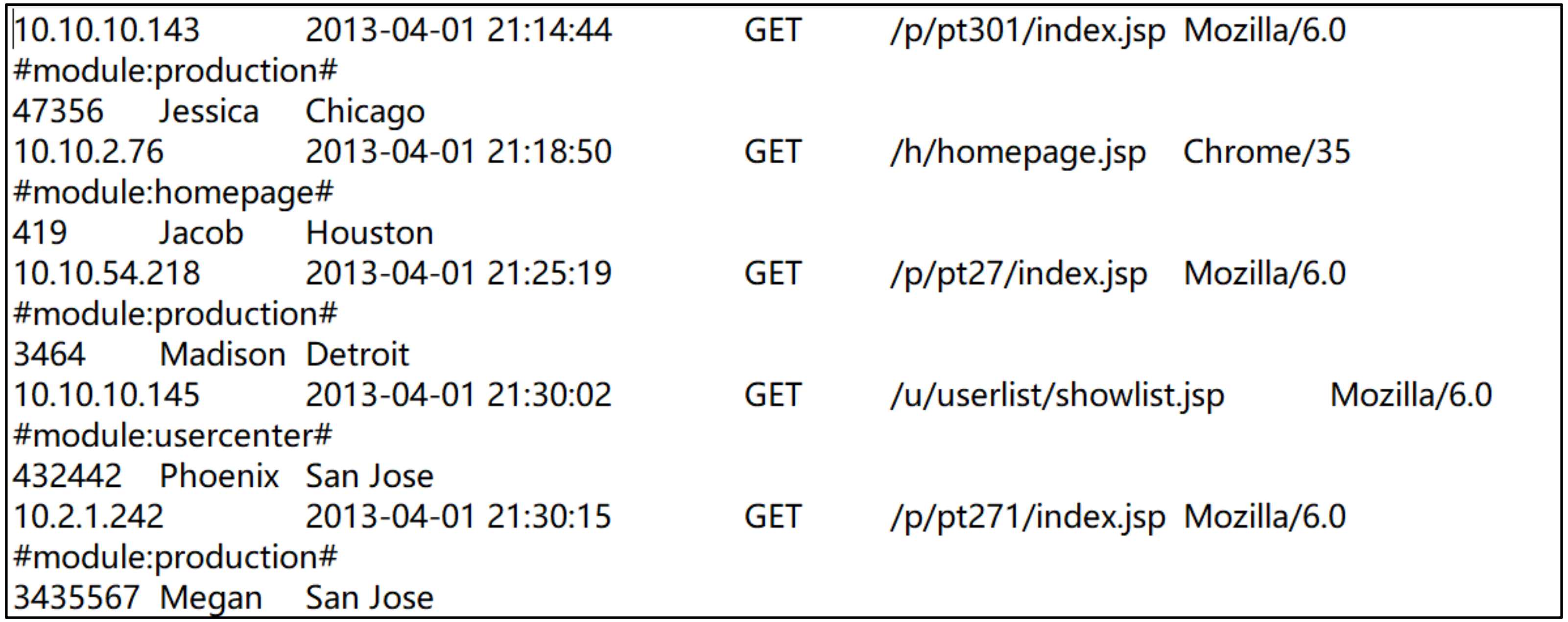
	A	B
1	=file("callSPL.txt").read()	/Read strings from file
2	=A1.words()	/Use the function A.words() to split the strings into words
3	=A2.group()	/Group words
4	=A3.ptop(-3;~.len())	/Select the top three words that appear most frequently
5	=A3(A4).(~(1)).concat@c()	/Concatenate the top three words into a string with commas

A5	Value
	the,property,name

✦ 2. Split: split with separator

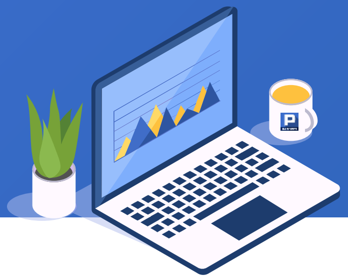


Organize the log file into structured data（a sequence table with the fields of USERID,UNAME,IP,TIME,URL,BROWSER,LOCATION, and MODULE）. Log format: the first line is IP, TIME, GET, URL, BROWSER; the second line is MODULE; the third line is USERID, UNAME, LOCATION.



```
10.10.10.143      2013-04-01 21:14:44        GET       /p/pt301/index.jsp  Mozilla/6.0
#module:production#
47356     Jessica    Chicago
10.10.2.76        2013-04-01 21:18:50        GET       /h/homepage.jsp     Chrome/35
#module:homepage#
419       Jacob      Houston
10.10.54.218      2013-04-01 21:25:19        GET       /p/pt27/index.jsp   Mozilla/6.0
#module:production#
3464      Madison    Detroit
10.10.10.145      2013-04-01 21:30:02        GET       /u/userlist/showlist.jsp        Mozilla/6.0
#module:usercenter#
432442    Phoenix    San Jose
10.2.1.242        2013-04-01 21:30:15        GET       /p/pt271/index.jsp  Mozilla/6.0
#module:production#
3435567   Megan      San Jose
```

✦ 2. Split: split with separator



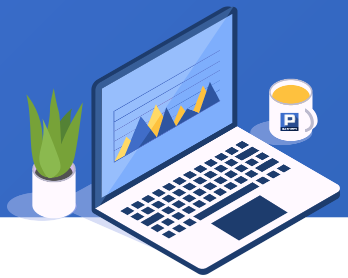
The function s.split(d) is used to split the string s into a sequence through the separator d. SPL is as follows:

	A	B
1	=file("log.txt").read@n()	/Read the sequence of strings by line from the file
2	=A1.group((#-1)\3)	/Use the group function to group every three rows
3	=A2.(~.conj(~.split("\t")))	/Using the s.split() function，each group splits each row by ” \t” and merges into a sequence
4	=A3.new(~(7):USERID,~(8):UNAME,~(1):IP,~(2):TIME,~(4):URL,~(5):BROWSER,~(9):LOCATION,left(~(6).split(":")(2),-1):MODULE)	/Generate structured data

A4

USERID	UNAME	IP	TIME	URL	BROWSER	LOCATION	MODULE
47356	Jessica	10.10.10.143	2013-04-01 21:14:44	/p/pt301/index.jsp	Mozilla/6.0	Chicago	production
419	Jacob	10.10.2.76	2013-04-01 21:18:50	/h/homepage.jsp	Chrome/35	Houston	homepage
...	…	…	…	…	…	…	…

✦ 2. Split: split with separator

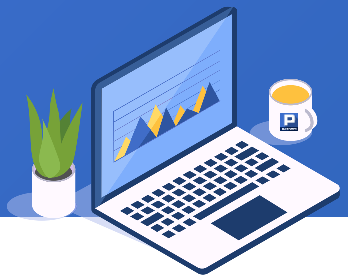


Query the product names purchased by customers, and separate them with commas when multiple products are purchased. The data of product table and customer sales table are as follows:

Product		
ID	**Name**	**Website**
R	Report	http://www.raqsoft.com.cn/r
P	esProc	http://www.raqsoft.com.cn/p
C	esCalc	http://www.raqsoft.com.cn/c
M	AI Models	http://www.yimming.com/
…	…	…

Sales		
ID	**Customer**	**Product**
1	VINET	R
2	TOMSP	P,R
3	HANAR	P,R,C
4	VICTE	P
…	…	…

✦ 2. Split: split with separator



The @c option of the function s.split(d) is used. When d is omitted, it is split with commas. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Product")	/Read product table
3	=A1.query("select * from Sales")	/Read sales table
4	=A3.run(Product=Product.split@c())	/Using @c option of the split function， split the products in sales table into sequence by comma and return
5	=A4.run(Product=Product.(A2.find(~).Name).concat@c())	/Find the product name according to the product ID, and concatenate the product name sequence into a string with commas

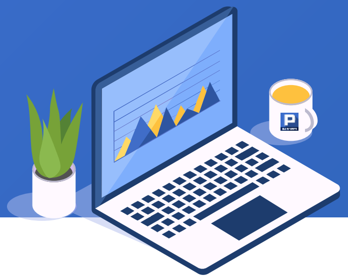
A4

ID	Customer	Product
1	VINET	[R]
2	TOMSP	[P,R]
3	HANAR	[P,R,C]
…	…	…

A5

ID	Customer	Product
1	VINET	Report
2	TOMSP	esProc,Report
3	HANAR	esProc,Report,esCalc
…	…	…

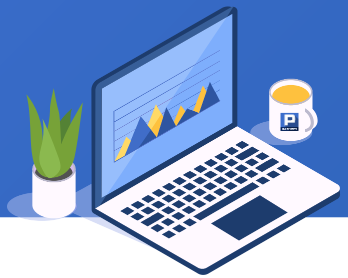
✦ 2. Split: split with separator



A website records the URL visited by users. Query the most frequently used search criteria. Some contents are as follows:

ID	User	Website
1	Rebecca	https://github.com/search?q=How+to+study+java%3F
2	Ashley	https://github.com/search?q=report&type=Code
3	Rachel	https://github.com/search?q=bigdata&type=Repositories
4	Rachel	https://github.com/search?l=Python&q=bigdata&type=Repositories
…	…	…

✦ 2. Split: split with separator

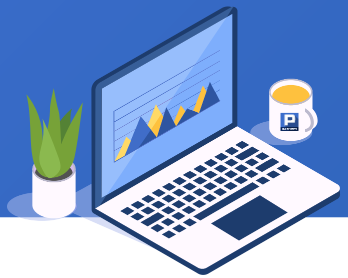


Using the @1 option of the function s.split(d) , find the first d and stop, that is, split into two segments. SPL is as follows:

	A	B
1	=file("loginUrls.txt").import@t()	/Read user login file
2	=A1.(Website.split@1("?")(2))	/Using the @1 option of the s.split() function, split into two segments according to the first ?
3	=A2.(~.split("&").select@1(like(~,"q=*")))	/The parameter is split by & , and select the criteria of q = * , which is the user search criteria
4	=A3.(~.split@1("=")(2))	/Split the search criteria into two segments by = and the second part is the search criteria
5	=A4.group()	/Group by search criteria
6	=A5.maxp(~.len())(1)	/Select the group with the largest number, which is the most frequently used search criteria

A6
Value
bigdata

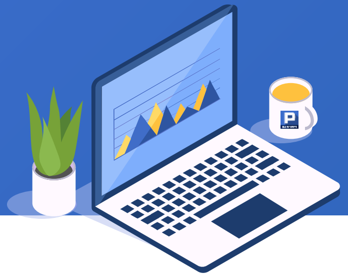
✦ 2. Split: split with regular expression



Remove all comments (<! --- >) from the HTML file. Part of the contents are as follows:

```
<html>
<!-- Row Highlight Javascript -->
<script type="text/javascript">
        window.onload=function(){
        var tfrow = document.getElementById('tfhover').rows.length;
        var tbRow=[];
        ...
};
...
</html>
```

✦ 2. Split: split with regular expression



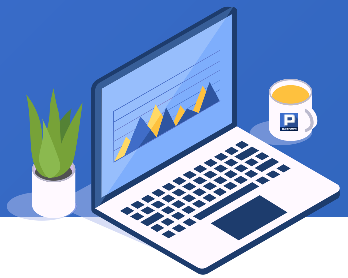
The @r option of the function s.split(d) is used, and d is interpreted as a regular expression. SPL is as follows:

	A	B
1	=file("table.html").read()	/Read html file
2	=A1.split@r("<!--.*-->")	/Use the @r option of the s.split() function to split the string according to the regular expression
3	=A2.concat()	/Concatenate the strings after splitting, that is, the HTML format string without comments
4	>file("table.html").write(A3)	/Write the string to file

A3

Value
<html><script type="text/javascript">window.onload=function(){…

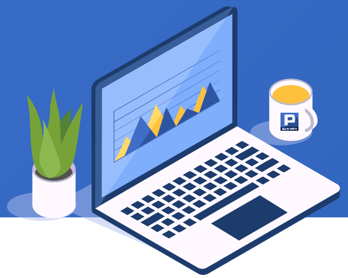
✦ 3. Parse：parse numerical value



The model performance table records various indexes of different models. We want to select the numerical target model (modeltype is 2) and present it with the indexes as the column names. Some data are as follows:

ID	ModelName	ModelType	Performance
1	HousePrice	2	SquareR=0.933743
2	HousePrice	2	MSE=295749426.986263
3	HousePrice	2	RMSE=17197.366862
4	HousePrice	2	GINI=0.197449
5	HousePrice	2	MAE=12509.456071
6	HousePrice	2	MAPE=7.798386
7	Titanic	1	GINI=0.654867
8	Titanic	1	AUC=0.827434
9	Titanic	1	KS=0.587658
…	…	…	…

✦ 3. Parse: parse numerical value



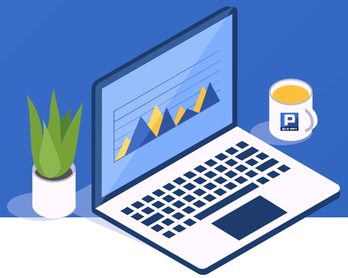
The function number(stringExp) is used to parse the string stringExp into a numeric value. SPL is as follows:

	A	B
1	=file("mps.txt").import@t()	/Import model performance file
2	=A1.select(ModelType:2)	/Select model type 2
3	=A2.group(ModelName)	/Group by model name
4	=A3(1).(Performance.split("=")(1)).concat@c()	/Concatenate the first group of index names into a string with commas
5	=create(${"ModelName,"+A4})	/Create an empty sequence table. The first column is the model name, followed by the model performance indexes
6	=A3.(A5.record(A3.~.ModelName | A3.~.(number(Performance.split("=")(2)))))	/The index is inserted into the sequence table of A5 in loop. Here we use the number() function to convert the split string into a numeric value.

A5

ModelName	SquareR	MSE	RMSE	GINI	MAE	MAPE
HousePrice	0.933743	295749426.986263	17197.366862	0.197449	12509.456071	7.798386
…	…	…	…	…	…	…

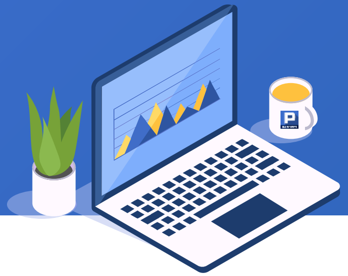
3. Parse：parse numerical value



According to the prediction results of Titanic survival model, count the proportion of female within people with survival probability over 80%. Some data are as follows:

Survived	PassengerId	Pclass	Name	Sex	…
Percent:10.461%	624	3	Braund, Mr. Owen Harris	male	…
Percent:9.108%	625	3	Cumings, Mrs. John Bradley	male	…
Percent:8.891%	626	1	Heikkinen, Miss. Laina	male	…
Percent:50.510%	627	2	Futrelle, Mrs. Jacques Heath	male	…
…	…		…	…	…

3. Parse：parse numerical value



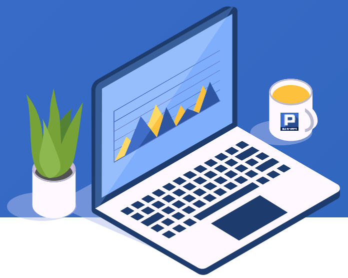
The function number(stringExp, format) is used to parse the string stringExp into a numerical value in fmt format. SPL is as follows:

	A	B
1	=file("titanic.csv").import@cqt()	/Import Titanic data file
2	=A1.run(Survived=Survived.split(":")(2))	/Split the field of survival probability by ":" and take the second part
3	=A2.run(Survived=number(Survived, "0%"))	/Use the number() function to parse the value in the specified format
4	=A3.select(Survived > 0.8)	/Select people with a survival probability of more than 80%
5	=string(A4.count(Sex=="female") / A4.len() , "0.000%")	/Calculate the proportion of females

A5

Value
97.619%

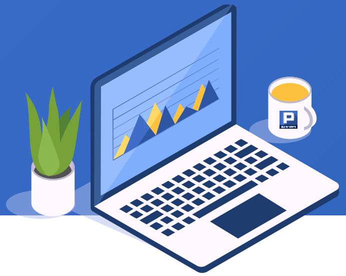
3. Parse：parse any single value



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

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]
…	…	…

✦ 3. Parse：parse any single value

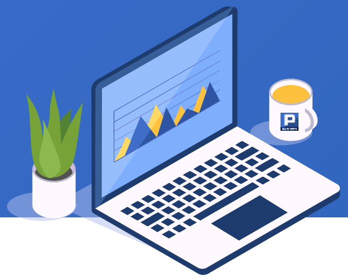


The parse(s) function parses the string s into the corresponding data type. SPL is as follows:

	A	B
1	=file("Olympic.csv").import@cqt()	/Import ranking of previous Olympic Games
2	=A1.run(Medal=parse(Medal))	/Use the parse() function to parse the medal field into a sequence
3	=A2.group(Game)	/Group by Game
4	=A3.select(~.select(Nation=="China").Medal>~.select(Nation=="Russia").Medal)	/Comparing the medal sequence size of China and Russia with ">" symbol, the number of gold medal, silver medal and bronze medal will be compared in order, and the higher ranking session of China will be selected.
5	=A4.(Game)	/List the sessions

A5	Game
	23
	25
	28
	29
	30

3. Parse：parse multiple values

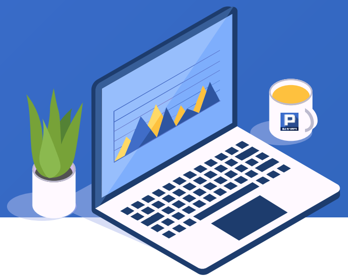


There are course table and course selection table. Query which courses are not selected by students. Multiple courses can be selected and separated by commas. Part of the data are as follows:

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
…	…	…

✦ 3. Parse：parse multiple values



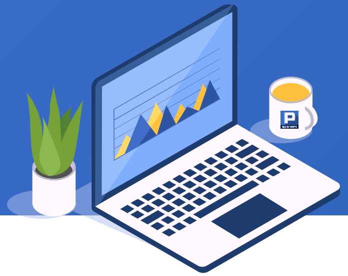
The @p option of the function s.split() is used to parse the members into corresponding data types after splitting.

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 course selection table
4	=A3.union(COURSE.split@cp())	/Using the @p and @c options of split function, the courses in the course selection table are split by commas and parsed into integers, and get the union of the course sequence by union() function
5	=A2.(ID)	/IDs of all courses
6	=A2(A5.pos([A5,A4].diff()))	/Use the diff() function to find the ID difference between the course table and the course selection table, that is, the course that no one selected. After positioning in A5, select from A2.

A6

ID	NAME	TEACHERID
1	Fundamentals of economic management	21

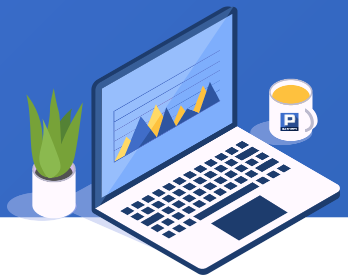


✦ 3. Parse： parse sequence table

The GDP and population data of major cities in China are copied from the system's clipboard. Export them to the file of CSV format. Part of the contents are as follows:

```
ID      City    GDP     Population
1       Shanghai        32679   2418
2       Beijing 30320   2171
3       Shenzhen        24691   1253
4       Guangzhou       23000   1450
5       Chongqing       20363   3372
6       Tianjin 18809   1557
7       Suzhou18597     1068
8       Chengdu         15342   1605
…
```

✦ 3. Parse： parse sequence table



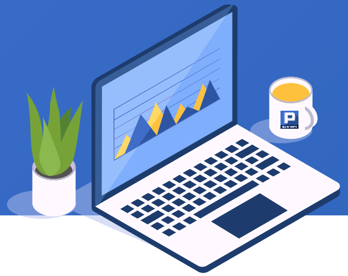
The S.import (;s) function is used to import the contents read out from the string S as records and return as the sequence table. Where s is the separator and the default is tab. The option @t means that the first line is the title. SPL is as follows:

	A	B
1	=clipboard()	/Return the contents of the clipboard as strings
2	=A1.import@t()	/Read the strings into sequence table. The separator is tab （\t） by default. The option @t means that the first line is the title
3	>file("GDP.csv").export@ct(A2)	/Export the sequence table of A2 to GDP.csv file

GDP.csv

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

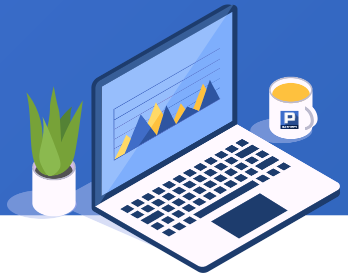
✦ 3. Parse：parse regular expression to match string



Take the number from customer address. Part of customer table are as follows:

ID	Name	City	Address
1	VINET	Beijing	124 Guangming North Road
2	TOMSP	Jinan	543 Qingnian East Road
3	HANAR	Qinhuangdao	22 Guanghua Street
4	VICTE	Nanjing	Qinglin bridge 68
…	…	…	…

✦ 3. Parse：parse regular expression to match string

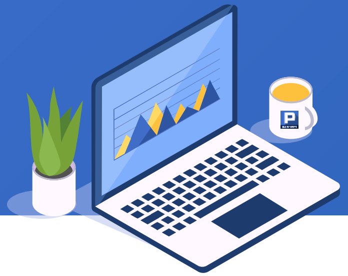


The function S.regex(rs) is used to match the string S with the regular expression rs. The array of matching sections is returned, and null is returned if the string is not matched. SPL is as follows:

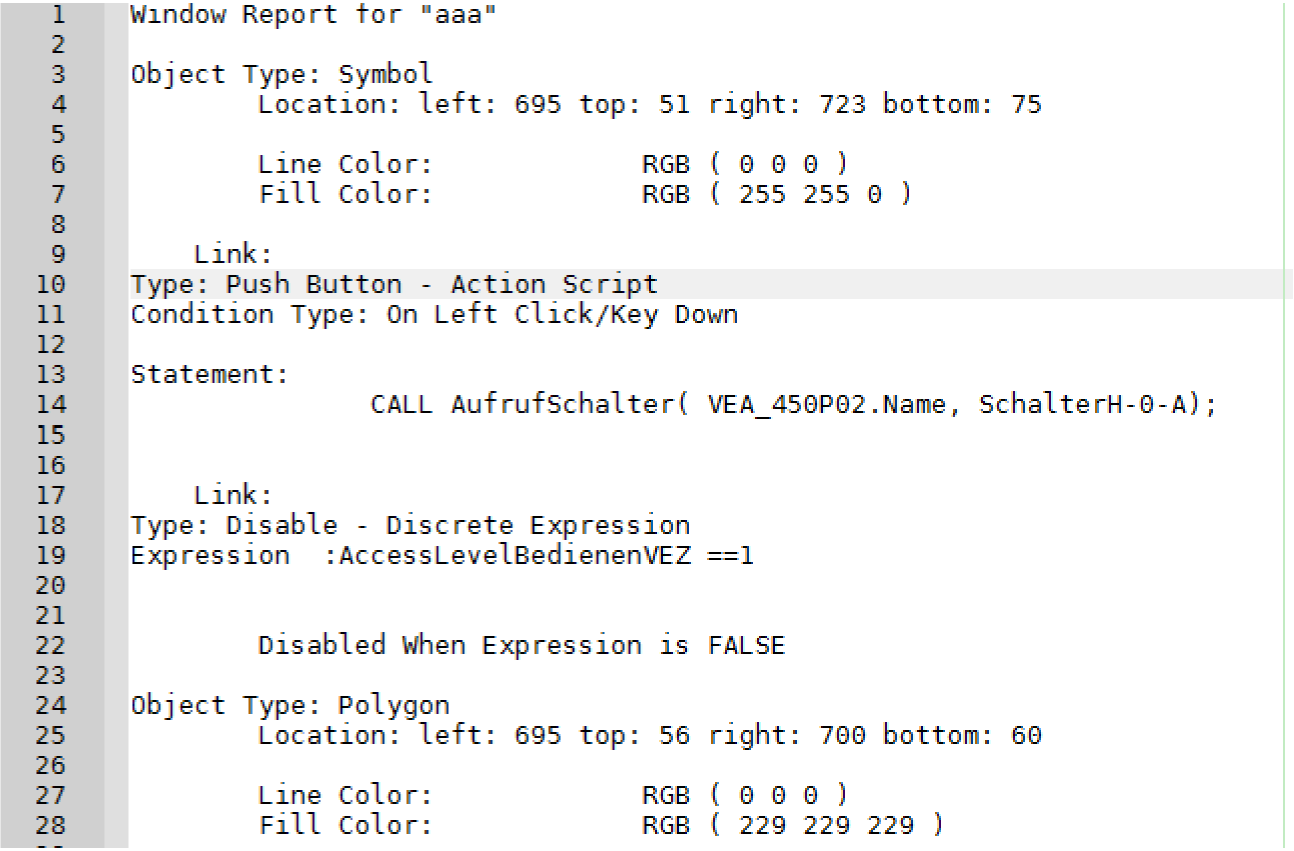
	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Customer")	/Read customer table
3	=A2.run(Address=number(Address.regex("\\D*(\\d+)\\D*")(1)))	/Use S.regex() function，get the street nameplate from address, and parse into number.

A3	ID	Name	City	Address
	1	VINET	Beijing	124
	2	TOMSP	Jinan	543
	3	HANAR	Qinhuangdao	22
	…	…	…	…

✦ 3. Parse： parse regular expression to match string

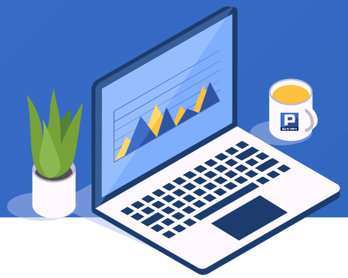


The log file is composed of indefinite number of lines. Need to be parsed into structured data. Part of the log file is as follows:



```
Window Report for "aaa"

Object Type: Symbol
        Location: left: 695 top: 51 right: 723 bottom: 75

        Line Color:             RGB ( 0 0 0 )
        Fill Color:             RGB ( 255 255 0 )

    Link:
Type: Push Button - Action Script
Condition Type: On Left Click/Key Down

Statement:
               CALL AufrufSchalter( VEA_450P02.Name, SchalterH-0-A);


    Link:
Type: Disable - Discrete Expression
Expression  :AccessLevelBedienenVEZ ==1


        Disabled When Expression is FALSE

Object Type: Polygon
        Location: left: 695 top: 56 right: 700 bottom: 60

        Line Color:             RGB ( 0 0 0 )
        Fill Color:             RGB ( 229 229 229 )
```

✦ 3. Parse: parse regular expression to match string

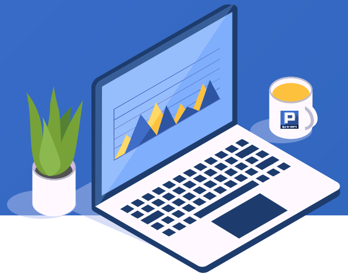


The function A.regex (rs,Fi) matches the string members in sequence A with the regular expression rs, and returns the results to form a sequence table with Fi as the field. SPL is as follows:

	A	B
1	=file("report.log").read()	/Read the log file and return as string
2	=A1.split("Object Type:").delete(1)	/Split the text content into multiple records according to the mark "Object Type:" and discard the first record
3	=A2.regex("(.+)[\\s\\S]+left:(.+)[\\s\\S]+top:(.+)[\\s\\S]+right:(.+)[\\s\\S]+bottom:(.+)[\\s\\S]+Line Color:(.+)[\\s\\S]+Fill Color:\\t\\t(.+)[\\S\\s]+Link:(.+)[\\s\\S]+Type: (.+)[\\s\\S]+Condition Type:(.+)[\\s\\S]+Statement:\\s+(.+)[\\s\\S]+Link:(.+)[\\s\\S]+Type: (.+)[\\s(\\S]+Expression :(.+)";ObjectType,left,top,right,bottom,lineColor,fillColor,ojbectLink,type,conditionType,statement,statementLink,statementType,lastExpress)	/Match each record with regular expression to find the complete record
4	=file("result.txt").export@t(A3)	/Export A3 result to result.txt

A3	ID	ObjectType	left	top	right	bottom	lineColor	fillColor	…
	1	Symbol	695	51	723	75	RGB (0 0 0)	RGB (255 255 0)	…
	…	…	…	…	…	…	…	…	…

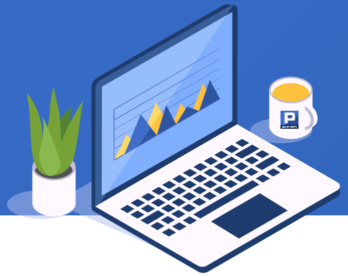
✦ 3. Parse：parse using codes



Find the average salary of employees who were born in the 1980s, and the age needs to be extracted from the ID number. Part of the employee table are as follows:

ID	Name	Identification	Salary
1	Rebecca	Driving license:495319197411204628	7000
2	Ashley	ID number:103263198007194980	11000
3	Rachel	ID number:721125197012173641	9000
4	Emily	ID number:619124198503071617	7000
5	Ashley	ID number:248238197505138795	16000
…	…	…	…

✦ 3. Parse：parse using codes

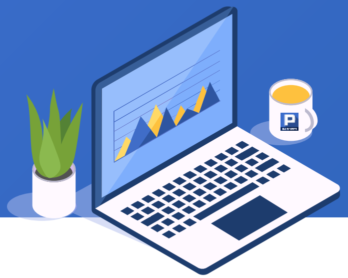


When a single function cannot solve the problem directly, multiple functions can be used to parse and process strings step by step. SPL is as follows:

	A	B
1	=connect("db").query("select * from Employee")	/Connect to database，read employee table
2	=A1.run(Identification=Identification.regex("\\D*(\\d+)")(1))	/Use S.regex() function to read the number part of ID
3	=A2.run(Identification=mid(Identification,7,4))	/Use the mid() function to read No.7 to No.10 digits of the ID, that is, the year of birth
4	=A3.run(Identification=number(Identification))	/Use the number() function to parse the year string into a number
5	=A4.select(Identification>=1980 && Identification <=1989)	/Select employees born in the 1980s
6	=A5.avg(Salary)	/Calculate the average salary

A6	ID
	7256.16

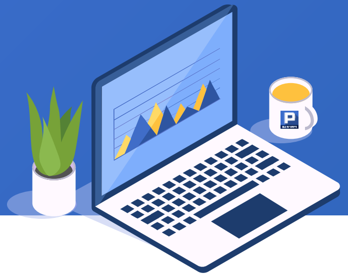
✦ 4. Complex processing: SQL analysis



The following SQL statement is used to select the employees in sales department whose salary is greater than 10000. Modify the department in the filter condition to the R&D department.

```
select
        EID,NAME,SURNAME,DEPT,SALARY
from
        Employee
where
        DEPT='Sales' and SALARY>10000
```

✦ 4. Complex processing: SQL analysis



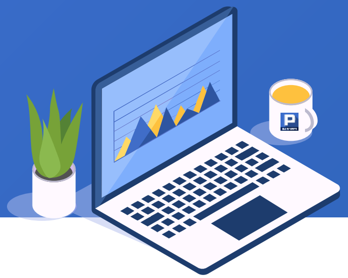
The function s.sqlparse(part) is used to split SQL into a sequence composed of various parts. The part parameter is used to replace the corresponding part of SQL with part and return the new SQL. The option @w represents the where statement and @s represents the select statement. SPL is as follows:

	A	B
1	select EID,NAME,SURNAME,DEPT,SALARY from Employee where DEPT='Sales' and SALARY>10000	/Define SQL constants
2	=A1.sqlparse@w()	/Use the @w option of s.sqlparse() function to get the where condition
3	=A2.split@t("and")	/Use the s.split() function to split the where condition, and the option @t indicates that each segment is trimed
4	=A3.pselect(like(~,"DEPT*"))	/Select the department condition
5	=A3(A4)="DEPT='R&D'"	/Replace the condition in department to R&D
6	=A3.concat(" and ")	/Concatenate the condition sequence with and
7	=A1.sqlparse@w(A6)	/Use the @w option of the s.sqlparse(part) function to replace the where condition

A6

Value
select EID,NAME,SURNAME,DEPT,SALARY from Employee where DEPT='R&D' and SALARY>10000

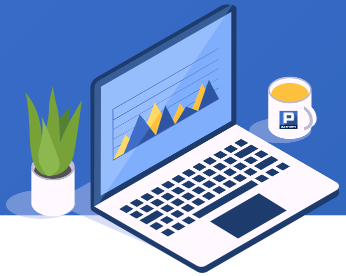
✦ 4. Complex processing: SQL translation



Part of the sales data of a company is stored in Oracle and some in MySQL. Find the number of orders with sales over 1000 during the period from March 18 to July 18, 2015. The order table structure is the same, as follows:

ORDERID	CUSTOMERID	EMPLOYEEID	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. Complex processing: SQL translation



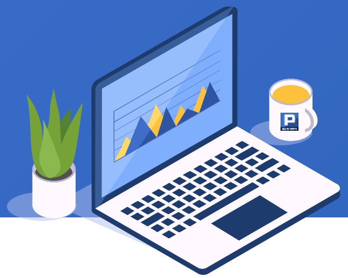
Function sql.sqltranslate(dbtype) translates functions in standard SQL into the format of the specified database.

SPL is as follows:

	A	B
1	select ORDERID,CUSTOMERID,EMPLOYEEID,ORDERDATE,AMOUNT from ORDERS where ORDERDATE between date('2015-03-18') and date('2015-07-18') and AMOUNT>1000	/Standard SQL
2	=A1.sqltranslate("ORACLE")	/Translate standard SQL into Oracle format
3	=A1.sqltranslate("MYSQL")	/Translate standard SQL into MYSQL format
4	=connect("oracle").query(A2)	/Connect Oracle and execute SQL
5	=connect("mysql").query(A3)	/Connect mysql and execute SQL
6	=[A4,A5].merge@ou(ORDERID)	/Orderly merge, remove orders with the same ID
7	=A6.len()	/Count quantity

A7	Value
	63

✦ 4. Complex processing: SQL translation



Standard SQL（A1）：

```
select ORDERID,CUSTOMERID,EMPLOYEEID,ORDERDATE,AMOUNT from ORDERS where ORDERDATE  between date('2015-03-18') and date('2015-07-18') and AMOUNT>1000
```

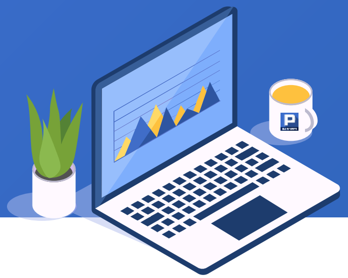
ORACLE（A2）：

```
select ORDERID,CUSTOMERID,EMPLOYEEID,ORDERDATE,AMOUNT from ORDERS where ORDERDATE  between TO_DATE('2015-03-18','YYYY-MM-DD') and TO_DATE('2015-07-18','YYYY-MM-DD') and AMOUNT>1000
```

MYSQL（A3）：

```
select ORDERID,CUSTOMERID,EMPLOYEEID,ORDERDATE,AMOUNT from ORDERS where ORDERDATE  between DATE_FORMAT('2015-03-18','%Y-%m-%d') and DATE_FORMAT('2015-07-18','%Y-%m-%d') and AMOUNT>1000
```

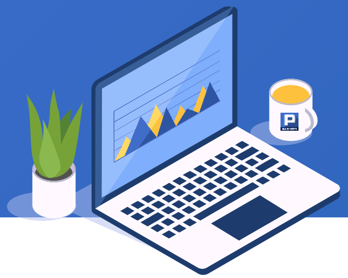
✦ 4. Complex processing: JSON processing



The following is the JSON data of the order information. It is divided into two levels: the first level is the country and area, and the second level is the detailed data. Now we want to import orders from North and South China in 2013.

```
[{"COUNTRY":"China","AREA":"Northeast China","ORDERS":[
{"ORDER_ID":10252,"CUSTOMER_ID":"SUPRD","EMPLOYEE_ID":4, …},
{"ORDER_ID":10318,"CUSTOMER_ID":"ISLAT","EMPLOYEE_ID":8, …},
…]},
{"COUNTRY":"China","AREA":"East China","ORDERS":[
{"ORDER_ID":10249,"CUSTOMER_ID":"TOMSP","EMPLOYEE_ID":6, …},
{"ORDER_ID":10251,"CUSTOMER_ID":"VICTE","EMPLOYEE_ID":3, …},
…]},
…]
```

✦ 4. Complex processing: JSON processing



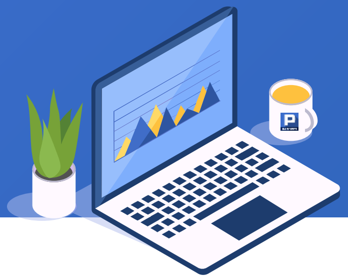
SPL is as follows, in which the function json(x) is used to read the string x into multi-layer records or sequence table in JSON format:

	A	B
1	=file("Orders.json").read()	/Read strings from JSON file
2	=json(A1)	/Use json(x) function to read A1 into multi-layer records or sequence table
3	[North China, South China]	/Define the sequence of constants: North and South China
4	=A2.select(COUNTRY=="China" && A3.contain(AREA))	/Filter records by criteria
5	=A4.news(ORDERS;COUNTRY, AREA,${A4.ORDERS.fname().concat@c()})	/Generate the sequence table, which is composed of fields of country, area and order details
6	=A5.select(year(ORDER_DATE)==2013)	/Select the records of 2013

A6

COUNTRY	AREA	ORDER_ID	CUSTOMER_ID	EMPLOYEE_ID	ORDER_DATE	…
China	North China	10402	ERNSH	8	2013-01-02	…
China	North China	10403	ERNSH	4	2013-01-03	…
…	…	…	…	…	…	…

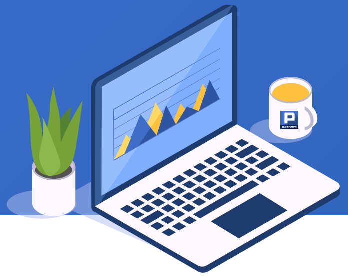
4. Complex processing: XML processing



In the configuration file raqsoftConfig.xml, query log level. Some contents are as follows:

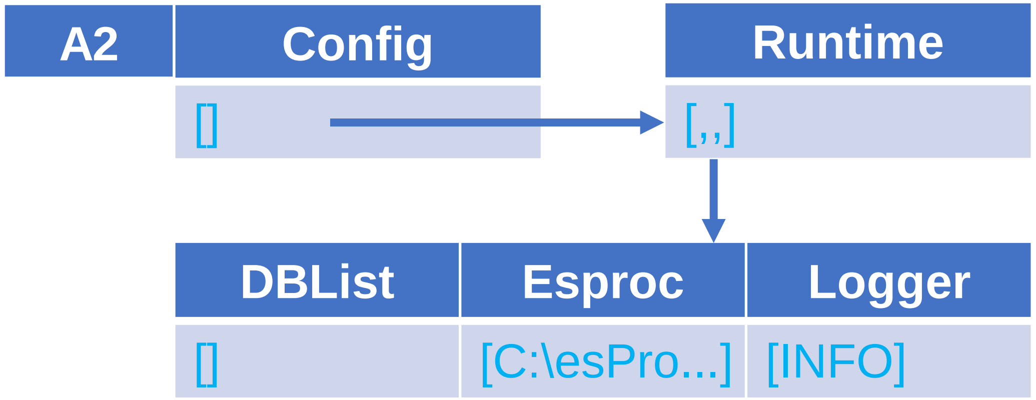
```xml
<?xml version="1.0" encoding="UTF-8"?>
<Config Version="2">
  <Runtime>
    ...
    <Logger>
      <Level>INFO</Level>
    </Logger>
  </Runtime>
</Config>
```

✦ 4. Complex processing: XML processing



SPL is as follows, in which the @x option of function s.import() is used to read from XML format string into multi-layer records or sequence table:

	A	B
1	=file("raqsoftConfig.xml").read()	/Read strings from file
2	=A1.import@x()	/Use s.import() to read from XML format string into multi-layer records or sequence table
3	=A2.Config.Runtime.Logger.Level	/Read log level



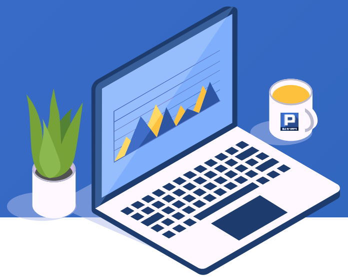
A2	Config
	[]

Runtime
[,,]

DBList	Esproc	Logger
[]	[C:\esPro...]	[INFO]

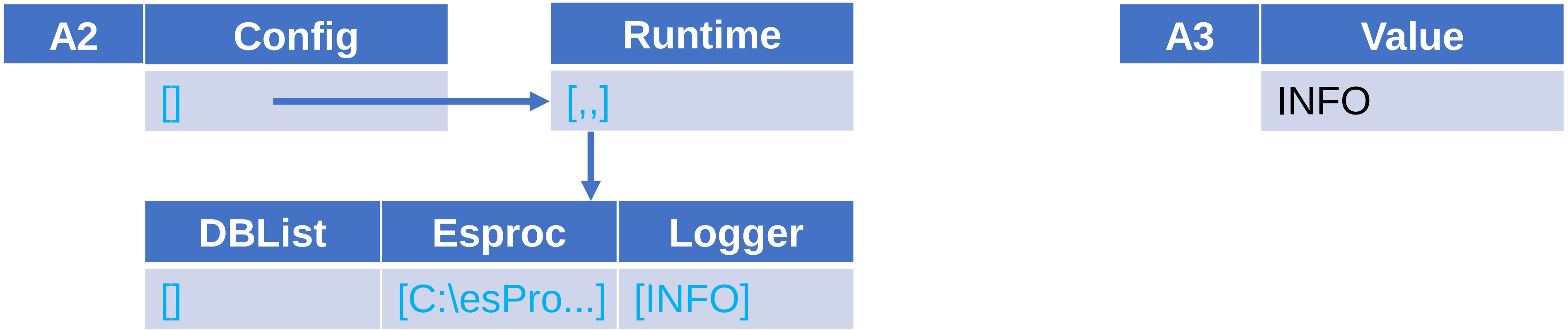
A3	Value
	INFO

✦ 4. Complex processing: XML processing



The function s.import@x() can be simplified to function xml(x), which parses the XML format string x into multi-layer records or sequence table. SPL is as follows:

	A	B
1	=file("raqsoftConfig.xml").read()	/Read strings from file
2	=xml(A1)	Use xml(x) function to parse XML format string x into multi-layer records or sequence table
3	=A2.Config.Runtime.Logger.Level	/Read log level



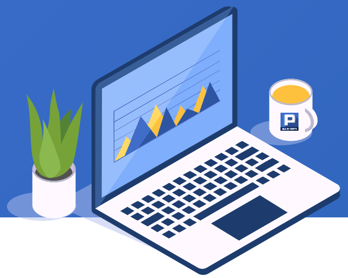
A2	Config
	[]

Runtime
[,,]

DBList	Esproc	Logger
[]	[C:\esPro...]	[INFO]

A3	Value
	INFO

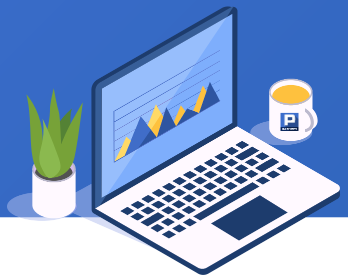
✦ 4. Complex processing: XML processing



In the configuration file raqsoftConfig.xml, query the list of database names. Part of the contents are as follows:

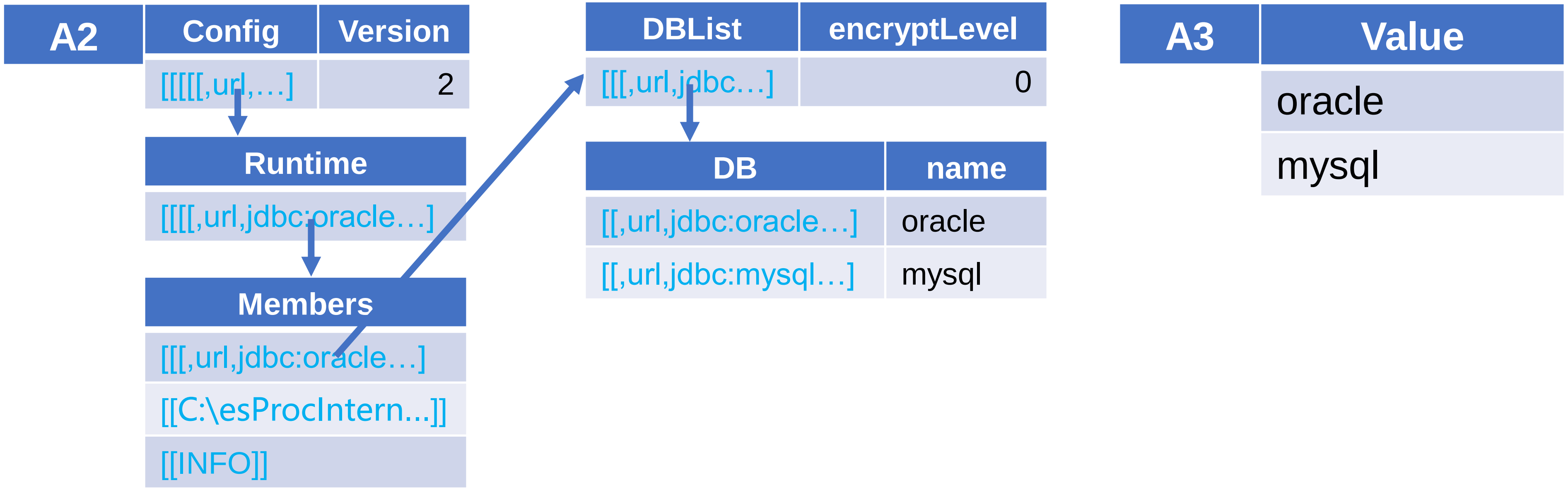
```
<?xml version="1.0" encoding="UTF-8"?>
<Config Version="2">
    <Runtime>
        <DBList encryptLevel="0">
            <DB name="oracle">
                <property name="url" value="jdbc:oracle:thin:@127.0.0.1:1521:root"/>
                ...
            </DB>
            <DB name="mysql"> ... </DB>
        </DBList>
        ...
    </Runtime>
</Config>
```

✦ 4. Complex processing: XML processing



The @s option of the function xml() parses XML string like <K F=v F=v …>D</K> into records with K,F,… as fields. Function s.import@x() also supports the @s option. SPL is as follows:

	A	B
1	=file("raqsoftConfig.xml").read()	/Read strings from file
2	=xml@s(A1)	/Use xml() function to parse XML format string into multi-layer records or sequence table
3	=A2.Config.Runtime(1).DBList.(name)	/Read database name list



A2	Config	Version
	[[[[,url,…]	2

Runtime
[[[,url,jdbc:oracle…]

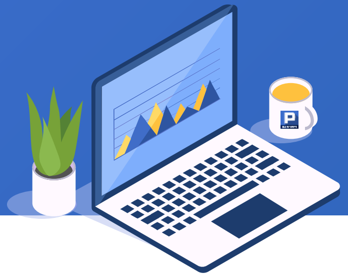
Members
[[[,url,jdbc:oracle…]
[[C:\esProcIntern...]]
[[INFO]]

DBList	encryptLevel
[[[,url,jdbc…]	0

DB	name
[[,url,jdbc:oracle…]	oracle
[[,url,jdbc:mysql…]	mysql

A3	Value
	oracle
	mysql

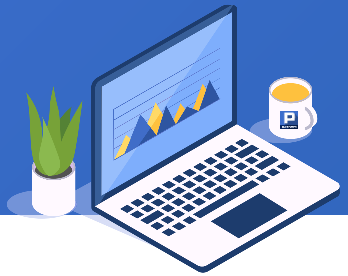
✦ 4. Complex processing: XML processing



In the configuration file raqsoftConfig.xml, query esProc configuration. Part of the contents are as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<Config Version="2">
    <Runtime>
        …
        <Esproc>
            <license>C:\esProcInternal.xml</license>
            <charSet>GBK</charSet>
            <dateFormat>yyyy-MM-dd</dateFormat>
            <timeFormat>HH:mm:ss</timeFormat>
            <dateTimeFormat>yyyy-MM-dd HH:mm:ss</dateTimeFormat>
            …
        </Esproc>
        …
    </Runtime>
</Config>
```

4. Complex processing: XML processing



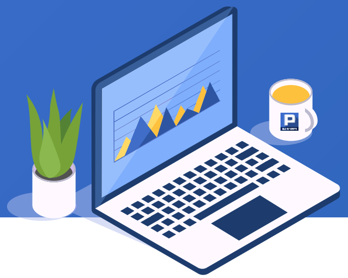
In the function xml(x,s), the parameter s indicates the layer identification to be taken out, the multi-layer is separated by / and the null indicates that it is taken from the root. SPL is as follows:

	A	B
1	=file("raqsoftConfig.xml").read()	/Read strings from file
2	=xml(A1, "Config/Runtime/Esproc")	/Use xml(x,s) function to obtain the contents of specified layers

A2

license	charSet	dateFormat	timeFormat	dateTimeFormat	…
C:\esProcInternal.xml	GBK	yyyy-MM-dd	HH:mm:ss	yyyy-MM-dd HH:mm:ss	…

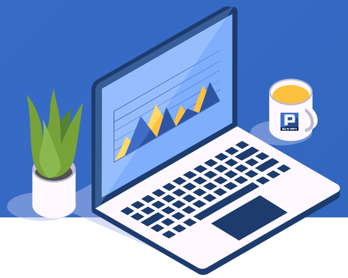
✦ 4. Complex processing: HTML analysis



Find the numbers in the body of the HTML file. Part of the contents of the file are as follows:

```
<!DOCTYPE html>
<html class="html__responsive html__unpinned-leftnav">
<head>
 <title>Stack Overflow - Where Developers Learn, Share, &amp; Build Careers</title>
        <link rel="shortcut icon" href="https://cdn.sstatic.net/Sites/stackoverflow/Img/favicon.ico?v=ec617d715196">
        <link rel="apple-touch-icon" href="https://cdn.sstatic.net/Sites/stackoverflow/Img/apple-touch-icon.png?v=c78bd457575a">
        <link rel="image_src" href="https://cdn.sstatic.net/Sites/stackoverflow/Img/apple-touch-icon.png?v=c78bd457575a">
...
</html>
```

4. Complex processing: HTML analysis

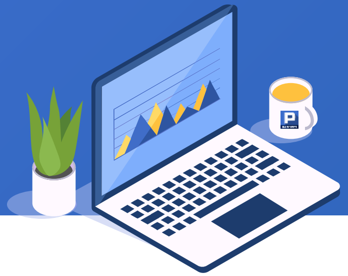


The function s.htmlparse() is used to get all the text in the HTML file. SPL is as follows:

	A	B
1	=file("sof.html").read()	/Read html file
2	=A1.htmlparse()	/Use the htmlparse() function to parse the HTML string and return the sequence of all the text
3	=A2.(~.words@d()).conj()	/Calculate the parsed text sequence in loop, take out the number in each string, and then get the concatenation column

A3	Members
	30
	3
	16.5
	5
	…

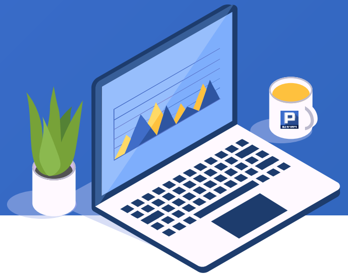
4. Complex processing: HTML analysis



Parse the score table from HTML file, and count the total score of each student.

```
<html>
…
<table id="tfhover" class="tftable" border="1">
<tr><th>CLASS</th><th>STUDENTID</th><th>SUBJECT</th><th>SCORE</th></tr>
<tr><td>Class one</td><td>1</td><td>Math</td><td>77</td>
<tr><td>Class one</td><td>1</td><td>PE</td><td>69</td>
<tr><td>Class one</td><td>1</td><td>English</td><td>84</td>
<tr><td>Class one</td><td>2</td><td>Math</td><td>80</td>
<tr><td>Class one</td><td>2</td><td>PE</td><td>97</td>
…
</table>
…
</html>
```

✦ 4. Complex processing: HTML analysis



The function s.htmlparse(tag:i:j) is used to get the jth text under the ith tag in HTML format string s. SPL is as follows:

	A	B
1	=file("table.html").read()	/Read html file
2	=A1.htmlparse("table":0)	/Use the htmlparse() function to parse the HTML string and return all the contents of the first table tag
3	=create(${A2(1).concat@c()})	/Create a sequence table with the first row heading
4	=A3.record(A2.to(2,).conj())	/Insert the data starting from the second row in turn into the sequence table of A3
5	=A3.groups(STUDENTID; sum(SCORE):TOTALSCORE)	/Group and aggregate the student score table, calculate the total score of each student

A3	STUDENTID	TOTALSCORE
	1	230
	2	258
	3	228
	…	…

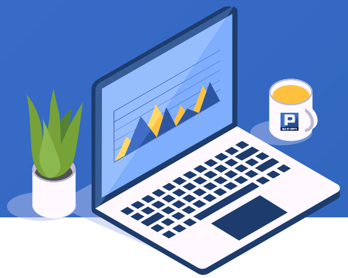
CONTENTS

1. Calculate the single value of date and time
2. Calculate date time sequence

02

Date time

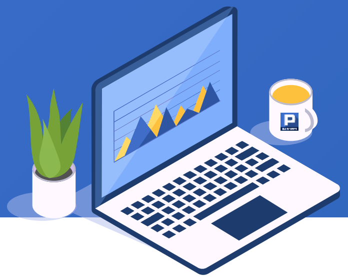
✦ 1. Calculate the single value of date and time



Query the orders delivered within the second day and arrived within three days after delivery in 2015. The order table is as follows:

ID	CustomerID	OrderDate	DeliveryDate	ArrivalDate	Amount
10248	VINET	2012/07/04	2012/07/16	2012/08/01	428.0
10249	TOMSP	2012/07/05	2012/07/10	2012/08/16	1842.0
10250	HANAR	2012/07/08	2012/07/12	2012/08/05	1523.5
10251	VICTE	2012/07/08	2012/07/15	2012/08/05	624.95
10252	SUPRD	2012/07/09	2012/07/11	2012/08/06	3559.5
…	…	…	…	…	…

✦ 1. Calculate the single value of date and time

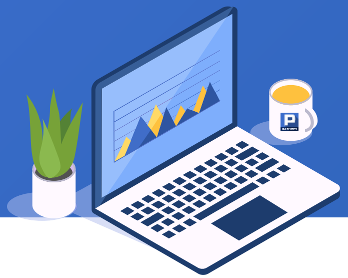


Use date + n to calculate the date on the nth day after the date. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Orders")	/Read order table
3	=A2.select(year(OrderDate)==2015 && OrderDate+1>=DeliveryDate && DeliveryDate+3>=ArrivalDate)	/Use the "+" symbol to calculate the date on the nth day after the date.

A3	ID	CustomerID	OrderDate	DeliveryDate	ArrivalDate	Amount
	11094	BERGS	2015/07/18	2015/07/18	2015/07/19	506.05
	11101	AROUT	2015/07/18	2015/07/18	2015/07/20	130.0
	11102	AROUT	2015/07/18	2015/07/19	2015/07/20	240.0
	…	…	…	…	…	…

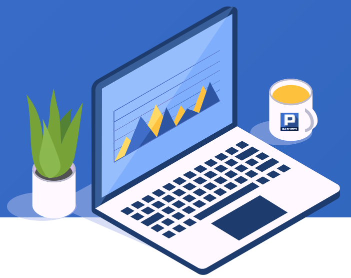
✦ 1. Calculate the single value of date and time



Query orders with delivery date over 30 days in 2015. The order table is as follows:

ID	CustomerID	OrderDate	DeliveryDate	Amount
10248	VINET	2012/07/04	2012/07/16	428.0
10249	TOMSP	2012/07/05	2012/07/10	1842.0
10250	HANAR	2012/07/08	2012/07/12	1523.5
10251	VICTE	2012/07/08	2012/07/15	624.95
10252	SUPRD	2012/07/09	2012/07/11	3559.5
…	…	…	…	…

✦ 1. Calculate the single value of date and time

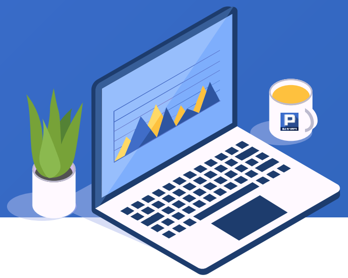


The symbol ” -” is used to calculate the number of days between the two dates. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Orders")	/Read order table
3	=A2.select(year(OrderDate)==2014 && DeliveryDate-OrderDate>30)	/Use the symbol "-" to calculate the number of days between the delivery date and the order date

A3	ID	CustomerID	OrderDate	DeliveryDate	Amount
	10924	BERGS	2014/03/04	2014/04/08	1835.7
	10927	LACOR	2014/03/05	2014/04/08	800.0
	10970	BOLID	2014/03/24	2014/04/24	224

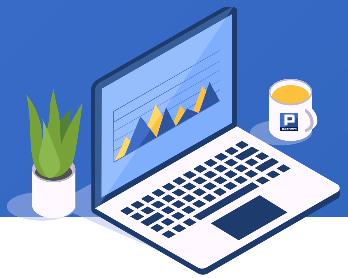
1. Calculate the single value of date and time



For each ID, when the first time 1 appears in value, start to accumulate until 0 appears. If 0 does not appear, it is compared with the system time. Part of the data are as follows:

ID	Time	Value
1	2020/07/08 15:00:00	1
1	2020/07/08 15:02:00	1
1	2020/07/08 15:04:00	1
1	2020/07/08 15:06:00	0
1	2020/07/08 15:08:00	0
1	2020/07/08 15:10:00	1
1	2020/07/08 15:20:00	0
2	2020/07/08 15:02:00	1

✦ 1. Calculate the single value of date and time

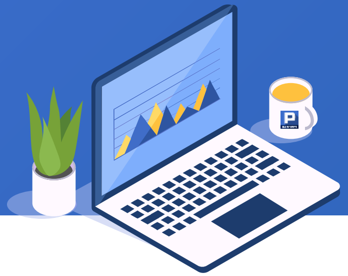


The function now() gets the date and time of the system at the moment. The function interval (datetimeExp1,datetimeExp2) calculates the interval between two date time type data, and the option @s returns the number of seconds of the difference. SPL is as follows:

	A	B
1	=file("table.txt").import@t()	/Read the file
2	=A1.group(ID).(~.group@o1(Value)|[null])	/Group by ID, and then merge and group according to the same value of adjacent values in each group. Take the first record of each small group. For the convenience of subsequent calculation, each large group makes up a null
3	=A2.news(~.len()\2;ID,(s=A2.~(#*2-1).Time):StartTime, interval@s(s,ifn(A2.~(#*2).Time,now()))/60:CumTime)	/The time of each group of odd rows is taken as the start time, and the time of corresponding even rows is taken as the end time to calculate the interval duration. If the corresponding even row is null, the current system time is taken as the end time to calculate the interval duration

A3	ID	StartTime	CumTime
	1	2020/07/08 15:00:00	6.0
	1	2020/07/08 15:10:00	10.0
	2	2020/07/08 15:02:00	28.0

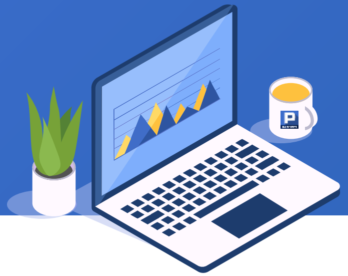
✦ 1. Calculate the single value of date and time



The current date is 2020/02/17. Calculate the rise rate of the Shanghai composite index last week. Part of the data are as follows:

Date	Open	Close	Amount
2020/02/17	2924.9913	2983.6224	3.67E11
2020/02/14	2899.8659	2917.0077	3.08E11
2020/02/13	2927.1443	2906.0735	3.35E11
2020/02/12	2895.5561	2926.8991	2.98E11
2020/02/11	2894.5414	2901.6744	3.03E11
…	…	…	…

✦ 1. Calculate the single value of date and time

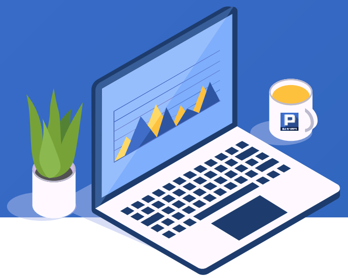


The function pdate(dateExp) is used to get the earliest day and the last day of the week / month / quarter of the specified date dateExp. SPL is as follows:

	A	B
1	=file("sh000001.csv").import@cqt()	/Read Shanghai composite index data
2	=A1.sort(Date)	/Sort by date
3	=pdate@w(A2.m(-1).Date)	/Use the @w option of the pdate() function to select the first day (Sunday) of the week of the day (2020/ 02/17)
4	=A2.select@z1(Date<=A3-2)	/Find the first record before last Friday from back to front, that is, the last record of the previous trading week
5	=pdate@w(A4.Date)	/Find the first day of the previous trading week (Sunday)
6	=A2.select@z1(Date<=A5-2)	/Find the first record of the previous Friday before last trading week from back to front, that is, the last record of the previous two trading week
7	=A4.Close/A6.Close-1	/Calculate rise rate

A7	Value
	0.01427

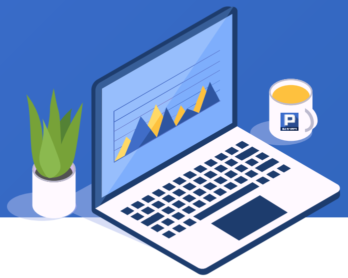
✦ 1. Calculate the single value of date and time



Calculate the average daily sales for each quarter in 2014. Part of the data in the sales table are as follows:

ORDERID	CUSTOMERID	EMPLOYEEID	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. Calculate the single value of date and time

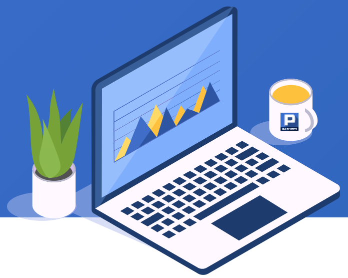


The function days(dateExp) gets the number of days in the year, quarter or month of the specified date dateExp. The option @q is used to get the number of days in the quarter of the specified date. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Sales")	/Read sales table
3	=A2.select(year(OrderDate)==2014)	/Select 2014 records
4	=A3.groups((month(OrderDate)+2)\3:Quarter; sum(Amount):Amount)	/Group and aggregate by quarter, and calculate the total sales of each quarter
5	=A4.run(Amount=Amount / days@q(date("2014/"/(Quarter*3)+"/01")))	/Use the days() function to calculate the days of each quarter, and then divide the total sales by the days to calculate the average daily sales

A5	Quarter	Amount
	1	1765.33
	2	1764.96
	3	2034.56
	4	2355.63

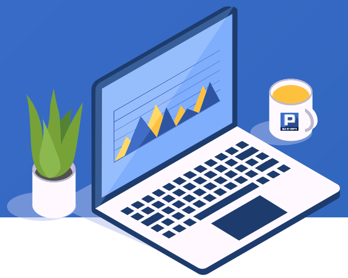
✦ 1. Calculate the single value of date and time



Query the average age of employees in each department. The employee table is as follows:

ID	NAME	BIRTHDAY	HIREDATE	DEPT	SALARY
1	Rebecca	1974/11/20	2005/03/11	R&D	7000
2	Ashley	1980/07/19	2008/03/16	Finance	11000
3	Rachel	1970/12/17	2010/12/01	Sales	9000
4	Emily	1985/03/07	2006/08/15	HR	7000
5	Ashley	1975/05/13	2004/07/30	R&D	16000
…	…	…	…	…	…

✦ 1. Calculate the single value of date and time

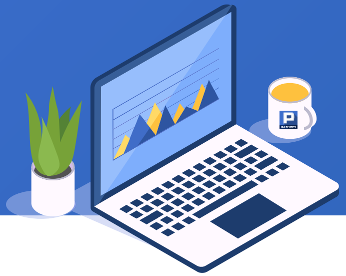


The function age(x) is used to calculate the number of years from x to the current date. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Employee")	/Read employee table
3	=A1.groups(DEPT; avg(age(BIRTHDAY)):AvgAge)	/Group summary to calculate of the average age of each department. The age() function is used to calculate the age of employees

A3	DEPT	AvgAge
	Administration	43.5
	Finance	38.83
	HR	41.05
	…	…

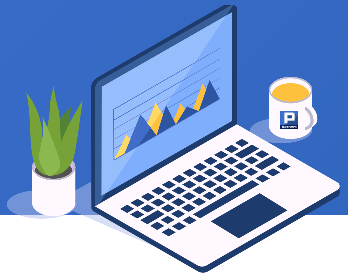
1. Calculate the single value of date and time



Query the total sales amount in the three months before May 21, 2014, and part of the data in the sales table are as follows:

ORDERID	CUSTOMERID	EMPLOYEEID	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. Calculate the single value of date and time

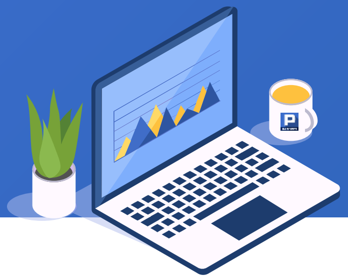


The function elapse(dateExp, n) calculates the new date with a certain time difference. When n is a negative number, it means the new date before n days / years / months. The option @m is used to calculate new date that differs by n months from the specified date. SPL is as follows:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Sales")	/Read sales table
3	=date("2014/05/21")	/Define date
4	=A2.select(OrderDate>=elapse@m(A3,-3) && OrderDate<A3)	/Use the elapse() function to calculate the date 3 months before the A3 date. Select the data for the previous three months from the sales data.
5	=A4.sum(Amount)	/Calculate the total sales amount

A5	Value
	154074.49

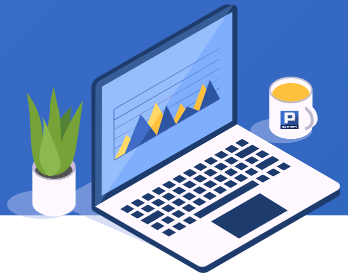
✦ 1. Calculate the single value of date and time



Calculate how many times that employees had not solved customer problems for more than 10 working days for each employee in 2014. Part of the data are as follows:

ID	CustomerID	EmployeeId	QuestionDate	SolveDate
1	OLDWO	2	2014/01/01	2014/01/09
2	WELLI	7	2014/01/01	2014/01/07
3	LAUGB	2	2014/01/01	2014/01/07
4	LINOD	8	2014/01/02	2014/01/08
5	REGGC	5	2014/01/02	2014/01/12
…	…	…	…	…

✦ 1. Calculate the single value of date and time

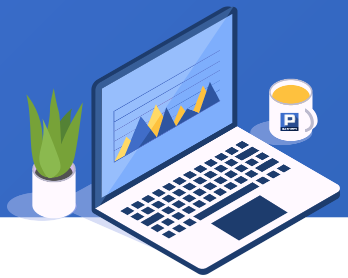


The function workday (t, k, h) calculates the date k working days away from the date t. h is a (non) holiday sequence, that is, if a member in h is not weekend, it is a holiday, if it is weekend, it is not a holiday; if it is weekend, it is calculated according to working day. SPL is as follows:

	A	B
1	=file("AfterSale.csv").import@ct()	/Import AfterSale table
2	[2014/01/01,2014/01/26,2014/01/31,2014/02/03,2014/02/04,2014/02/05,2014/02/06,2014/02/08,2014/04/07,2014/05/01,2014/05/02,2014/05/04,2014/06/02,2014/09/08,2014/09/28,2014/10/01,2014/10/02,2014/10/03,2014/10/06,2014/10/07,2014/10/11]	/Defining 2014 holidays
3	=A1.select(year(QuestionDate)==2014 && workday(QuestionDate, 10, A2) < SolveDate)	/Use the workday() function to calculate the date after 10 working days and remove the holidays
4	=A3.groups(EmployeeId; count(~):Count)	/Group and aggregate by employee and count times

A4	EmployeeID	Count
	1	2
	2	1
	3	2
	…	…

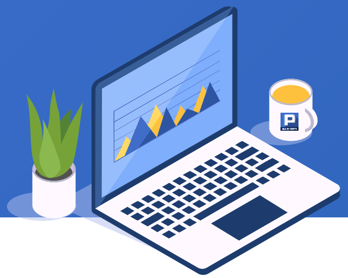
2. Calculate date time sequence



List the names of personnel on duty in each working day from 2020/04/27 to 2020/05/08. Part of the contents of attendance table are as follows:

ID	Date	Name
1	2020/04/27	Emily
2	2020/04/28	Emily
3	2020/04/28	Johnson
4	2020/04/29	Emily
5	2020/04/30	Johnson
…	…	…

✦ 2. Calculate date time sequence

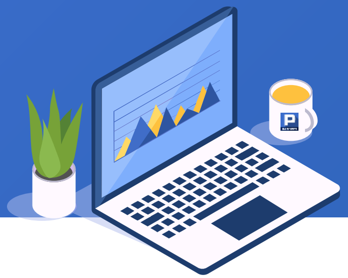


The function workdays(b, e, h) calculates the sequence of working days between date b and date e, including b and e. h is a (non) holiday sequence, that is, if a member in h is not weekend, it is a holiday, if it is weekend, it is not a holiday; if it is weekend, it is calculated according to working day. SPL is as follows:

	A	B
1	[2020/04/27,2020/05/08]	/Define start and end dates
2	=workdays(A1(1),A1(2),[date("2020/05/01")])	/The workdays() function calculates the working days in the interval, excluding the holiday May 1
3	=file("Duty.txt").import@t()	/Import duty table
4	=A3.align@a(A2, Date)	/The duty table is grouped in alignment with the working day sequence, and match all records in each group
5	=A4.new(~.Date:Date, ~.(Name).concat@c():Names)	/Create a sequence table, and concatenate the names of each group with commas

A5	Date	Names
	2020/04/27	Emily
	2020/04/28	Emily,Johnson
	2020/04/29	Emily
	…	…

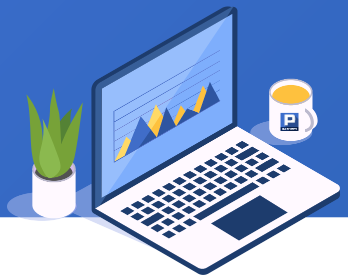
✦ 2. Calculate date time sequence



When overlapping parts are not counted repeatedly, calculate the total number of days contained in multiple time periods. Part of the data are as follows:

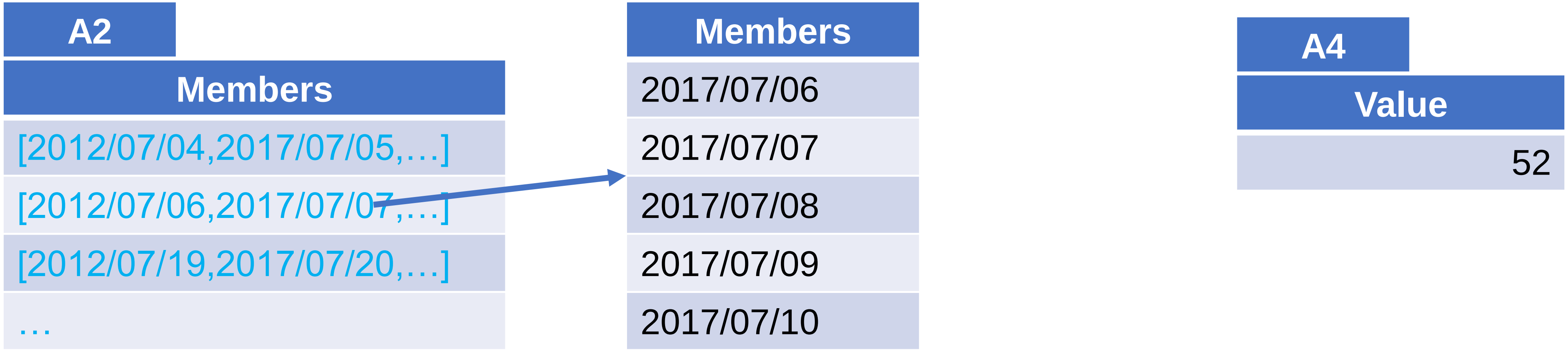
ID	Start	End
1	2012/07/04	2012/07/16
2	2012/07/06	2012/07/10
3	2012/07/19	2012/07/24
4	2012/07/22	2012/07/25
5	2012/07/30	2012/08/02
…	…	…

✦ 2. Calculate date time sequence



The function periods(s, e, i) is used to return a sequence of time values of each interval i from s to e (including endpoints). The default unit is day, and i is 1 by default. SPL is as follows:

	A	B
1	=file("periods.txt").import@t()	/Read periods from file
2	=A1.(periods(Start,End))	/Calculate the dates contained in each period in loop
3	=A2.union()	/Find the union of dates
4	=A3.len()	/Calculation the number of days



A2

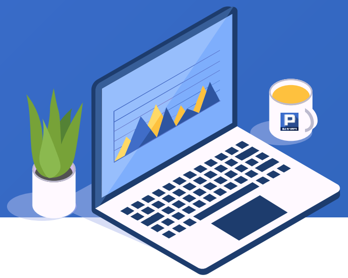
Members
[2012/07/04,2017/07/05,…]
[2012/07/06,2017/07/07,…]
[2012/07/19,2017/07/20,…]
…

Members
2017/07/06
2017/07/07
2017/07/08
2017/07/09
2017/07/10

A4

Value
52

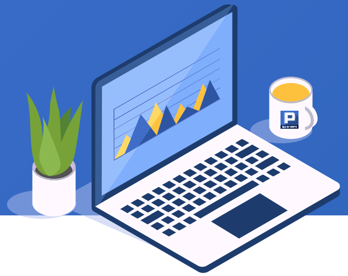
2. Calculate date time sequence



Sales records from January 20, 2014 to January 20, 2015 (not included) are equally divided into 4 groups according to the date and stored in the files respectively. Some data in the sales table are as follows:

ORDERID	CUSTOMERID	EMPLOYEEID	ORDERDATE	AMOUNT
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
10404	MAGAA	2	2014/01/03	1675.0
…	…	…	…	…

✦ 2. Calculate date time sequence



The function range (s,e,k:n) divides the interval between s and e equally into n parts, returns the header of k and k+1 segments, and returns two sequences. SPL is as follows:

	A	B	C
1	=file("Sales.txt").import@qt()		/Import sales table
2	[2014/01/20,2015/01/20]		/Define start and end dates
3	for 4	=range(A2(1),A2(2), A3:4)	/Calculate in loop, use the range function to divide the date interval into 4 parts, and return the A3 part each time
4		=A1.select(B3(1)<=OrderDate && OrderDate<B3(2))	/Select records according to the subinterval of each date
5		=file("Sales"+string(B3(1), "yyyyMMdd")+".txt").export@t(B4)	/Create file and export the records selected by B4

A3	Members	Members	Members	Members
	2014/01/20	2014/04/22	2014/07/22	2014/10/21
	2014/04/22	2014/07/22	2014/10/21	2015/01/20

THANKS