SPL Base

esProc Tutorial

Alignment Grouping

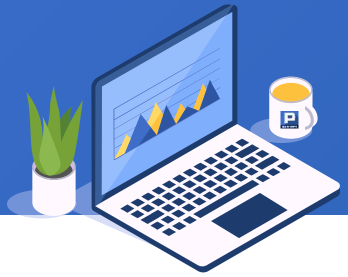
CONTENTS

CONTENTS

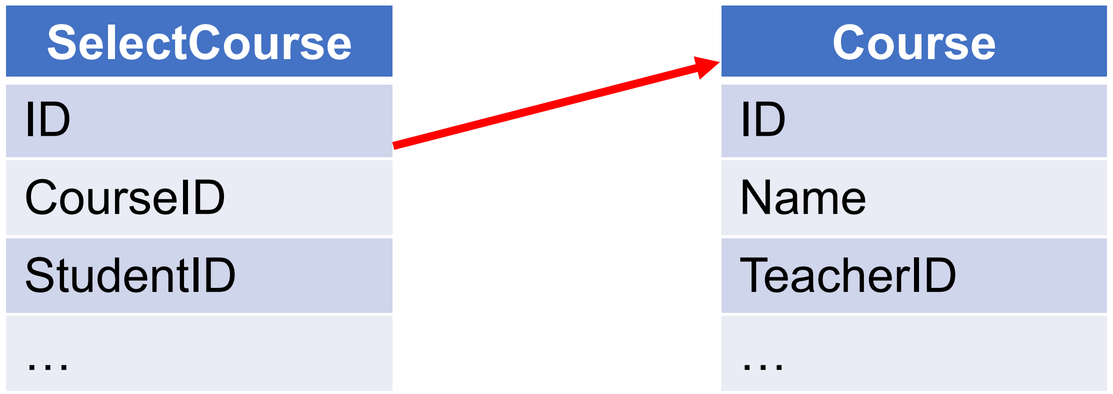
01

Alignment Grouping

✦ 1. Return the first matching member for each group



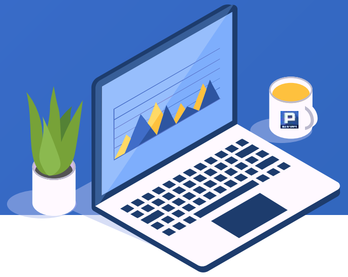
There are SelectCourse table and Course table. Find which courses are not selected and list them according to Course table’ s ID field.



SelectCourse
ID
CourseID
StudentID
…

Course
ID
Name
TeacherID
…

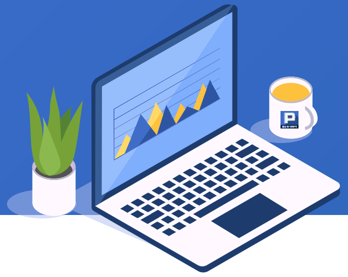
✦ 1. Return the first matching member for each group



The SPL script uses align(A:x, y) function to implement the alignment grouping:

	A	B
1	=connect("db")	/Connect to the database
2	=A1.query("select * from SelectCourse")	/Get data from *SelectCourse* table
3	=A1.query("select * from Course")	/Get data from *Course* table
4	=A2.align(A3:ID,CourseID)	/Group *SelectCourse* table by *Course* table’ s ID field and return the first member of each group
5	=A3(A4.pos@a(null))	/Get records of *Course* table where the course isn’ t selected (the corresponding value in the grouped *SelectCourse* table is null)

✦ 1. Return the first matching member for each group



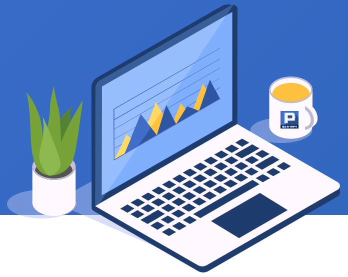
A4

Members
(null)
[13,2,7]
[7,3,41]
[45,4,28]
[3,5,52]
[1,6,59]
[10,7,13]
[8,8,49]
[6,9,57]
(null)

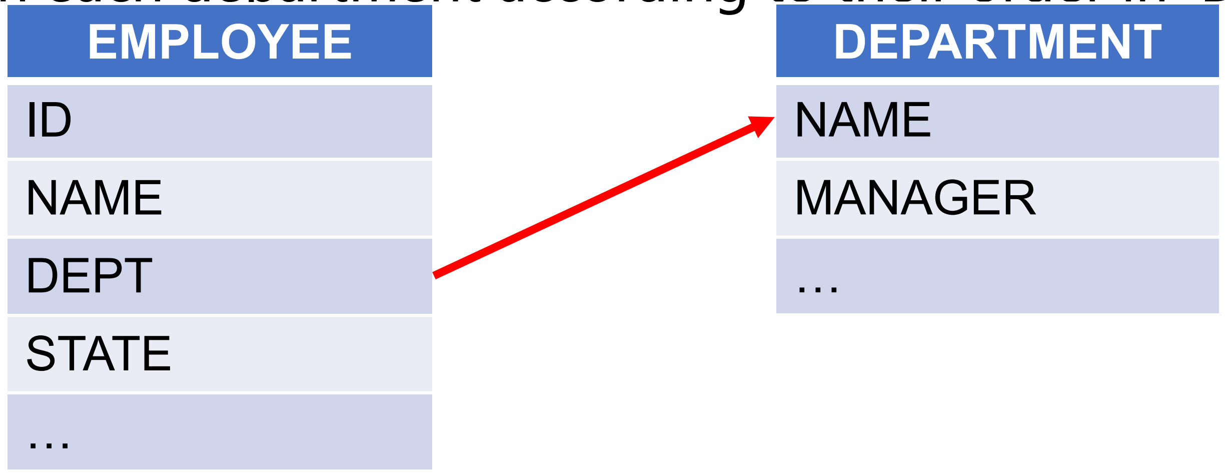
A5

ID	NAME	TeacherID
1	Environmental protection and sustainable development	5
10	Music appreciation	18

2. Return all matching members for each group



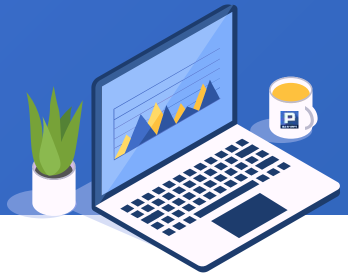
There are EMPLOYEE table and DEPARTEMENT table. Count the number of employees in each department according to their order in DEPARTMENT table.



EMPLOYEE
ID
NAME
DEPT
STATE
…

DEPARTMENT
NAME
MANAGER
…

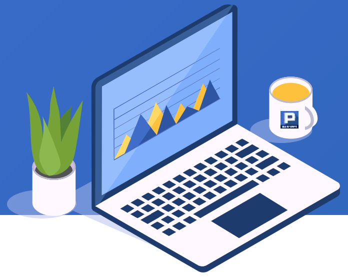
2. Return all matching members for each group



SPL performs alignment grouping using @a option:

	A	B
1	=connect("db")	/Connect to the database
2	=A1.query("select * from EMPLOYEE")	/Get data from EMPLOYEE table
3	=A1.query("select * from DEPARTMENT")	/Get data from DEPARTMENT table
4	=A2.align@a(A3:ID, DEPARTMENT)	/Group EMPLOYEE table by the order of the departments listed in DEPARTMENT table and use @a option to return all matching records
5	=A4.new(DEPT, ~.count():COUNT)	/Count the number of employees in each department

✦ 2. Return all matching members for each group



A4

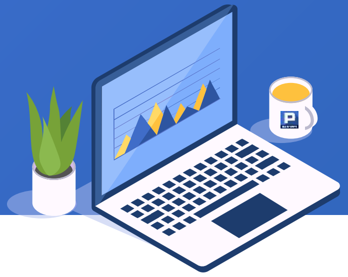
Members
[[18,Jonathan,Admin,...], [20,Alexis, Admin,...], …]
[[1,Rebecca,R&D,...],[5,Ashley,R&D,...],...]
[[3,Rachel,Sales,...],[6,Matthew,Sales,...],...]
…

ID	NAME	DEPT	STATE
1	Rebecca	R&D	California
5	Ashley	R&D	Texas
10	Ryan	R&D	Pennsylvania
…	…	…	…

A5

DEPT	COUNT
Admin	4
R&D	29
Sales	187
…	…

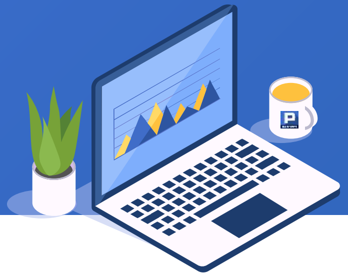
3. Put mismatching members in a separate group



Based on EMPLOYEE table, calculate the average salary according to [California, Texas, New York, Florida] and put employees in other states in a separate group.

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
…	…	…	…

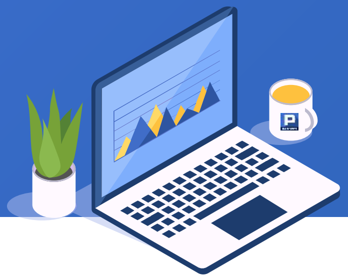
✦ 3. Put mismatching members in a separate group



In the SPL script, align() function works with @an options to perform the grouping:

	A	B
1	=connect("db")	/Connect to the database
2	=A1.query("select * from EMPLOYEE")	/Get data from EMPLOYEE table
3	[California,Texas,New York,Florida]	/Define a sequence of states
4	=A2.align@an(A3,STATE)	Group EMPLOYEE table by the order of the specified states; @a option returns all matching records for each group and @n option put mismatching records in a new group
5	=A4.new(if (#>A3.len(),"Other",STATE):STATE,~.avg(SALARY):AvgSalary)	/Calculate average salary for each group and return a new table sequence. Rename the new group Other; otherwise the field name will display as the state in the first record

✦ 3. Put mismatching members in a separate group



A4

Members
[[1,Rebecca,California,...], [6,Matthew,California,...], …]
[[4,Emily,Texas,...],[5,Ashley,Texas,...],...]
[[2,Ashley,New York,...],[12,Jessica,New York,...],...]
[[13,Daniel, Florida,...],[14,Alyssa,Florida,...],...]
[[3,Rachel,New Mexico,...],[7,Alexis,Illinois,...],...]

ID	NAME	STATE	SALARY
3	Rachel	New Mexico	9000
7	Alexis	Illinois	9000
10	Ryan	Pennsylvania	13000
19	Samantha	Pennsylvania	10000
…	…	…	…

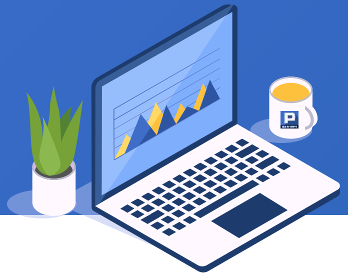
A5

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

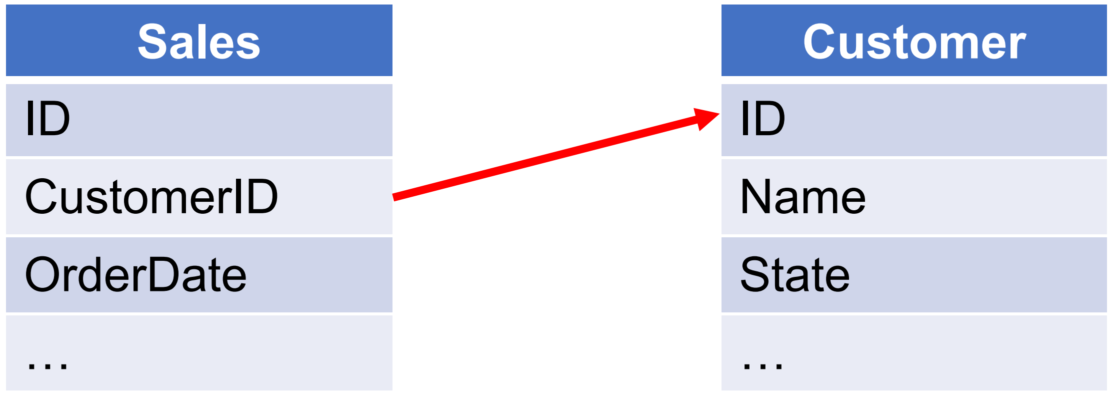
CONTENTS

Sequence-number-based Alignment Grouping

✦ 1. Return the first matching member for each group



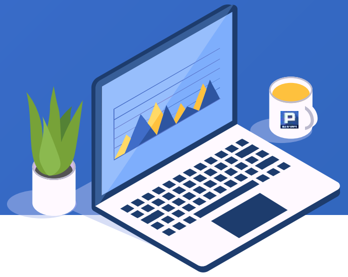
There are Sales table and Customer table. Find customers that don’ t have orders in 2014.



Sales
ID
CustomerID
OrderDate
…

Customer
ID
Name
State
…

✦ 1. Return the first matching member for each group



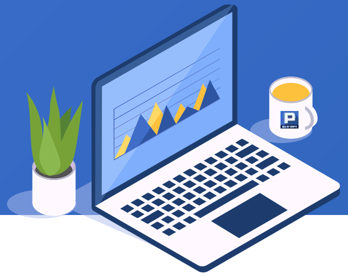
In the SPL script below, align(n, y) function is used to implement alignment grouping:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Sales")	/Get data from Sales table
3	=A1.query("select * from Customer")	/Get data from Customer table
4	=A3.(ID)	/Get ID field from Customer table
5	=A2.align(A4.len(), A4.pos(CustomerID))	/Group Sales table by aligning CustomerID to A4’ s ID
6	=A3(A5.pos@a(null))	/Get customer records that don’ t have orders (whose corresponding value in the grouped Sales table are null)

A6

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

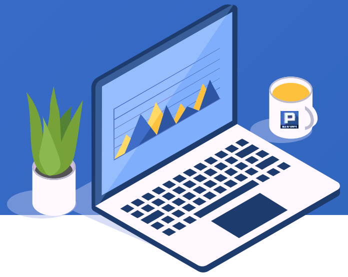
✦ 2. Return all matching members for each group



Based on Orders table, list the orders count in every month of 2013 in order.

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

✦ 2. Return all matching members for each group



align(n, y) function uses @a option to return all matching records for each group:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Orders where year(OrderDate)=2013")	/Get records of 2013 from Orders table
3	=A2.align@a(12,month(OrderDate))	/Divide the selected orders records into 12 groups by months; @a option returns all matching records for each group
4	=A3.new(#:Month,~.count():OrderCount)	/the orders count for each month

2. Return all matching members for each group

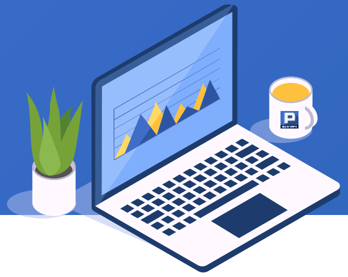
A3

Members
[[10400,EASTC,1],[10401,RATTC,1],…]
[[10433,PRINI,2],[10434,FOLCO,2],…]
[[10462,CONSH,3],[10463,SUPRD,3],…]
[[10492,BOTTM,4],[10493,LAMAI,4],…]
[[10523,SEVES,5],[10524,BERGS,5],…]
[[10555,SAVEA,6],[10556,SIMOB,6],…]
[[10585,WELLI,7],[10586,REGGC,7],…]
[[10618,MEREP,8],[10619,MEREP,8],…]
[[10651,WANDK,9],[10652,WANDK,9],…]
[[10688,VAFFE,10],[10689,BERGS,10],…]
[[10726,EASTC,11],[10727,REGGC,11],…]
[[10760,MAISD,12],[10761,RATTC,12],…]

A4

Month	OrderCount
1	33
2	29
3	30
4	31
5	32
6	30
7	33
8	33
9	37
10	38
11	34
12	48

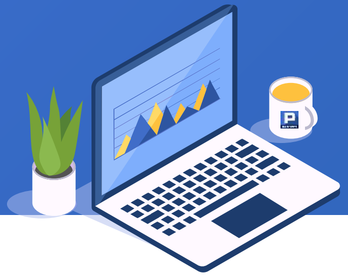
✦ 3. Put each member into groups by a corresponding sequence



Group PostRecord table by label 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
…	…	…	…

✦ 3. Put each member into groups by a corresponding sequence



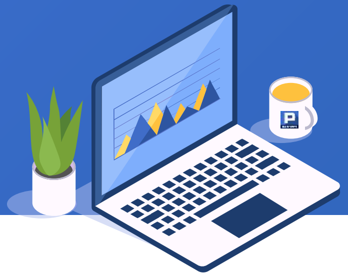
In the SPL script, align(n, y) function uses @r option to put a record in multiple corresponding group.

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from PostRecord")	/Get data from PostRecord table
3	=A2.conj(Label.split(",")).id()	/Split each Label value by comma and union them into one sequence to get all unique labels
4	=A2.align@ar(A3.len(),A3.pos(Label.split(",")))	/@r option put each post into groups according to its labels
5	=A4.new(A3(#):Label,~.count():Count).sort@z(Count)	/Count the posts under each label and arrange the result in descending order

A5

Label	Count
SPL	7
SQL	6
Basics	5
…	…

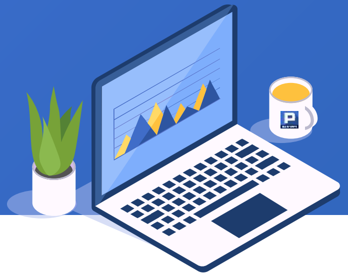
4. Segmental function



Group EMPLOYEE table by salary ranges (<8000, ≥8000&≤12000, and >12000) and count employees in each group.

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. Segmental function



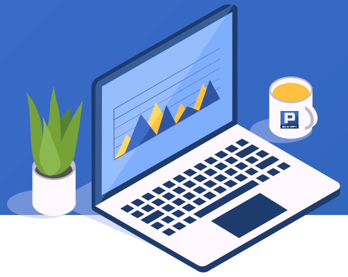
In the SPL script, align(n,y) function uses pseg(x) function to divide records into segments:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from EMPLOYEE")	/Get data from EMPLOYEE table
3	[0,8000,12000]	/Define salary ranges
4	=A2.align@a(A3.len(),A3.pseg(SALARY))	/pseg() function locates the range for a salary
5	=A4.new(A3 (#):SALARY,~.count():COUNT)	/Count employees in each group

A5

SALARY	COUNT
0	308
8000	153
12000	39

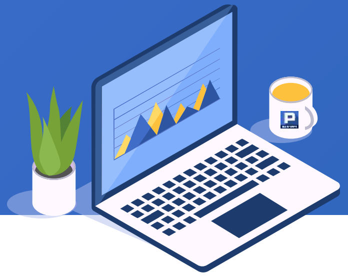
4. Segmental function



Group EMPLOYEE table by hire date (<10, ≥10&≤20, and >20) and calculate average salary for each group.

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. Segmental function



In the SPL script, align(n,y) function uses pseg(x,y) function to divide records into segments:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from EMPLOYEE")	/Get data from EMPLOYEE table
3	[0,10,20]	/Define hire date ranges
4	=A2.align@a(A3.len(),A3.pseg(year(now())-~,year(HIREDATE)))	/pseg() function gets the range containing a certain hire date
5	=A4.new(A3(#):EntryYears,~.avg(SALARY):AvgSalary)	/Calculate average salary for each group

A5

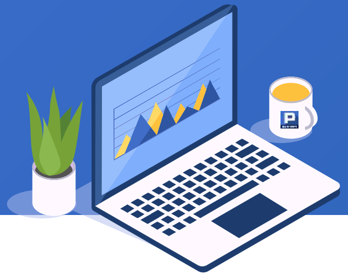
EntryYears	AvgSalary
0	6807.69
10	7417.78
20	7324.32

CONTENTS

03

Enumeration Grouping

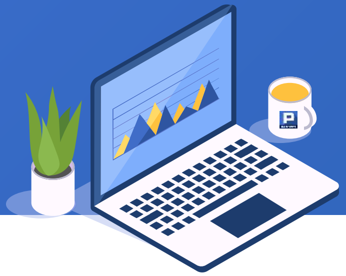
✦ 1. Put each member into the first matching group



Group UrbanPopulation table by population ranges.

ID	City	Population	Province
1	Shanghai	12286274	Shanghai
2	Beijing	9931140	Beijing
3	Chongqing	7421420	Chongqing
4	Guangzhou	7240465	Guangdong
5	Hong Kong	7010000	Hong Kong Special Administrative Region
…	…	…	…

✦ 1. Put each member into the first matching group



The SPL script uses enum() function to perform enumeration grouping:

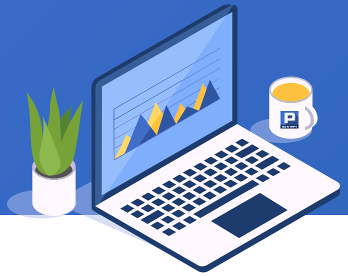
	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from UrbanPopulation")	/Get data from UrbanPopulation
3	[?>2000000,?>1000000,?>500000,?<=500000]	/Megacity: >two million, Super city: >one million & <two million， Big city: >half million & < one million, Other cities
4	=A2.enum(A3,Population)	/Perform enumeration grouping by A3' s conditions

A4

Members
[[1,Shanghai,12286274,...], [2,Beijing, 9931140,...], …]
[[28,Changsha,1965282,...], [29,Nanchang,1900817,...], …]
[[69,Huainan,974026,...], [70,Haikou, 967336,...], …]
[]

ID	City	Population	Province
1	Shanghai	12286274	Shanghai
2	Beijing	9931140	Beijing
3	Chongqing	7421420	Chongqing
…	…	…	…

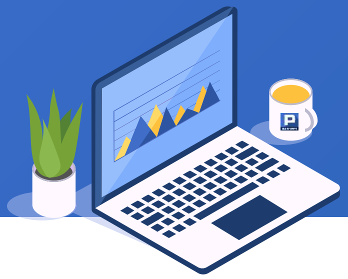
2. Put mismatching members in a separate group



Group EMPLOYEE table by age groups and calculate average salary for each group.

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
…	…	…	…

2. Put mismatching members in a separate group



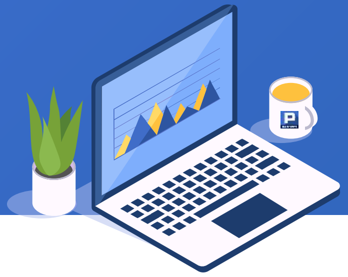
The enum() function works with @n option to put mismatching records in a separate group:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from EMPLOYEE")	/Get data from EMPLOYEE table
3	[?<35,?<45]	/Define two age groups: <35 & <45
4	=A2.enum@n(A3, year(now())-year(BIRTHDAY))	/Calculate ages by birthdays and perform enumeration grouping by the specified age groups and put mismatching records into a new group
5	=A4.new(if (#>A3.len(), "Other",A3(#)):AGE,~.avg(SALARY):AvgSalary)	/Other Name the new group Other and calculate average salary in each group

A5

AGE	AvgSalary
?<35	7234.69
?<45	7440.65
Other	7367.05

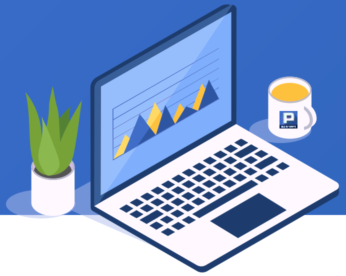
✦ 3. Match each member to multiple groups



Based on GDP table, calculate average GDP for direct-controlled municipalities, first-tier cities and second-tier cities. One record may match multiple conditions. Beijing, for example, is both a direct-controlled municipality and a first tier city.

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. Match each member to multiple groups



enum() function uses @r option to match each record to multiple conditions:

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from GDP")	/Get data from 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]	/List direct-controlled cities, first-tier cities and second-tier cities
4	=A2.enum@r(A3,City)	/Group records by the listed groups
5	=A4.new(A3(#):Area,~.sum(GDP)/~.sum(Population)*10000:CapitaGDP)	/Calculate average GDP for 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

THANKS

for your attention