SET

CONTENTS

01

Set operation

where

```
select * from EMPLOYEE where GENDER=“M”
and DEPT=“HR”;
```

```
select * from EMPLOYEE where GENDER="M"
and DEPT!="HR";
```

```
select * from EMPLOYEE where GENDER="M"
or DEPT="HR";
```

Set operation

```
select * from EMPLOYEE where GENDER="M"
intersect
select * from EMPLOYEE where DEPT="HR";
```

Intersect

```
select * from EMPLOYEE where GENDER="M"
minus
select * from EMPLOYEE where DEPT="HR";
```

Minus

```
select * from EMPLOYEE where GENDER="M"
union
select * from EMPLOYEE where DEPT="HR";
```

Union

```
select * from EMPLOYEE where GENDER="M"
uninon all
select * from EMPLOYEE where DEPT="HR";
```

Union all

The Basic Operations of Sets-Intersect, Minus and Union

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=A1.select(GENDER=="M")	
3	=A1.select(DEPT=="HR")	
4	=A1.select(GENDER=="M"&&DEPT=="HR")	=A2^A3
6	=A1.select(GENDER=="M"&&DEPT!="HR")	=A2\A3
8	=A1.select(GENDER=="M"||DEPT=="HR")	=A2&A3

Conditional Query

Set operations

02

Set orientation and Discreteness

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=A1(1)	/The first record
3	=A2.NAME="Jim"	/Modify the NAME field of the first record

A1 when A1 executed

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY
1	1	Rebecca	Moore	F	California	1974-11-20
2	2	Ashley	Wilson	F	New York	1980-07-19

A1 after A3 executed

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY
1	1	Jim	Moore	F	California	1974-11-20
2	2	Ashley	Wilson	F	New York	1980-07-19

A2 when A2 executed

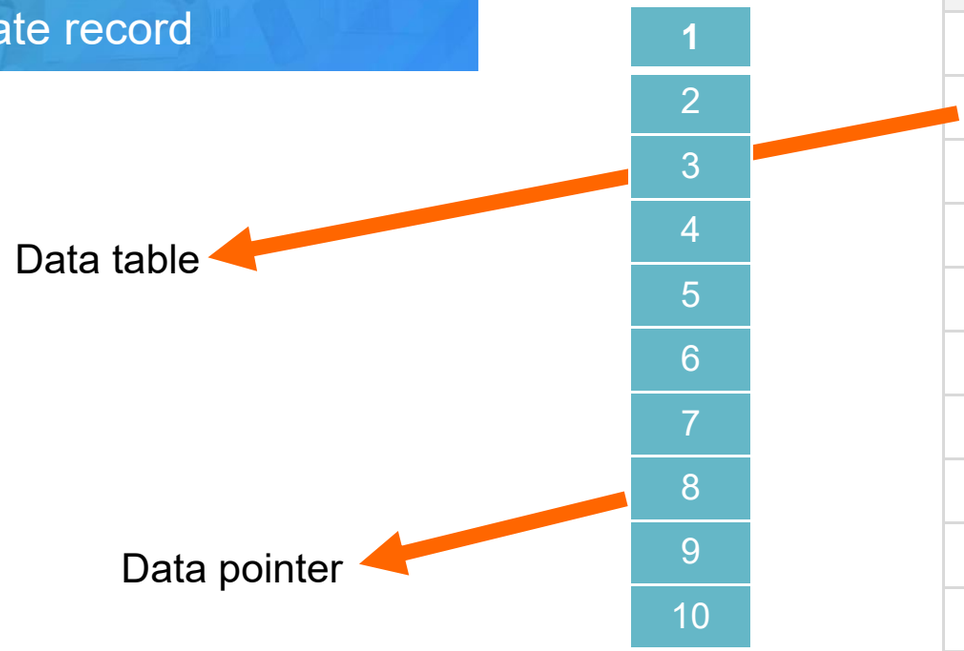
EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE
1	Rebecca	Moore	F	California	1974-11-20	2005-03-11

A2 after A3 executed

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE
1	Jim	Moore	F	California	1974-11-20	2005-03-11

A2 is the pointer of the first record in set A1 , and it is not the real data, when the data in A1 changes, A2 changes accordingly.

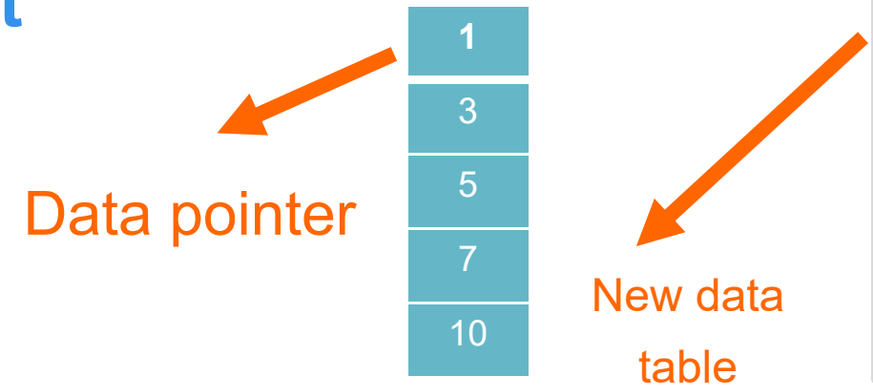
Understanding discreteness-separate record



NAME	GENDER	AGE
Rebecca	F	44
Ashley	M	38
Rachel	F	48
Emily	F	34
Ashley	M	43
Matthew	M	34
Alexis	F	46
Megan	F	39
Victoria	F	35
Ryan	M	43

SPL： A.select(AGE>40) **Statement** **SQL：** select * from A where AGE>40

Result



NAME	GENDER	AGE
Rebecca	F	44
Rachel	F	48
Ashley	M	43
Alexis	F	46
Ryan	M	43

Understanding discreteness-separate record

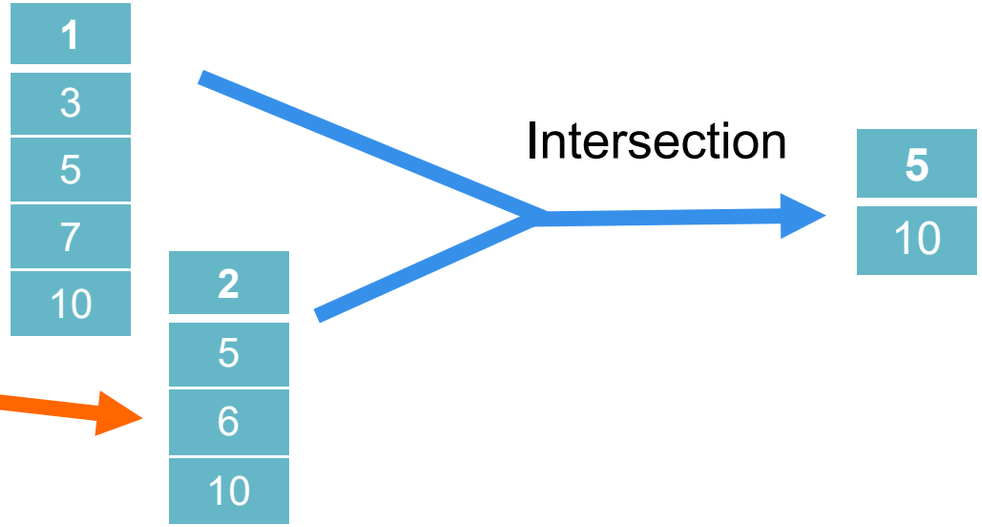
Intersection operation of SPL

Data table

Data pointer

	NAME	GENDER	AGE
1	Rebecca	F	44
2	Ashley	M	38
3	Rachel	F	48
4	Emily	F	34
5	Ashley	M	43
6	Matthew	M	34
7	Alexis	F	46
8	Megan	F	39
9	Victoria	F	35
10	Ryan	M	43

	A
1	=file("E:/txt/name.txt").import@t()
2	=A1.select(AGE>40)
3	=A1.select(GENDER=="M")
4	=A2^A3



Set Operations of SQL


```
select * from EMPLOYEE where GENDER="M" and DEPT="HR";
select * from EMPLOYEE where GENDER="M" and DEPT!="HR";
select * from EMPLOYEE where GENDER="M" or DEPT="HR";
```

```
select * from EMPLOYEE where GENDER="M"
intersect
select * from EMPLOYEE where DEPT="HR";
```

Intersect

```
select * from EMPLOYEE where GENDER="M"
minus
select * from EMPLOYEE where DEPT="HR";
```

Minus

```
select * from EMPLOYEE where GENDER="M"
union
select * from EMPLOYEE where DEPT="HR";
```

Union

```
select * from EMPLOYEE where GENDER="M"
union all
select * from EMPLOYEE where DEPT="HR";
```

Union all

Where statement advocated by SQL

The intersection, difference and union operations of SPL are just to compare record pointers, while the intersection, difference and union operations of SQL are to compare full records and relatively slow. This is why SQL advocates using the form above and UNION ALL.

Calculate the age difference and salary difference between Jim and Lucy

SQL

```
SELECT (SELECT age FROM employee WHERE name='Jim' ) – ( SELECT age FROM employee WHERE name='Lucy' ) FROM dual
SELECT (SELECT salary FROM employee WHERE name=' Jim' ) – ( SELECT salary FROM employee WHERE name=' Lucy' ) FROM dual
```

SPL

	A	B
1	=file("E:/txt/employees.txt").import@t()	
2	=A1.select@1(name==“Jim”)	/Select record of Jim
3	=A1.select@1(name==“Lucy”)	/Select record of Lucy
4	=A2.age-A3.age	/Calculate age difference
5	=A2.salary-A3.salary	/Calculate salary difference
6	/Query results can exist independently outside the set and can be used repeatedly.	

A.maxp vs A.max

Sequence	Function	Result	Description
[11,5,4,3,2,1,4,5,3]	A.maxp(~*~)	11	Returns a member that maximizes the square
	A.max(~*~)	121	Returns the maximum square value
	A.maxp(~%3)	11	Returns a member that maximizes the remainder of division 3
	A.max(~%3)	2	Returns the maximum of the remainder divided by three
	A.maxp@a(~%3)	[11,5,2,5]	Returns all sequence of members that maximize the remainder of division 3
	A.maxp@z(~%3)	5	Look from back to front
	A.maxp@za(~%3)	[5,2,511]	Look for all the members from back to front

Data

序号	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Jessica	87	88	78
3	Brianna	89	90	75
4	Emma	88	84	94
5	Zachary	75	81	85
6	Sophia	74	86	93
7	Hannah	90	76	95
8	Christopher	71	81	86
9	Sean	98	86	81
10	Tyler	87	78	93

Function	Description
A.maxp(Chinese)	Return the highest record of Chinese
A.max(Chinese)	Returns the maximum value of Chinese
A.maxp@a(Chinese)	Returns all the highest records of Chinese

Name	Math	Chinese	English
Natalie	84	90	84

值
90

序号	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Brianna	89	90	75

Calculate the natural days between the last lowest price and the earliest highest price of the stock 600036 in 2017.

```
with t as (select *, row_number() over(order by tdate) rn from stktrade
where sid='600036' and tdate between '2017-01-01' and '2017-12-31'),
t1 as (select * from t where close=(select min(close) from t)),
t2 as (select * from t where close=(select max(close) from t)),
t3 as (select * from t1 where rn=(select max(rn) from t1)),
t4 as (select * from t2 where rn=(select min(rn) from t2))
select abs(datediff(t3.tdate,t4.tdate)) inteval
from t3,t4;
```

Calculate the natural days between the last lowest price and the earliest highest price of the stock 600036 in 2017.

	A
1	=connect("mysql")
2	=A1.query@x("select * from stktrade where sid='600036'and tdate between'2017-01-01'and'2017-12-31'order by tdate")
3	=A2.minp@z(close)
4	=A2.maxp(close)
5	=abs(A3.tdate-A4.tdate)

A3: Look from back to front for the record of close's first minimum

A4: Find the record of close's first maximum from front to back

03

Genericity of Sets

	A	B
1	[1,a3,2,5.4,$[4.5],2011-8-8]	/Sequence
2	=[A1,4]	/Expression

The results of A1 and A2 are as follows:

Index	Member
1	1
2	a3
3	2
4	5.4
5	$[4.5]
6	2011-08-08

Index	Member
1	[1,a3,2, ...]
2	4

Double click

Index	Member
1	1
2	a3
3	2
4	5.4
5	$[4.5]
6	2011-08-08

Is it meaningless for the actual business?

Count the total number of women in employees and family members.

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=demo.query("select * from FAMILY")	
3	=A1|A2	/Merge records to form sequence
4	=A3.count(left(GENDER,1)=="F")	/Count the total number of women in employees and family members.

A1~A4 results:

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000

Index	Member
1	[1,Rebecca,Moore, ...]
2	[2,Ashley,Wilson, ...]
3	[3,Rachel,Johnson, ...]

Value
275

Index	EID	NAME	RELATION	GENDER	AGE
1	1	Jacky	child	Male	15
2	1	Linda	child	Female	17
3	1	Vincent	spouse	Male	52

Index	Member
498	[498,Daniel,Smith, ...]
499	[499,Nicole,Smith, ...]
500	[500,Joseph,Smith, ...]
501	[1,Jacky,child, ...]
502	[1,Linda,child, ...]
503	[1,Vincent,spouse, ...]

The arbitrariness of set members also allows the set itself to be a member.

	A	B
1	[[1,2,3,4,5],[1,3,5,7,9],[2,3,5,7]]	
2	=A1.conj()	/conj
3	=A1.isect()	/isect
4	=A1.(~.sum())	/sum of all colums
5	=A1.(~.(~*~))	/squaring of column elements

A2~A5 results

A2

Index	Member
1	1
2	2
3	3
4	4
5	5
6	1
7	3
8	5

A3

Index	Member
1	3
2	5

A4

Index	Member
1	15
2	25
3	17

A5

Index	Member
1	[1,4,9, ...]
2	[1,9,25, ...]
3	[4,9,25, ...]

Array can also be a member of a sequence

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=A1.select(STATE=="California")	/Select employees in California
3	=A1.select(STATE=="Indiana")	/Select employees in Indiana
4	=A1.select(STATE=="Florida")	/Select employees in Florida
5	=[A2,A3,A4]	/Place the above three arrays in the same sequence
6	=A5.(~.count())	/Calculate the number of employees in each array separately
7	=A5.(~.STATE)	/the names of the states in the first record of each array.
8	=A5.(STATE)	/the names of the states in the first record of each array
9	=A5.new(STATE,~.count():Count)	/Calculate the number of employees in each array and generate a table

A6~A9 results

A6

Index	Member
1	55
2	7
3	62

A7

Index	Member
1	California
2	Indiana
3	Florida

A8

Index	Member
1	California
2	Indiana
3	Florida

A9

Index	STATE	Count
1	California	55
2	Indiana	7
3	Florida	62

Replace field values with records?
Find out the employees whom the age of couples is over 80

	A	B
1	=file("E:\\txt\\Employees.txt").import@t().keys(ID)	/Set ID as primary key
2	=file("E:\\txt\\EmpRel.txt").import@t()	
3	=A2.select(Relationship=="Spouse")	/Filter the spouse relationship in A2
4	>A3.switch(Emp1,A1;Emp2,A1)	/Replace both employee fields in the employee relationship table with corresponding records
5	=A3.select(age(Emp1.Birthday)+age(Emp2.Birthday)>80)	/Filter out records with the sum of ages greater than 80
6	=A5.run(~.modify(Emp1.Name:Emp1,Emp2.Name:Emp2))	/Modify the records to the Name filed of records

A3 when A6 executed

Index	Emp1	Emp2	Relationship
1	21	22	Spouse
2	Tiger	Mike	Spouse
3	Jim	Howard	Spouse
4	Ed	Lucy	Spouse

A3 when A3 executed

Index	Emp1	Emp2	Relationship
1	21	22	Spouse
2	10	1	Spouse
3	5	19	Spouse
4	16	3	Spouse

A3 after A4 executed

Index	Emp1	Emp2	Relationship
1	21	22	Spouse
2	10	1	Spouse
3	5	19	Spouse
4	16	3	Spouse

A5 when A5 executed

Index	Emp1	Emp2	Relationship
1	10	1	Spouse
2	5	19	Spouse
3	16	3	Spouse

A5 after A6executed

Index	Emp1	Emp2	Relationship
1	Tiger	Mike	Spouse
2	Jim	Howard	Spouse
3	Ed	Lucy	Spouse

Emp1 field

ID	Name	Gender	Post	Birthday	AccountNo	BasePay
21	Joe	Male	R&D Leader	1984-09-1...	528924335...	3500.0

Emp2 field

ID	Name	Gender	Post	Birthday	AccountNo	BasePay
22	Ken	Female	Sale	1982-07-1...	824387323...	3200.0

04

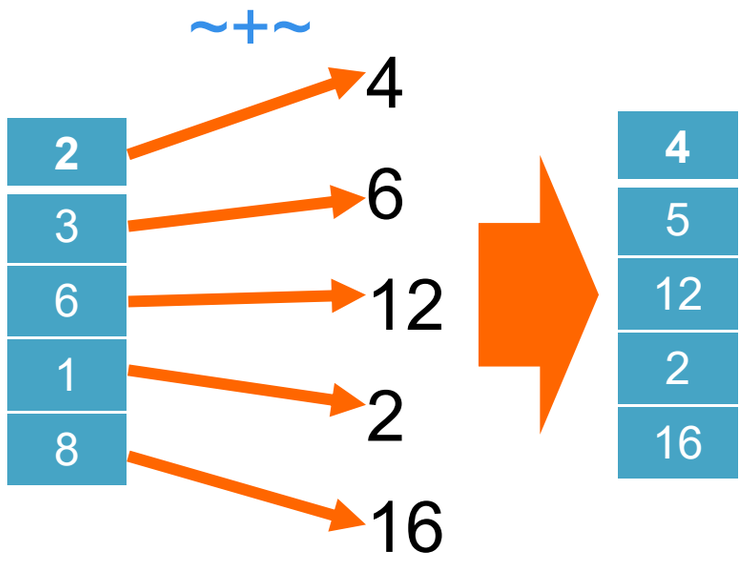
Loop for sets

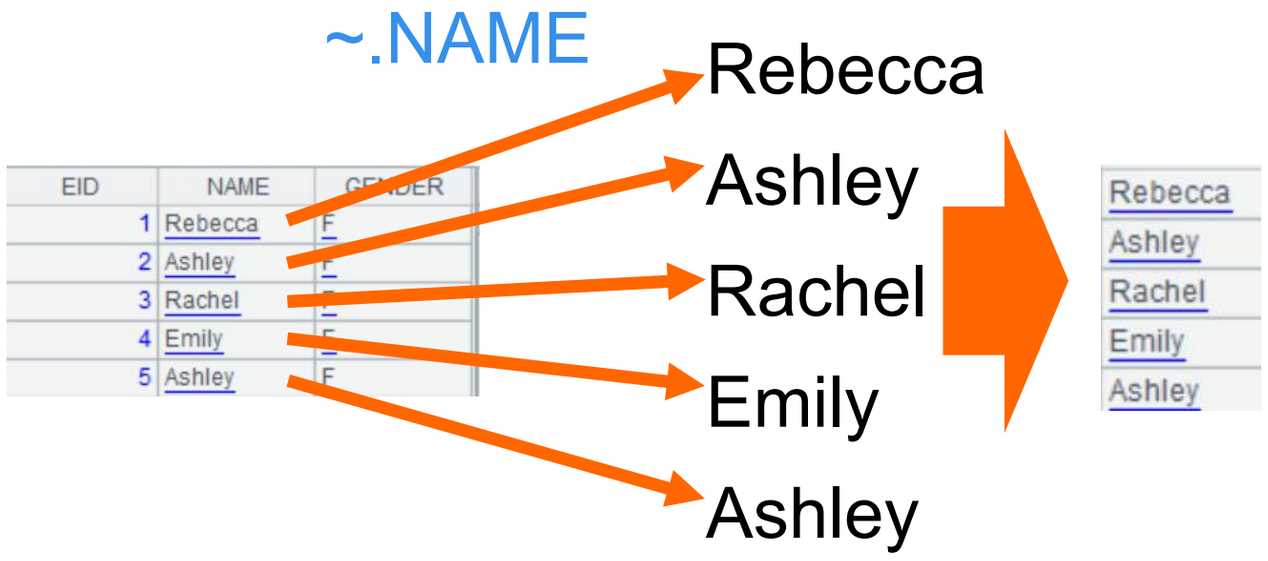
Set operations

	A	B
1	[2,3,6,1,8]	
2	=A1.sum()	/Sum 20
3	=A1.avg()	/Avg 4.0
4	=A1.median()	/Median 3
5	=A1.variance()	/Variance 6.8
6	=A1.max()	/Max 8
7	=A1.min()	/Min 1

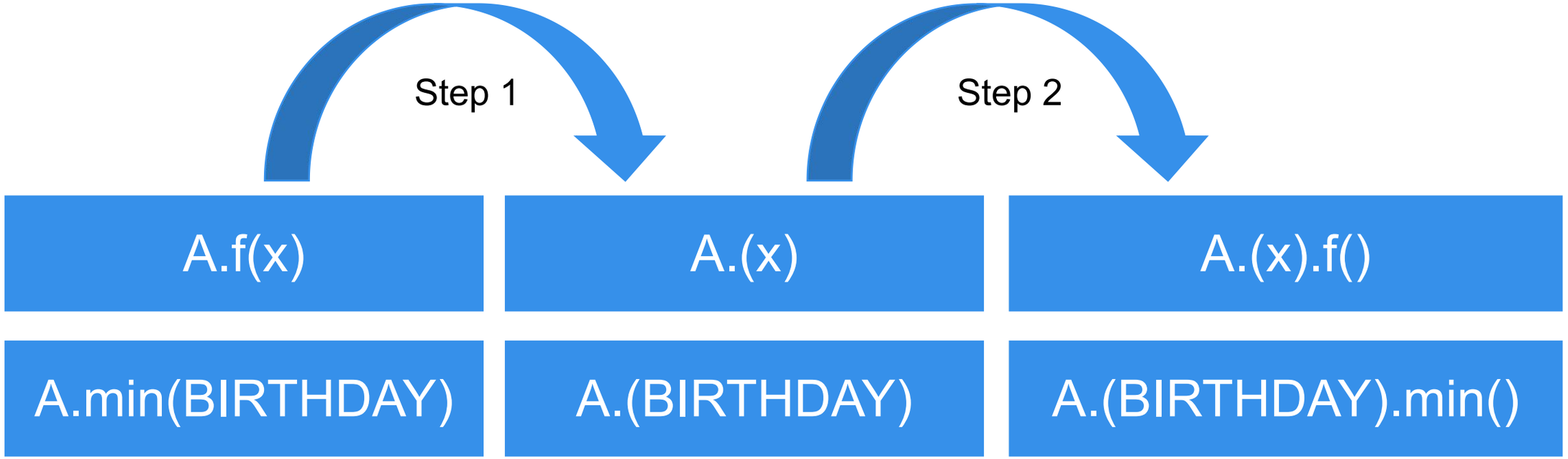
A.(X) - Calculate x for each member and return the composed sequence

	A	B
1	[2,3,6,1,8]	
2	=A1.(~+~)	/~Refer to current members
3	=demo.query("select * from EMPLOYEE")	
4	=A4.(NAME)	/Return a sequence composed of NAME fields





A.f(x)



Computed column

Aggregate calculation

Aggregation operation with parameters

	A	B
1	[2,3,6,1,8]	
2	=A1.sum(~*~)	/Sum of squares 114
3	=demo.query("select * from EMPLOYEE")	
4	=A3.min(~.BIRTHDAY)	/Min value of birthday1968-11-05
5	=A3.min(BIRTHDAY)	/Min value of birthday1968-11-05
6	=A3.avg(interval@y(BIRTHDAY,HIREDATE))	/Average length of service 27.282

A.new()—— Create new table

	A	B
1	[1,2,3,4,5]	
2	=A1.new(~:Origin,~*~:Square)	/Create new table of origin and square

Result:

Index	Origin	Square
1	1	1
2	2	4
3	3	9
4	4	16
5	5	25

Generate table from table

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=A1.new(NAME,age(BIRTHDAY):Age)	/Generate a new table with NAME and Age as fields
3	=A1.new(NAME)	/Generate a new table with NAME as the field
4	=A1.(NAME)	/Sequence composed of NAME field

A2~A4 results:

Index	NAME	Age
1	Rebecca	44
2	Ashley	39
3	Rachel	48
4	Emily	34
5	Ashley	44
6	Matthew	35
7	Alexis	46
8	Megan	40
9	Victoria	35
10	Ryan	43

Index	NAME
1	Rebecca
2	Ashley
3	Rachel
4	Emily
5	Ashley
6	Matthew
7	Alexis
8	Megan
9	Victoria
10	Ryan

Index	Member
1	Rebecca
2	Ashley
3	Rachel
4	Emily
5	Ashley
6	Matthew
7	Alexis
8	Megan
9	Victoria
10	Ryan

derive() function in addition to new()

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=A1.derive(NAME+SURNAME:FULLNAME,age(BIRTHDAY):Age)	/Add FULLNAME and Age fields

A1~A2 results:

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000
4	4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000
5	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY	FULLNAME	Age
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000	RebeccaM...	44
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000	AshleyWils...	39
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000	RachelJoh...	48
4	4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000	EmilySmith	34
5	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000	AshleySmith	44

How to construct a constant table?

	A	B	C	D
1	Natalie	M	44	173
2	Jessica	M	32	182
3	Brianna	F	26	157
4	Emma	M	43	168
5	Zachary	F	29	165
6	Sophia	F	36	170
7	=create(NAME,GENDER,AGE,HEIGHT)			
8	=A7.record([A1:D6])			

A7 when A7 executed

Index	NAME	GENDER	AGE	HEIGHT

A7 after A8 executed

Index	NAME	GENDER	AGE	HEIGHT
1	Natalie	M	44	173
2	Jessica	M	32	182
3	Brianna	F	26	157
4	Emma	M	43	168
5	Zachary	F	29	165
6	Sophia	F	36	170

Increase the age of all employees by 1

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=A1.new(NAME,age(BIRTHDAY):Age)	/Create new table with NAME and Age fields
3	=A2.run(Age=Age+1)	/Modify the Age field

A2 when A2 executed

Index	NAME	Age
1	Rebecca	44
2	Ashley	39
3	Rachel	48
4	Emily	34
5	Ashley	44

A2 after A3 executed

Index	NAME	Age
1	Rebecca	45
2	Ashley	40
3	Rachel	49
4	Emily	35
5	Ashley	45

Data are stored in documents named after year and month by month (in the form of "201801.txt" and "201901.txt"). Now it is necessary to make statistics on all months' data and find out the documents for the required months.

	A	B
1	=to(0,364).(date("2018-01-01")+~)	/The date sequence of one year is obtained by loop function
2	for A1	="E:/txt/file/"+string(year(A2))+string(month(A2),"00")+".txt"
3		=@|if(file(B2).exists(),B2)

A1, B3 results:

Index	Member
1	2018-01-01
2	2018-01-02
3	2018-01-03
4	2018-01-04
5	2018-01-05
6	2018-01-06
7	2018-01-07
8	2018-01-08
9	2018-01-09
10	2018-01-10

Index	Member
1	E:/txt/file/20180101.txt
2	E:/txt/file/20180104.txt
3	E:/txt/file/20180106.txt
4	E:/txt/file/20180124.txt
5	E:/txt/file/20180126.txt
6	E:/txt/file/20180128.txt
7	E:/txt/file/20180129.txt
8	E:/txt/file/20180130.txt
9	E:/txt/file/20180202.txt
10	E:/txt/file/20180204.txt

	A	B
1	=to(0,364).(date("2018-01-01")+~)	
2	=A1.("E:/txt/file/"+string(year(~))+string(month(~),"00")+".txt")	/Generate file name
3	=A2.select(file(~).exists())	/Select existing files

A1~A3 results:

Index	Member
1	2018-01-01
2	2018-01-02
3	2018-01-03
4	2018-01-04
5	2018-01-05
6	2018-01-06
7	2018-01-07
8	2018-01-08
9	2018-01-09
10	2018-01-10

Index	Member
1	E:/txt/file/20180101.txt
2	E:/txt/file/20180102.txt
3	E:/txt/file/20180103.txt
4	E:/txt/file/20180104.txt
5	E:/txt/file/20180105.txt
6	E:/txt/file/20180106.txt
7	E:/txt/file/20180107.txt
8	E:/txt/file/20180108.txt
9	E:/txt/file/20180109.txt
10	E:/txt/file/20180110.txt

Index	Member
1	E:/txt/file/20180101.txt
2	E:/txt/file/20180104.txt
3	E:/txt/file/20180106.txt
4	E:/txt/file/20180124.txt
5	E:/txt/file/20180126.txt
6	E:/txt/file/20180128.txt
7	E:/txt/file/20180129.txt
8	E:/txt/file/20180130.txt
9	E:/txt/file/20180202.txt
10	E:/txt/file/20180204.txt

Nested loops

A table contains surnames and given names, which are now used to compose the names of the test data by Cartesian product. Please skillfully utilize ~, and get：

[Emily Smith, Alexis Smith, Ryan Smith,
Emily Wilson, Alexis Wilson, Ryan Wilson,
Emily Johnson, Alexis Johnson, Ryan Johnson]

NAME	SURNAME
Emily	Smith
Alexis	Wilson
Ryan	Johnson

	A	B
1	=file("E:/txt/fullname.txt").import@t()	
2	=A1.(SURNAME)	
3	=A1.(NAME)	
4	=A3.~/" "/A2.~	/No loop
5	=A2.(A3.~/" "/A2.~)	/Loop A2
6	=A3.(A2.(A3.~/" "/A2.~))	/Loop A3
7	=A3.(A2.(A3.~/" "/~)).conj()	/Simplify and merge

A2~A7 results:

Index	Member
1	Smith
2	Wilson
3	Johnson

Index	Member
1	Emily
2	Alexis
3	Ryan

Value
Emily Smith

Index	Member
1	Emily Smith
2	Emily Wilson
3	Emily Johnson

Index	Member
1	[Emily Smith,Emily Wilson,Emily Johnson]
2	[Alexis Smith,Alexis Wilson,Alexis Johnson]
3	[Ryan Smith,Ryan Wilson,Ryan Johnson]

Index	Member
1	Emily Smith
2	Emily Wilson
3	Emily Johnson
4	Alexis Smith
5	Alexis Wilson
6	Alexis Johnson
7	Ryan Smith
8	Ryan Wilson
9	Ryan Johnson

Combination of names

[Emily Emily, Emily Alexis, Emily Ryan
Alexis Emily, Alexis Alexis, Alexis Ryan,
Ryan Emily, Ryan Alexis, Ryan Ryan]

HOW

	A	B
1	=file("E:\\txt\\name.txt").import@t()	
2	=A1.(NAME)	
3	=A2.~/" "/A2.~	/No loop
4	=A2.(A2.~/" "/A2.~)	/Loop A2
5	=A2.(A2.(A2.~/" "/A2.~))	/Loop A2
6	=A2.(A2.(A2.~/" "/~)).conj()	/Simplify and merge

A2~A6 results:

Index	Member
1	Emily
2	Alexis
3	Ryan

Value
Emily Emily

Index	Member
1	Emily Emily
2	Alexis Alexis
3	Ryan Ryan

Index	Member
1	[Emily Emily,Alexis Alexis,Ryan Ryan]
2	[Emily Emily,Alexis Alexis,Ryan Ryan]
3	[Emily Emily,Alexis Alexis,Ryan Ryan]

Index	Member
1	Emily Emily
2	Alexis Alexis
3	Ryan Ryan
4	Emily Emily
5	Alexis Alexis
6	Ryan Ryan
7	Emily Emily
8	Alexis Alexis
9	Ryan Ryan

	A	B
1	=file("E:\\txt\\name.txt").import@t()	
2	=A1.(NAME)	
3	=A2.(A2.(A2.~/" "/A2.~))	/Confusing~
4	=A2.((x=A2.~,A2.(x/" "/~)))	/Introducing temporary variable x to refer to outer A1
5	=A2.((x=~,A2.(x/" "/~))).conj()	/Simplify and merge

A3~A5 results:

Index	Member
1	[Emily Emily,Alexis Alexis,Ryan Ryan]
2	[Emily Emily,Alexis Alexis,Ryan Ryan]
3	[Emily Emily,Alexis Alexis,Ryan Ryan]

Index	Member
1	[Emily Emily,Emily Alexis,Emily Ryan]
2	[Alexis Emily,Alexis Alexis,Alexis Ryan]
3	[Ryan Emily,Ryan Alexis,Ryan Ryan]

Index	Member
1	Emily Emily
2	Emily Alexis
3	Emily Ryan
4	Alexis Emily
5	Alexis Alexis
6	Alexis Ryan
7	Ryan Emily
8	Ryan Alexis
9	Ryan Ryan

05

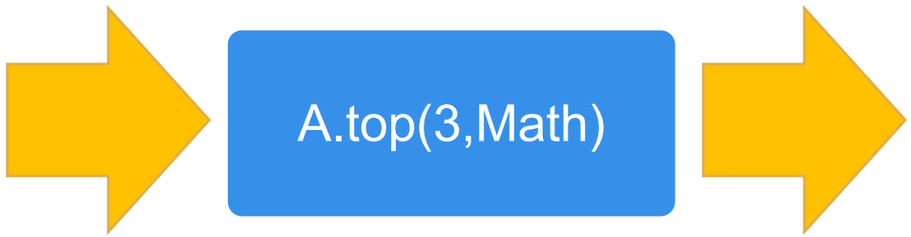
Understanding aggregation

Generally aggregate functions return single value (such as sum/max), but we can also allow aggregate functions to return sets, so topN can be treated as aggregate function.

Example: Look up the three lowest math scores

Score table

序号	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Jessica	87	88	78
3	Brianna	89	90	75
4	Emma	88	84	94
5	Zachary	75	81	85
6	Sophia	74	86	93
7	Hannah	90	76	95
8	Christopher	71	81	86
9	Sean	98	86	81
10	Tyler	87	78	93



Index	Member
1	71
2	74
3	75

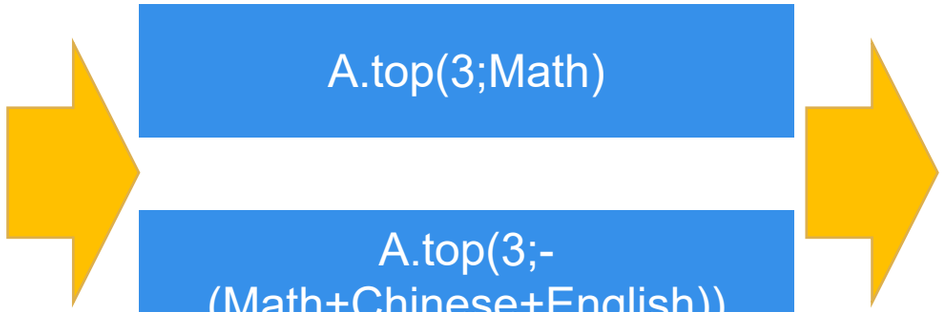
Returned set

TopN can also return the corresponding records (similar to maxp/minp).

Example: Check the scores of the three lowest math scores and the three highest total scores.

Score table

Index	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Jessica	87	88	78
3	Brianna	89	90	75
4	Emma	88	84	94
5	Zachary	75	81	85
6	Sophia	74	86	93
7	Hannah	90	76	95
8	Christopher	71	81	86
9	Sean	98	86	81
10	Tyler	87	78	93



Index	Name	Math	Chinese	English
1	Christopher	71	81	86
2	Sophia	74	86	93
3	Zachary	75	81	85

Index	Name	Math	Chinese	English
1	Emma	88	84	94
2	Sean	98	86	81
3	Hannah	90	76	95

TopN functions, as aggregation functions, can be used in grouping as sum/count.

Calculate the increase rate of each stock in the last two days.

	A	
1	=file("E:/txt/stocknew_price1.txt").import@t()	
2	=A1.groups(stockid;top(2;-DT))	/The last two days of trading records are taken after grouping.
3	=A2.new(stockid,#2(1).CL-#2(2).CL:rises)	/Calculate the increase rate

A1~A3 Results:

Index	stockid	DT	CL
1	1001	2019-04-01	3.95
2	1026	2019-04-01	2.08
3	1028	2019-04-01	18.13
4	1070	2019-04-01	14.29
5	1107	2019-04-01	4.04
6	1134	2019-04-01	11.73
7	1137	2019-04-01	44.03
8	1147	2019-04-01	20.58
9	1206	2019-04-01	12.15
10	1213	2019-04-01	38.06

Index	stockid	top(2;-DT)
1	1001	[[1001,2019-04-24,3.64]...
2	1026	[[1026,2019-04-24,2.[illegible]]...
3	1028	[[1028,2019-04-24,18.[illegible]...
4	1070	[[1070,2019-04-24,13.[illegible]...
5	1107	[[1107,2019-04-24,4.09[illegible]...
6	1134	[[1134,2019-04-24,10.9[illegible]...
7	1137	[[1137,2019-04-24,45.2[illegible]...
8	1147	[[1147,2019-04-24,20.5...
9	1206	[[1206,2019-04-24,13.4...
10	1213	[[1213,2019-04-24,38.8...

Index	stockid	rises
1	1001	-0.3599999999999999
2	1026	-0.04999999999999982
3	1028	-2.0100000000000016
4	1070	-1.0399999999999991
5	1107	0.34999999999999964
6	1134	0.3100000000000005
7	1137	2.8999999999999986
8	1147	0.9400000000000013
9	1206	-0.5700000000000003
10	1213	-2.1299999999999955

Index	stockid	DT	CL
1	1001	2019-04-24	3.64
2	1001	2019-04-23	4.0


THANKS