Structured Text Computing

Raqsoft - esProc

www.raqsoft.com

Understanding Structured Text

Structured text, that is, line text, each line corresponds to a record, the number of fields in each line is the same, which is equivalent to a two-dimensional table in the database. Here are some common structured text.

Txt format, split by"\ T" , with title

CLASS	NAME	English	Chinese	Math	
1	Adams	Brooke	63	31	69
1	Adams	Hannah	89	85	79
1	Adams	Jonathan	88	87	91
1	Allen	Ashley	98	97	97
1	Allen	Brandon	93	76	78
1	Baker	Danielle	83	40	95
1	Brown	Amanda	94	59	81

CSV format, split by ",", Untitled

```
498765,5431438,2019-03-12,2019-04-12,138.5903
34524,5443211,2019-03-15,2019-04-15,208.0805
821741,5461707,2019-03-22,2019-04-22,421.2097
263534,5472320,2019-03-26,2019-04-26,212.6537
238853,5459750,2019-03-21,2019-04-21,817.4593
21071,5393299,2019-03-01,2019-04-01,112.0961
415214,5342607,2019-02-16,2019-03-16,947.1053
224972,5472584,2019-03-26,2019-04-26,133.766
```

Txt format, split by " |" , with title

```
EID|NAME|SURNAME|GENDER|STATE|BIRTHDAY|HIREDATE|DEPT|SALARY
1|Rebecca|Moore|F|California|1974-11-20|2005-03-11|R&D|7000
2|Ashley|Wilson|F|New York|1980-07-19|2008-03-16|Finance|11000
3|Rachel|Johnson|F|New Mexico|1970-12-17|2010-12-01|Sales|9000
4|Emily|Smith|F|Texas|1985-03-07|2006-08-15|HR|7000
5|Ashley|Smith|F|Texas|1975-05-13|2004-07-30|R&D|16000
6|Matthew|Johnson|M|California|1984-07-07|2005-07-07|Sales|11000
7|Alexis|Smith|F|Illinois|1972-08-16|2002-08-16|Sales|9000
8|Megan|Wilson|F|California|1979-04-19|1984-04-19|Marketing|11000
9|Victoria|Davis|F|Texas|1983-12-07|2009-12-07|HR|3000
```

CONTENTS

CONTENTS

01

Single file basic operation

Filter

Small file filtering, select students' scores of class 10

	A	B
1	=file("E:/txt/students_scores.txt").import@t()	/@t option, read the first line as the title, default "\ t" split
2	=A1.select(CLASS==10)	/Select the scores of class 10 and calculate immediately

Text content

CLASS	NAME	English	Chinese	Math
1	Adams Brooke	63	31	69
1	Adams Hannah	89	85	79
1	Adams Jonathan	88	87	91
1	Allen Ashley	98	97	97

A2 result:

Index	CLASS	NAME	English	Chinese	Math
1	10	Adams Ashl...	89	49	91
2	10	Adams Kayla	85	74	45
3	10	Allen Danielle	62	77	88
4	10	Allen Samuel	85	51	57
5	10	Anderson D...	53	74	50

Large file filtering, select students' scores of class 10

	A	B
1	=file("E:/txt/students_scores.txt").cursor@t()	/@t option, read the first line as the title
2	=A1.select(CLASS==10)	/Select the scores of class 10 and the calculation is delayed
3	=A2.fetch()	/Fetch data from cursor and perform additional calculation in A2 at the same time

A1~A3 results:

Value
com.raqsoft.dm.cursor.FileCursor@bcce68f

Value
com.raqsoft.dm.cursor.FileCursor@bcce68f

Index	CLASS	NAME	English	Chinese	Math
1	10	Adams Ashl...	89	49	91
2	10	Adams Kayla	85	74	45
3	10	Allen Danielle	62	77	88
4	10	Allen Samuel	85	51	57
5	10	Anderson D...	53	74	50

Aggregate

Small file aggregation, calculate the total score of Chinese

	A	B
1	=file(“E:/txt/students_scores.csv”).import@t(;,”,”)	/SPL can specify the file separator, such as ”,” here.
2	=A1.sum(Chinese)	/Calculate the total score of Chinese

Text content

```
CLASS,NAME,English,Chinese,Math
1,Adams Brooke,63,31,69
1,Adams Hannah,89,85,79
1,Adams Jonathan,88,87,91
1,Allen Ashley,98,97,97
```

A2 result:

Value
181025

Large file aggregation, calculate the total score of Chinese

	A	B
1	=file("E:/txt/students_scores.csv").cursor@tc()	/When the separator is ”,”, @c can be used
2	=A1.total(sum(Chinese))	/Calculate the total score of Chinese

A1, A2 results:

Value
com.raqsoft.dm.cursor.FileCursor@56225d7

Value
181025

Column calculation

Small file column calculation, calculate the total score of students

	A	B
1	=file("E:/txt/students_scores_.txt").import@t(;,"|")	/The file is divided by "|", and SPL can specify the separator.
2	=A1.derive(English+Chinese+Math:total_score)	/Add a column of the total score of students

Text content

```
CLASS|NAME|English|Chinese|Math
1|Adams Brooke|63|31|69
1|Adams Hannah|89|85|79
1|Adams Jonathan|88|87|91
1|Allen Ashley|98|97|97
```

A2 result:

Index	CLASS	NAME	English	Chinese	Math	total_score
1	1	Adams Bro...	63	31	69	163
2	1	Adams Han...	89	85	79	253
3	1	Adams Jon...	88	87	91	266
4	1	Allen Ashley	98	97	97	292
5	1	Allen Brand...	93	76	78	247

Large file column calculation, calculate the total score of students

	A	B
1	=file("E:/txt/students_scores_.txt").cursor@t(;,"|")	/The file is divided by "|", and SPL can specify the separator.
2	=A1.derive(English+Chinese+Math:total_score)	/derive calculates the total score and returns the cursor
3	=A2.fetch@x(100)	/Fetch data and perform the calculation, close the cursor.

A3 result:

Index	CLASS	NAME	English	Chinese	Math	total_score
1	1	Adams Bro...	63	31	69	163
2	1	Adams Han...	89	85	79	253
3	1	Adams Jon...	88	87	91	266
4	1	Allen Ashley	98	97	97	292
5	1	Allen Brand...	93	76	78	247

Comprehensive calculation

Small file comprehensive calculation,

calculate the Chinese average score of the students in class 10 and the Chinese average score of the students who pass the Chinese Course

	A	B
1	=file("E:/txt/students_scores_.txt").import@t(CLASS,Chinese;,"|")	/The file is divided by "|", take Class and Chinese
2	=A1.select(CLASS==10)	/Select scores of class 10
3	=[A2.avg(Chinese),A2.avg(if(Chinese>=60,Chinese))]	/Calculate the average, and the average of students who pass the course

Text content

```
CLASS|NAME|English|Chinese|Math
1|Adams Brooke|63|31|69
1|Adams Hannah|89|85|79
1|Adams Jonathan|88|87|91
1|Allen Ashley|98|97|97
```

A2, A3 results:

Index	CLASS	Chinese
1	10	49
2	10	74
3	10	77
4	10	51
5	10	74

Index	Member
1	62.666666666666664
2	78.70588235294117

Large file comprehensive calculation,

calculate the Chinese average score of the students in class 10 and the Chinese average score of the students who pass the Chinese Course

	A	B
1	=file("E:/txt/students_scores_.txt").cursor@t(CLASS,Chinese;,"|")	/Read in Class and Chinese by cursor
2	=A1.select(CLASS==10)	/Add select calculation
3	=A2.total(avg(Chinese),avg(if(Chinese>=60,Chinese)))	/Calculate the average, and the average of students who pass the course

A3 result:

Index	Member
1	62.666666666666664
2	78.70588235294117

Read file

Problem 1: Specify field separator

File content

```
CLASS,NAME,English,Chinese,Math
1,Adams Brooke,63,31,69
1,Adams Hannah,89,85,79
1,Adams Jonathan,88,87,91
1,Allen Ashley,98,97,97
```

Divided by ","

SPL code

	A
1	=file(path).import@t(;,",")
2	=file(path).import@tc()

SPL output

Index	CLASS	NAME	English	Chinese	Math
1	1	Adams Bro...	63	31	69
2	1	Adams Ha...	89	85	79
3	1	Adams Jon...	88	87	91
4	1	Allen Ashley	98	97	97

```
CLASS|NAME|English|Chinese|Math
1|Adams Brooke|63|31|69
1|Adams Hannah|89|85|79
1|Adams Jonathan|88|87|91
1|Allen Ashley|98|97|97
```

Divided by "|"

	A
1	=file(path).import@t(;,"|")

Index	CLASS	NAME	English	Chinese	Math
1	1	Adams Bro...	63	31	69
2	1	Adams Ha...	89	85	79
3	1	Adams Jon...	88	87	91
4	1	Allen Ashley	98	97	97

Read file

Problem 2： The first line is the content, no title

File content

1	Adams Brooke	63	31	69
1	Adams Hannah	89	85	79
1	Adams Jonathan	88	87	91
1	Allen Ashley	98	97	97
1	Allen Brandon	93	76	78
1	Baker Danielle	83	40	95

SPL code

No title

	A
1	=file(path).import()

SPL output

Index	_1	_2	_3	_4	_5
1	1	Adams Bro...	63	31	69
2	1	Adams Ha...	89	85	79
3	1	Adams Jon...	88	87	91
4	1	Allen Ashley	98	97	97
5	1	Allen Bran...	93	76	78

Read file

Problem 3： The automatically recognized field type or date format is incorrect

File content

```
user_id,gender,age,insertdate
483833,M,19,2018/1/11
156772,M,31,2018/1/13
173388,M,34,2018/1/21
199107,F,25,2018/1/5
122560,M,23,2018/1/2
```

SPL code

user_id should be a string, Data format：yyyy/MM/dd

Normal reading, Run function modification

	A
1	=file(path).import@t()
2	=A2.run(user_id=string(user_id),insertdate=date(insertdate,"yyyy/MM/dd"))

Correct reading

	A
1	=file(path).import@t(user_id:string,gender,age,insertdate:date:"yyyy/MM/dd")

SPL output

Index	user_id	gender	age	insertdate
1	483833	M	19	2018/1/11
2	156772	M	31	2018/1/13
3	173388	M	34	2018/1/21
4	199107	F	25	2018/1/5
5	122560	M	23	2018/1/2

Normal reading

Index	user_id	gender	age	insertdate
1	483833	M	19	2018-01-11
2	156772	M	31	2018-01-13
3	173388	M	34	2018-01-21
4	199107	F	25	2018-01-05
5	122560	M	23	2018-01-02

Run function modification

Index	user_id	gender	age	insertdate
1	483833	M	19	2018-01-11
2	156772	M	31	2018-01-13
3	173388	M	34	2018-01-21
4	199107	F	25	2018-01-05
5	122560	M	23	2018-01-02

Correct reading

Read file

Problem 4: Read partial fields

File content

CLASS	NAME	English	Chinese	Math
1	Adams Brooke	63	31	69
1	Adams Hannah	89	85	79
1	Adams Jonathan	88	87	91
1	Allen Ashley	98	97	97
1	Allen Brandon	93	76	78

SPL code

	A
1	=file(path).import@t(CLASS,Chinese)
2	=file(path).import@t(#1,#4)

SPL output

Index	CLASS	Chinese
1	1	31
2	1	85
3	1	87
4	1	97
5	1	76

Read file

Problem 5： character set

File content

```
user_id,reg_mon,gender,age,cell_province,id_province,id_city,insertdate
483833,2017-04,男,19,c29,c26,c26241,2018-12-11
156772,2016-05,男,31,c11,c11,c11159,2018-02-13
173388,2016-05,男,34,c02,c02,c02182,2018-08-21
199107,2016-07,女,25,c09,c09,c09046,2018-06-05
122560,2016-03,男,23,c05,c05,c05193,2018-04-02
```

SPL code

	A
1	=file(path).import@tc()

Normal reading

	A
1	=file(path:"utf-8").import@tc()

Specify character set reading

SPL output

Normal reading

Index	user_id	reg_mon	gender	age	cell_provin...	id_province	id_city	insertdate
1	483833	2017-04	鐢□	19	c29	c26	c26241	2018-12-11
2	156772	2016-05	鐢□	31	c11	c11	c11159	2018-02-13
3	173388	2016-05	鐢□	34	c02	c02	c02182	2018-08-21
4	199107	2016-07	瀛□	25	c09	c09	c09046	2018-06-05
5	122560	2016-03	鐢□	23	c05	c05	c05193	2018-04-02

Specify character set reading

Index	user_id	reg_mon	gender	age	cell_provin...	id_province	id_city	insertdate
1	483833	2017-04	男	19	c29	c26	c26241	2018-12-11
2	156772	2016-05	男	31	c11	c11	c11159	2018-02-13
3	173388	2016-05	男	34	c02	c02	c02182	2018-08-21
4	199107	2016-07	女	25	c09	c09	c09046	2018-06-05
5	122560	2016-03	男	23	c05	c05	c05193	2018-04-02

CONTENTS

02

Single file advanced operation

Sort

Small file sorting 1: Rank students' scores in ascending Chinese order

	A	B
1	=file("E:/txt/students_score.txt").import@t()	/Read in file
2	=A1.sort(Chinese)	/Sort in ascending order

A1, A2 results:

Index	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Jessica	87	88	78
3	Brianna	89	90	75

Index	Name	Math	Chinese	English
1	Hannah	90	76	95
2	Tyler	87	78	93
3	Zachary	75	81	85

Large file sorting 1: Rank students' scores in ascending Chinese order

	A	B
1	=file("E:/txt/students_score.txt").cursor@t()	/Create cursor
2	=A1.sortx(Chinese)	/Sort in ascending order and return cursor
3	=A2.fetch@x(100)	/Fetch data

A3 result:

Index	CLASS	NAME	English	Chinese	Math	total_score
1	1	Adams Bro...	63	31	69	163
2	1	Adams Han...	89	85	79	253
3	1	Adams Jon...	88	87	91	266
4	1	Allen Ashley	98	97	97	292
5	1	Allen Brand...	93	76	78	247

Sort

Small file sorting 2: Rank students' scores in descending order of total scores

	A	B
1	=file("E:/txt/students_score.txt").import@t()	/Read in file
2	=A1.sort@z(Math+English+Chinese)	/Calculate column descending sorting

A1, A2 results:

Index	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Jessica	87	88	78
3	Brianna	89	90	75

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

Large file sorting 2: Rank students' scores in descending order of total scores

	A	B
1	=file("E:/txt/students_score.txt").cursor@t()	/Create cursor
2	=A1.sortx@z(Math+English+Chinese)	/Calculate column descending sorting and return cursor
3	=A2.fetch@x(100)	/Fetch data

A3 result:

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

Sort

Small file sorting 3: Rank students in ascending class order and total scores in descending order

	A	B
1	=file("E:/txt/students_scores.txt").import@t()	/Read in file
2	=A1.sort(CLASS,-(English+Chinese+Math))	/Sort by ascending class order and total scores descending order

A1, A2 results:

Index	CLASS	NAME	English	Chinese	Math
1	1	Adams Bro...	63	31	69
2	1	Adams Ha...	89	85	79
3	1	Adams Jon...	88	87	91

Index	CLASS	NAME	English	Chinese	Math
1	1	Allen Ashley	98	97	97
2	1	Lewis Anto...	93	92	94
3	1	Adams Jon...	88	87	91

Large file sorting 3: Rank students in ascending class order and total scores in descending order

	A	B
1	=file("E:/txt/students_scores.txt").cursor@t()	/Create cursor
2	=A1.sortx(CLASS,-(English+Chinese+Math))	/Sort by requirement and return cursor
3	=A2.fetch@x(100)	/Fetch data

A3 result:

Index	CLASS	NAME	English	Chinese	Math
1	1	Allen Ashley	98	97	97
2	1	Lewis Anto...	93	92	94
3	1	Adams Jon...	88	87	91

Group

Small file grouping and aggregation

Example: Count the number of user logins in each province

	A	B
1	=file("E:/txt/user_info_reg.csv").import@tc()	/Read in file
2	=A1.groups(id_province;count(~):cnt)	/count after grouping

A1, A2 results:

Index	user_id	reg_mon	age	cell_provin...	id_province	id_city	insertdate	reg_time
1	483833	2017-04	19	c29	c26	c26241	2018-12-11	56558
2	156772	2016-05	31	c11	c11	c11159	2018-02-13	81617
3	173388	2016-05	34	c02	c02	c02182	2018-08-21	729
4	199107	2016-07	25	c09	c09	c09046	2018-06-05	86299
5	122560	2016-03	23	c05	c05	c05193	2018-04-02	2657

Index	id_province	cnt
1	c01	27202
2	c02	61735
3	c03	14433
4	c04	100639
5	c05	48326

Group

Large file grouping and aggregation (small result set)

Example: Count the number of user logins in each province

	A	B
1	=file("E:/txt/user_info_reg.csv").cursor@tc()	/Create cursor
2	=A1.groups(id_province;count(~):cnt)	/count after grouping

A1, A2 results:

Value
com.raqsoft.dm.cursor.FileCursor@5ae3860d

Index	id_province	cnt
1	c01	27202
2	c02	61735
3	c03	14433
4	c04	100639
5	c05	48326

Group

Large file grouping and aggregation （large result set）

Example： Count the total login time of each user

	A	B
1	=file("E:/txt/user_info_reg.csv").cursor@tc()	/Create cursor
2	=A1.groupx(user_id;sum(reg_time):total_reg)	/sum after grouping and return cursor
3	=A2.fetch(1000)	

A1~A3 results:

Value
com.raqsoft.dm.cursor.FileCursor@3a544c70

Value
com.raqsoft.dm.cursor.MemoryCursor@3bd310fd

Index	user_id	total_reg
1	1	2345
2	2	74990
3	3	53724
4	4	47153
5	5	23507

Group

Small file filtering after grouping

Example: Identify users who log in for less than 1000 minutes

	A	B
1	=file("E:/txt/user_info_reg.csv").import@tc()	Read in file
2	=A1.groups(user_id;sum(reg_time):total_reg)	/ sum after grouping
3	=A2.select(total_reg<1000)	/Filtering

A1~A3 results:

Index	user_id	reg_mon	age	cell_province	id_province	id_city	insertdate	reg_time
1	483833	2017-04	19	c29	c26	c26241	2018-12-11	56558
2	156772	2016-05	31	c11	c11	c11159	2018-02-13	81617
3	173388	2016-05	34	c02	c02	c02182	2018-08-21	729
4	199107	2016-07	25	c09	c09	c09046	2018-06-05	86299
5	122560	2016-03	23	c05	c05	c05193	2018-04-02	2657

Index	user_id	total_reg
1	1	2345
2	2	74990
3	3	53724
4	4	47153
5	5	23507

Index	user_id	total_reg
1	41	512
2	68	130
3	90	486
4	203	865
5	519	556

Group

Large file filtering after grouping

Example： Identify users who log in for less than 1000 minutes

	A	B
1	=file("E:/txt/user_info_reg.csv").cursor@tc()	/Create cursor
2	=A1.groupx(user_id;sum(reg_time):total_reg)	/sum after grouping and return cursor
3	=A2.select(total_reg<1000).fetch()	/Fetch data and filter

A1~A3 results:

Index	user_id	reg_mon	age	cell_province	id_province	id_city	insertdate	reg_time
1	483833	2017-04	19	c29	c26	c26241	2018-12-11	56558
2	156772	2016-05	31	c11	c11	c11159	2018-02-13	81617
3	173388	2016-05	34	c02	c02	c02182	2018-08-21	729
4	199107	2016-07	25	c09	c09	c09046	2018-06-05	86299
5	122560	2016-03	23	c05	c05	c05193	2018-04-02	2657

Index	user_id	total_reg
1	1	2345
2	2	74990
3	3	53724
4	4	47153
5	5	23507

Index	user_id	total_reg
1	41	512
2	68	130
3	90	486
4	203	865
5	519	556

Deduplicate

Small file deduplicates, find all user IDs

	A	B
1	=file("E:/txt/user_info_reg.csv").import@tc()	/Read the specified field
2	=A1.id(user_id)	/Deduplication, view user ID

A1~A2 results:

Index	user_id	reg_mon	age	cell_provin...	id_province	id_city	insertdate	reg_time
954205	363648	2017-01	23	c08	c19	c19303	2018-01-17	51365
954206	292977	2016-10	22	c06	c27	c27051	2018-06-13	84647
954207	644550	2017-09	25	c04	c04	c04348	2018-04-10	18970
954208	608246	2017-08	35	c04	c04	c04319	2018-12-06	54282
954209	834041	2018-04	24	c25	c09	c09294	2018-09-11	33953

Index	Member
928191	928191
928192	928192
928193	928193
928194	928194
928195	928195

Large file deduplicates, find all user IDs

	A	B
1	=file("E:/txt/user_info_reg.csv").cursor@tc()	/Create cursor
2	=A1.id(user_id)	/Deduplication, view user ID

A2 result:

Index	Member
928191	928191
928192	928192
928193	928193
928194	928194
928195	928195

Count distinct

Small file count distinct,
Remove the duplicate data by date and product, and then count the number of records.

	A	B
1	=file("E:/txt/PRODUCT_SALE.txt").import@t(DATE,PID)	/Read the specified field
2	=A1.groups(date(DATE),PID)	/Deduplication
3	=A2.len()	/Count non duplicate records

A1~A3 results:

Index	DATE	PID
9999998	2018-12-31	10014923
9999999	2018-12-31	10040866
10000000	2018-12-31	10057996

Index	DATE	PID
9849395	2018-12-31	10099926
9849396	2018-12-31	10099939
9849397	2018-12-31	10099955

Value
9849397

Large file count distinct,
Remove the duplicate data by date and product, and then count the number of records.

	A	B
1	=file("E:/txt/PRODUCT_SALE.txt").cursor@t(DATE,PID)	/Read the specified field
2	=A1.groupx(date(DATE),PID)	/Deduplication
3	=A2.skip()	/Count non duplicate records

A3 result:

Value
9849397

Group deduplication count

Small file groups deduplication count, Count the number of days with sales records for each product

	A	B
1	=file("E:/txt/PRODUCT_SALE.txt").import@t(DATE,PID)	/Read the specified field
2	=A1.groups(PID,date(DATE))	/Deduplication
3	=A2.groups(PID;count(1):no_sdate)	/Group, count the days with sales records

A1~A3 results:

Index	DATE	PID
9999998	2018-12-31	10014923
9999999	2018-12-31	10040866
10000000	2018-12-31	10057996

Index	PID	date(DATE)
9849395	10100001	2018-11-26
9849396	10100001	2018-11-28
9849397	10100001	2018-12-10

Index	PID	no_sdate
99998	10099999	93
99999	10100000	100
100000	10100001	109

Large file groups deduplication count, Count the number of days with sales records for each product

	A	B
1	=file("E:/txt/PRODUCT_SALE.txt").cursor@t(DATE,PID)	/Create cursor
2	=A1.groupx(date(DATE),PID)	/Deduplication
3	=A2.groups(PID;count(1):no_sdate)	/Group, count the days with sales records

A3 result:

Index	PID	no_sdate
99998	10099999	93
99999	10100000	100
100000	10100001	109

Parallel filtering to filter out the product sales records in October (multi cursor)

	A
1	=now()
2	=file("E:/txt/PRODUCT_SALE.txt").cursor@t()
3	=A2.select(month(DATE)==10)
4	=A3.fetch(100000)
5	=interval@ms(A1,now())

Single cursor filtering

A4, A5 results:

Index	ID	PID	DATE	QUANTITY	SID
99999	2051417	10051515	2011-10-02	37	10075
100000	2051976	10056022	2011-10-02	67	10143

Value
1525

	A
1	=now()
2	=file("E:/txt/PRODUCT_SALE.txt").cursor@mt()
2	=file("E:/txt/PRODUCT_SALE.txt").cursor@t().mcursor()
3	=A2.select(month(DATE)==10)
4	=A3.fetch(100000)
5	=interval@ms(A1,now())

Multiple cursor filtering, method 1: reading data and filtering are parallel

Multiple cursor filtering, method 2: Only filtering is parallel

A4, A5 results:

Index	ID	PID	DATE	QUANTITY	SID
99999	3573602	10028733	2016-10-09	103	10689
100000	3579513	10010320	2016-10-09	35	10074

Value
1088

Index	ID	PID	DATE	QUANTITY	SID
99999	6110232	10037262	2010-10-16	36	10413
100000	6113472	10011967	2010-10-16	54	10663

Value
1261

Note: when multiple cursors are used, except fetch all, the returned result set may change the original data order.

Parallel group aggregation to calculate the total sales volume of each product (multiple cursors)

	A
1	=now()
2	=file("E:/txt/PRODUCT_SALE.txt").cursor@t()
3	=A2.groups(PID;sum(QUANTITY):total_num)
4	=interval@ms(A1,now())

Single cursor filtering

A4, A5 results:

Index	PID	total_num
1	10000002	5799
2	10000003	6554

Value
10043

	A
1	=now()
2	=file("E:/txt/PRODUCT_SALE.txt").cursor@mt()
2	=file(“E:/txt/PRODUCT_SALE.txt”).cursor@t().mcursor()
3	=A2.groups(PID;sum(QUANTITY):total_num)
4	=interval@ms(A1,now())

Multiple cursor filtering, method 1: reading data and filtering are parallel

Multiple cursor filtering, method 2: Only filtering is parallel

A4, A5 results:

Index	PID	total_num
1	10000002	5799
2	10000003	6554

Value
4054

Index	PID	total_num
1	10000002	5799
2	10000003	6554

Value
7192

CONTENTS

03

Join Calculation

SQL Join

1. Cartesian product

Employee

ID	NAME	DEPT
1	David	1
2	Daniel	2
3	Andrew	1

×

Department

ID	NAME
1	Sales
2	R&D

=

ID	NAME	DEPT	ID	NAME
1	David	1	1	Sales
1	David	1	2	R&D
2	Daniel	2	1	Sales
2	Daniel	2	2	R&D
3	Andrew	1	1	Sales
3	Andrew	1	2	R&D

2. Conditional filtering

ID	NAME	DEPT	ID	NAME
1	David	1	1	Sales
~~1~~	~~David~~	~~1~~	~~2~~	~~R&D~~
~~2~~	~~Daniel~~	~~2~~	~~1~~	~~Sales~~
2	Daniel	2	2	R&D
3	Andrew	1	1	Sales
~~3~~	~~Andrew~~	~~1~~	~~2~~	~~R&D~~

Employee.DEPT = Department.ID →

ID	NAME	DEPT	ID	NAME
1	David	1	1	Sales
2	Daniel	2	2	R&D
3	Andrew	1	1	Sales

SPL Join

Employee

ID	NAME	DEPT
1	David	1
2	Daniel	2
3	Andrew	1

JOIN/SWITHCH

Department

ID	NAME
1	Sales
2	R&D

=

Employee	Department
[1, David, 1]	[1, Sales]
[2, Daniel, 2]	[2, R&D]
[3, Andrew, 1]	[1, Sales]

After joining two or more sets, SPL takes the tuple composed of set members as members instead of simply expanding the data structures of all sets. SPL is not only more in line with the concept and original meaning of join, the relationship between tables is more clear and visible, and the syntax is more concise than SQL.

Join calculation

Two small files foreign key join 1 Example: Find out employees whose couple age is greater than 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")	/Select the spouse relationship in table 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(Emp1=Emp1.Name,Emp2=Emp2.Name)	/Change the record to the name field of the record

A1

Index	ID	Name	Gender	Post	Birthday	AccountNo	BasePay
1	1	Mike	Female	Sale	1968-12-0...	536936891...	5600.0
2	2	Jake	Male	Vice Presid...	1962-02-1...	964107677...	2500.0
3	3	Lucy	Female	Sale	1973-08-3...	665248245...	10800.0
4	4	Andy	Male	Sales Man...	1968-09-1...	650028860...	7500.0
5	5	Jim	Male	Sales Man...	1965-03-0...	441380247...	4700.0

A2

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

A3 after A3 executed

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

Foreign key objectification

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

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

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

A3 after A4 executed

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

A5 after A5 executed

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

A5 after A6 executed

Join calculation

Two small files foreign key join 2 Example：Find out the department with the youngest department manager.

	A	B
1	=file("E:/txt/EMPLOYEE.txt").import@t()	/Read employee information
2	=file("E:/txt/DEPARTMENT.txt").import@t()	/Read department information
3	=A2.join(MANAGER,A1:EID,~:manager)	/Employee information foreign key objectification and joins with department Information.
4	=A3.minp(manager.(age(BIRTHDAY))).DEPT	/Find the department with the youngest department manager.

A1

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

A2

Index	DEPT	MANAGER
1	Administrat...	20
2	Finance	2
3	HR	162
4	Marketing	47
5	Production	58
6	R&D	5
7	Sales	40
8	Technology	55

A3

Index	DEPT	MANAGER	manager
1	Administration	20	[20,Alexis,Alle...
2	Finance	2	[2,Ashley,Wils...
3	HR	162	[162,Gabriel,...
4	Marketing	47	[47,Elizabeth
5	Production	58	[5
6	R&D	5	[5
7	Sales	40	[40,Madeline,...
8	Technology	55	[55,Olivia,And...

Foreign key objectification

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
20	Alexis	Allen	F	Florida	1977-08-07	2007-08-07	Administrat...	16000

A4

Value
Finance

Join calculation

Two small files foreign key join 3 Example： Add user information in user information table to user credit information table to form a wide table.

	A	B
1	=file("E:/txt/lending_info.csv").import@tc()	/Read lending information
2	=file("E:/txt/user_info.csv":"utf-8").import@tc()	/Read user information, character set is "UTF-8"
3	=A2.group@1s(user_id)	/user_id deduplicates，take the first item after grouping to ensure that the primary key is unique
4	=A3.fname().m(2:)	/List user information other than user ID
5	=A1.join(user_id,A3:user_id,${A4.concat@c()})	/Join the two tables to form a wide table

A1

Index	user_id	listing_id	auditing_d...	due_date	due_amt
129998	233442	5319333	2019-02-10	2019-03-10	168.3364
129999	20165	5336095	2019-02-15	2019-03-15	350.2759
130000	265473	5460170	2019-03-21	2019-04-21	293.8277

A2

Index	user_id	reg_mon	gender	age	cell_province	id_province	id_city	insertdate
954207	644550	2017-09	男	25	c04	c04	c04348	2018-04-10
954208	608246	2017-08	男	35	c04	c04	c04319	2018-12-06
954209	834041	2018-04	男	24	c25	c09	c09294	2018-09-11

A3

Index	user_id	reg_mon	gender	age	cell_provin...	id_province	id_city	insertdate
928193	928193	2019-03	男	23	c07	c07	c07297	2019-03-29
928194	928194	2019-03	男	28	c20	c26	c26243	2019-03-29
928195	928195	2019-03	男	23	c29	c29	c29063	2019-03-30

Foreign key join needs to ensure that the primary key must be unique.
In other words, user ID in A2 must be unique.

A5

Index	user_id	listing_id	auditing_d...	due_date	due_amt	reg_mon	gender	age	cell_provin...	id_province	id_city	insertdate
129998	233442	5319333	2019-02-10	2019-03-10	168.3364	2016-08	女	30	c20	c05	c05103	2019-02-09
129999	20165	5336095	2019-02-15	2019-03-15	350.2759	2015-03	男	34	c06	c06	c06195	2018-11-30
130000	265473	5460170	2019-03-21	2019-04-21	293.8277	2016-09	男	29	c13	c13	c13003	2019-03-20

Join calculation

One large file joins one small file 1

Example: Products information and sales information are stored in two tables. Please calculate the total sales of products with sales quantity less than 10.

	A	B
1	=file("E:/txt/Products.txt").import@t().primary@i(ID)	/Read products file and create index(in-memory table)
2	=file("E:/txt/Sales.txt").cursor@t()	/Either single cursor or multi cursor
3	=A2.select(quantity<=10)	/Cursor filtering
Method 1 4	=A3.switch(productid,A1:ID)	/Attach foreign key objectification to cursor with switch
5	=A4.groups(;sum(quantity*productid.Price):total)	/Sum and aggregation
Method 2 4	=A3.join(productid,A1:ID,~:products)	/Attach foreign key objectification to cursor with join
5	=A4.groups(;sum(quantity*products.Price):total)	/Sum and aggregation
Method 3 4	=A3.join(productid,A1:ID,Price)	/Splicing price field with join
5	=A4.groups(;sum(quantity*Price):total)	/Sum and aggregation

A1, A5 results:

Index	ID	Name	Category	Price
1	1	Apple juice	Low-end	18.0
2	2	Mile	Low-end	19.0
3	3	Tomato sa...	Low-end	10.0
4	4	Salt	Low-end	22.0
5	5	Sesame oil	Low-end	21.35

Index	total
1	142740.18000000008

Join calculation

One large file joins one small file 2

Example: Add user information in user information table to user credit information table to form a wide table.

	A	B
1	=file("E:/txt/lending_info.csv").cursor@tc()	/Create cursor
2	=file("E:/txt/user_info.csv":"utf-8").import@tc()	/Read user information, character set is "UTF-8"
3	=A2.group@1s(user_id)	/user_id deduplicates，take the first item after grouping
4	=A3.fname().m(2:)	/List user information other than user ID
5	=A1.join(user_id,A3:user_id,${A4.concat@c()})	/Cursor foreign key joins small table, return cursor
6	=A5.fetch@x(100)	/Fetch 100 rows，close cursor

A1

Value
com.raqsoft.dm.cursor.FileCursor@158cf9ff

A2

Index	user_id	reg_mon	gender	age	cell_provin...	id_province	id_city	insertdate
1	483833	2017-04	男	19	c29	c26	c26241	2018-12-11
2	156772	2016-05	男	31	c11	c11	c11159	2018-02-13
3	173388	2016-05	男	34	c02	c02	c02182	2018-08-21
4	199107	2016-07	女	25	c09	c09	c09046	2018-06-05
5	122560	2016-03	男	23	c05	c05	c05193	2018-04-02

A6

Index	user_id	listing_id	auditing_d...	due_date	due_amt	reg_mon	gender	age	cell_provin...	id_province	id_city	insertdate
1	498765	5431438	2019-03-12	2019-04-12	138.5903	2017-05	男	37	c11	c11	c11245	2019-03-11
2	34524	5443211	2019-03-15	2019-04-15	208.0805	2015-07	男	26	c25	c25	c25074	2019-03-14
3	821741	5461707	2019-03-22	2019-04-22	421.2097	2018-03	女	25	c22	c22	c22308	2019-03-21
4	263534	5472320	2019-03-26	2019-04-26	212.6537	2016-09	女	42	c17	c17	c17290	2019-03-25
5	238853	5459750	2019-03-21	2019-04-21	817.4593	2016-08	男	44	c10	c26	c26057	2018-12-21

Join calculation

Two large files join

Example: The order table and order details table are stored in two files respectively. Calculate the total consumption amount of each client.

	A	B
1	=file(“E:/txt/Orders.txt”).cursor@t().sortx(orderid)	/Sortx is not required if the data is known to be ordered by OrderID.
2	=file("E:/txt/OrderDetails.txt").cursor@t().sortx(orderid)	
3	=joinx(A1:order,orderid;A2: detail,orderid)	/Using joinx to implement, two cursors are joined
4	=A3.groups(order.clientid:clientid;sum(detail.price):amount)	/Calculate to obtain each client’s consumption amount.

Orders table

orderid	clientid	date
10012	100658	2019-02-13
10023	103478	2019-01-12
10040	108013	2019-01-04
10045	100373	2019-01-20
10054	102525	2019-03-07
10057	102740	2019-03-21
10068	107448	2019-03-18
10095	107735	2019-03-27
10108	106552	2019-03-28
10114	108699	2019-01-10
10120	101530	2019-02-15
10134	101134	2019-02-01

Order details table

orderid	no	productid	price
10012	1	3018	428.5
10012	2	3019	349.2
10023	1	3019	349.2
10040	1	3093	139.5
10040	2	3070	137.9
10040	3	3050	210.6
10045	1	3012	21.8
10054	1	3064	462.5
10057	1	3049	123.5
10057	2	3059	186.1
10068	1	3077	145.8
10068	2	3070	137.9

A4 result:

Index	clientid	amount
1	100008	12350.0
2	100011	53400.00000000006
3	100015	13789.999999999976
4	100037	44200.0
5	100042	48380.00000000006
6	100075	27290.000000000044
7	100077	109920.00000000055
8	100083	12479.999999999984
9	100087	48040.00000000009
10	100088	59529.99999999963

Join calculation

Set operations of small files 1

Example： Find community club members according to requirements.

	A	B
1	=file(“E:/txt/running.txt”).import@t().(NAME,SURNAME)	/Members of running club
2	=file("E:/txt/ball.txt").import@t().([NAME,SURNAME])	/Members of ball club
3	=A1|A2	Sum，the sum of the two clubs
4	=A1&A2	/Union，members who sign up for at least one club
5	=A1^A2	/Intersection，members who sign up for both clubs
6	=A1\A2	/Minus，members who sign up for running club only

A1

Index	Member
28	[Jacob,Moore]
29	[Jacob,Wilson]
30	[Jonathan,Miller]

A2

Index	Member
34	[Daniel,Smith]
35	[Alyssa,Smith]
36	[Cameron,Johnson]

A3

Index	Member
64	[Daniel,Smith]
65	[Alyssa,Smith]
66	[Cameron,Johnson]

A4

Index	Member
57	[Daniel,Smith]
58	[Alyssa,Smith]
59	[Cameron,Johnson]

A5

Index	Member
5	[Jacob,Moore]
6	[Jacob,Wilson]
7	[Jonathan,Miller]

A6

Index	Member
21	[Abigail,Smith]
22	[Nathan,Johnson]
23	[Joshua,King]

Join calculation

Set operations of small files 2

Example: The user login information is stored in different files monthly. Query the user login information according to different requirement.

	A	B
1	=file("E:/txt/user_login_info_1.txt").import@t().group@1(userid)	/User's first login information in January
2	=file("E:/txt/user_login_info_2.txt").import@t().group@1(userid)	/User's first login information in February
3	=file("E:/txt/user_login_info_3.txt").import@t().group@1(userid)	/User's first login information in March
4	=[A1,A2,A3].merge(userid)	/Merge the user's first login information of each month in order according to the userid
5	=[A1,A2,A3].merge@u(userid)	/Union，Users who log in at least once in 3 months
6	=[A1,A2,A3].merge@i(userid)	/Intersection，Users logged in every month for 3 months
7	=[A1,A2,A3].merge@d(userid)	/Difference，Users logged in only in January

A1

Index	userid	login
93675	699997	2019-01-2
93676	699998	2019-01-1
93677	700000	2019-01-13 14:57:35

A2

Index	userid	login
96997	699998	2019-02-03
96998	699999	2019-02-04
96999	700000	2019-02-12 08:46:00

A3

Index	userid	login
97586	699998	2019-03-2
97587	699999	2019-03-1
97588	700000	2019-03-01 00:52:27

A4

Index	userid	login
288262	700000	2019-03-0
288263	700000	2019-02-12
288264	700000	2019-01-13 14:57:35

A5

Index	userid	login
99996	699998	2019-01-1
99997	699999	2019-02-0
99998	700000	2019-01-13 14:57:35

A6

Index	userid	login
88669	699997	2019-01-2
88670	699998	2019-01-1
88671	700000	2019-01-13 14:57:35

A7

Index	userid	login
64	698338	2019-01-24
65	699398	2019-01-10
66	699763	2019-01-13 14:27:32

Join calculation

Set operations of large files

Example： The user login information is stored in different files monthly. Query the user login information according to different requirement.

	A	B
1	=file("E:/txt/user_login_info_1.txt").cursor@t().sortx(userid).group@1(userid)	/The first login information of users in January, February and March. Sortx is not required if the data is known to be in order.
2	=file("E:/txt/user_login_info_2.txt").cursor@t().sortx(userid).group@1(userid)	
3	=file("E:/txt/user_login_info_3.txt").cursor@t().sortx(userid).group@1(userid)	
4	=[A1,A2,A3].mergex(userid).fetch()	/Merge the user's first login information of each month in order according to the userid
4	=[A1,A2,A3].mergex@u(userid).fetch()	/Union，Users who log in at least once in 3 months
4	=[A1,A2,A3].mergex@i(userid).fetch()	/Intersection，Users logged in every month for 3 months
4	=[A1,A2,A3].mergex@d(userid).fetch()	/Difference, users logged in in January and not logged in in February or March

Index	userid	login
288262	700000	2019-03-01 00:52:27
288263	700000	2019-02-12 08:46:00
288264	700000	2019-01-13 14:57:35

Index	userid	login
99996	699998	2019-01-17 18:13:56
99997	699999	2019-02-04 02:58:13
99998	700000	2019-01-13 14:57:35

Index	userid	login
88669	699997	2019-01-27 06:37:22
88670	699998	2019-01-17 18:13:56
88671	700000	2019-01-13 14:57:35

Index	userid	login
64	698338	2019-01-24 09:27:19
65	699398	2019-01-10 02:16:04
66	699763	2019-01-13 14:27:32

CONTENTS

04

SQL and command line

where

SQL computing structured text data 1 (Filter)

Example: Find out the scores of students in class 10

	A	B
1	$select * from E:/txt/Students_scores.txt where CLASS=10	/SQL filtering

A1 result:

Index	CLASS	NAME	English	Chinese	Math
1	10	Adams Ashley	89	49	91
2	10	Adams Kayla	85	74	45
3	10	Allen Danielle	62	77	88
4	10	Allen Samuel	85	51	57
5	10	Anderson Des...	53	74	50

select - Aggregation

SQL computing structured text data 2 （Aggregation)

Example: Calculate the average score of Chinese for all students

	A	B
1	$select avg(Chinese) from E:/txt/Students_scores.txt	/SQL aggregation

A1 result:

Index	_1
1	62.16517857142857

Select – Column calculation

SQL computing structured text data 3 (Column calculation)

Example: Add a column of total scores of students

	A	B
1	$select *,English+Chinese+Math as total_score from E:/txt/students_scores.txt	/SQL adds a calculated column

A1 result:

Index	CLASS	NAME	English	Chinese	Math	total_score
1	1	Adams Bro...	63	31	69	163
2	1	Adams Han...	89	85	79	253
3	1	Adams Jon...	88	87	91	266
4	1	Allen Ashley	98	97	97	292
5	1	Allen Brand...	93	76	78	247

case...when...

SQL computing structured text data 4 (case...when...)

Example: Add one column: if the English score is equal to or higher than 60, it will be considered as pass, and the others will be considered as fail.

	A	B
1	$select *, case English when English>=60 then 'Pass' else 'Fail' end as English_evaluation from E:/txt/students_scores.txt	/SQL adds an “English evaluation” column

A1 result:

Index	CLASS	NAME	English	Chinese	Math	English_evaluat...
1	1	Adams Brooke	63	31	69	Fail
2	1	Adams Hannah	89	85	79	Fail
3	1	Adams Jonathan	88	87	91	Fail
4	1	Allen Ashley	98	97	97	Fail
5	1	Allen Brandon	93	76	78	Fail

order by

SQL computing structured text data 5 (Sort)

Example: Sort in ascending order of class, and in descending order of total score

	A	B
1	$select * from E:/txt/students_scores.txt order by CLASS,English+Chinese+Math desc	/Sort according to conditions

A1 result:

Index	CLASS	NAME	English	Chinese	Math
1	1	Allen Ashley	98	97	97
2	1	Lewis Anto...	93	92	94
3	1	Adams Jon...	88	87	91
4	1	Walker Ja...	84	83	89
5	1	Adams Ha...	89	85	79

group by

SQL computing structured text data 6 (Grouping and aggregation)

Example: Query the mathematical average of each class

	A	B
1	$select CLASS,avg(English) as avg_En from E:/txt/students_scores.txt group by CLASS	/Grouping and aggregation

A1 result:

Index	CLASS	avg_En
1	1	74.43103448275862
2	2	77.34375
3	3	72.72857142857143
4	4	69.6046511627907
5	5	70.34615384615384

having

SQL computing structured text data 7 (Grouping and filtering)

Example: Query classes with an average English score of less than 70

	A	B
1	$select CLASS,avg(English) as avg_En from E:/txt/students_scores.txt group by CLASS having avg(English)<70	/Grouping and filtering

A1 result:

Index	CLASS	avg_En
1	4	69.6046511627907
2	7	69.86

distinct

SQL computing structured text data 8 (Deduplicate)

Example: View class ID

	A	B
1	$select distinct(CLASS) from E:/txt/students_scores.txt	/distinct

A1 result:

Index	_1
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8

count(distinct)

SQL computing structured text data 8 (count distinct)

Example: Count the quantity of products

	A	B
1	$select count(distinct PID) from E:/txt/PRODUCT_SALE.txt	/count(distinct)

A1 result:

Index	_1
1	100000

group by…group by…/distinct…group by…

SQL computing structured text data 9 (Group and count distinct)

Example: Count the number of days with sales records for each product

	A	B
1	$select PID,count(*) as no_sdate from (select PID from E:/txt/PRODUCT_SALE.txt group by PID,DATE) group by PID	/group+group，group and count distinct
2	$select PID,count(*) as no_sdate from (select distinct PID,DATE from E:/txt/PRODUCT_SALE.txt) group by PID	/distinct+group，group and count distinct

A1 result:

Index	PID	no_sdate
99998	10099999	93
99999	10100000	100
100000	10100001	109

join——Single foreign key

SQL computing structured text data 10 (Join between files)

Example： Products information and sales information are stored in two tables. Please calculate the total sales of products with sales quantity less than 10.

	A	B
1	$select sum(S.quantity*P.Price) as total from E:/txt/Sales.txt as S join E:/txt/Products.txt as P on S.productid=P.ID where S.quantity<=10	/join，filter，aggregate

A1 result:

Index	total
1	142740.18000000008

join——Single layer multi foreign keys

SQL computing structured text data 11 （Join between files）

Example: Query the employees of HR department in California State

	A	B
1	$select e.NAME as NAME from E:/txt/EMPLOYEE_J.txt as e join E:/txt/DEPARTMENT.txt as d on e.DEPTID=d.DEPTID join E:/txt/STATE.txt as s on e.STATEID=s.STATEID where d.NAME='HR' and s.NAME='California'	/Single layer multi foreign keys join

A1 result:

Index	NAME
1	Gabriel
2	Megan

join——Multi-layer foreign keys

SQL computing structured text data 12 （ Join between files ）

Example：Look up employees in New York state whose manager is in California state.

	A	B
1	$select e.NAME as ENAME from E:/txt/EMPLOYEE.txt as e join E:/txt/DEPARTMENT.txt as d on e.DEPT=d.NAME join E:/txt/EMPLOYEE.txt as emp on d.MANAGER=emp.EID where e.STATE='New York' and emp.STATE='California'	/Multi-layer foreign keys join

A1 result:

Index	ENAME
1	Jessica
2	Alexis
3	Cameron
4	Ashley
5	Brandon
6	Grace
7	Jacob
8	William
9	Matthew
10	Emily

Subquery

SQL computing structured text data 13 (Subquery)

Example: Find out the department with the youngest department manager

	A
1	$select DEPT from (select emp.BIRTHDAY as BIRTHDAY,emp.DEPT as DEPT from E:/txt/DEPARTMENT.txt as dept left join E:/txt/EMPLOYEE.txt emp on dept.MANAGER=emp.EID) where BIRTHDAY=(select max(BIRTHDAY) from (select emp1.BIRTHDAY as BIRTHDAY from E:/txt/DEPARTMENT.txt as dept1 left join E:/txt/EMPLOYEE.txt as emp1 on dept1.MANAGER=emp1.EID))

A1 result:

Index	DEPT
1	Finance

Command line

Command line execution of simple SQL (absolute path)

Command line cd to directory esProc/bin的 (where esprocx.exe is located)， execute the script in the following format：.\esprocx+space+" -r" +space+" SQL" .

Example： Calculate the average salary of each department

Command line content

```
.\esprocx -r  "select DEPT,avg(SALARY) from E:/txt/EMPLOYEE.txt group by DEPT"
```

```
PS E:\esproc\esProc\bin> .\esprocx -r "select DEPT,avg(SALARY) from EMPLOYEE.txt group by DEPT"
```

Running result

```
Administration  10000.0
Finance 7395.833333333333
HR      7263.1578947368425
Marketing       7409.090909090909
Production      7285.714285714285
R&D     8241.379310344828
Sales   7286.096256684492
Technology      7319.148936170212
```

Command line

Command line execution of simple SQL (Relative path)

Example： Calculate the average salary of each department

Command line content

```
.\esprocx -r "select DEPT,avg(SALARY) from EMPLOYEE.txt group by DEPT"
```

```
PS E:\esproc\esProc\bin> .\esprocx -r "select DEPT,avg(SALARY) from EMPLOYEE.txt group by DEPT"
```

Note: the processed file can be an absolute path, or it can be located in the main directory or search directory.

Running result

```
Administration  10000.0
Finance 7395.833333333333
HR      7263.1578947368425
Marketing       7409.090909090909
Production      7285.714285714285
R&D     8241.379310344828
Sales   7286.096256684492
Technology      7319.148936170212
```

The main directory and search directory can be set in the environment tab of the program menu options, as shown in the figure:

CONTENTS

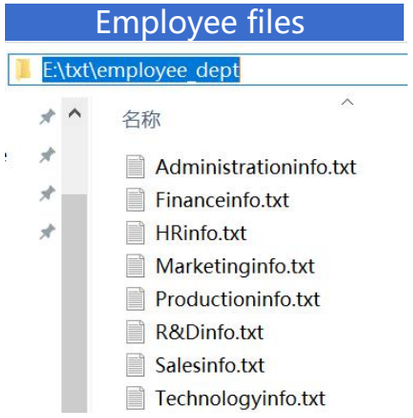
05

Merge and split

Merge and split

Multi file merge 1

Example: The employee data of each department is stored in different files in the same directory. Please merge the employee data and sort it according to the employee number, and then export it.



	A	B
1	=directory@p("E:/txt/employee_dept")	/List files with full pathnames in the file directory
2	=A1.(file(~).import@t())	/Read employee data of each department
3	=A2.conj().sort(EID)	/Merge and sort
4	=file("E:/txt/EMPLOYEE.txt").export@t(A3)	/Export

A1

Index	Member
1	E:\txt\employee_dept\Administrationinfo.txt
2	E:\txt\employee_dept\Financeinfo.txt
3	E:\txt\employee_dept\HRinfo.txt
4	E:\txt\employee_dept\Marketinginfo.txt
5	E:\txt\employee_dept\Productioninfo.txt
6	E:\txt\employee_dept\R&Dinfo.txt
7	E:\txt\employee_dept\Salesinfo.txt
8	E:\txt\employee_dept\Technologyinfo.txt

A2

Index	Member
1	[[18,Jonathan,Moore, ...],[20,Alexis,Alle
2	[[2,Ashley,Wilson, ...],[13,Daniel,Davis,
3	Emily,Smith, ...],[9,Victoria,Davis, ...]
4	Megan,Wilson, ...],[17,Hannah,Johr
5	6,Christopher,He
6	[[1,Rebecca,Moore,
7	[[3,Rachel,Johnson
8	[[55,Olivia,Anderso

A3

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
6	6	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000
7	7	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000
8	8	Megan	Wilson	F	California	1979-04-19	1984-04-19	Marketing	11000

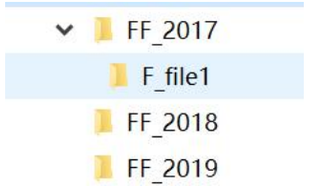
Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	18	Jonathan	Moore	M	Florida	1971-03-07	2000-03-07	Administrat...	7000
2	20	Alexis	Allen	F	Florida	1977-08-07	2007-08-07	Administrat...	16000
3	26	Timothy	Miller	M	Florida	1977-12-24	2007-12-24	Administrat...	5000
4	42	Michael	Jones	M	Pennsylva...	1978-08-20	2008-08-20	Administrat...	12000

Merge and split

Multi file merge 2

Example: Read multi-level directory recursively and merge files under the directory.

File directory



File content

FF_file1	1
FF_file1	2
FF_file1	3

	A	B
1	=directory@p(path)	/List the full directory of file names in the directory
2	=A1.(file(~).import())	/Import the file in the root directory
3	=A2.conj()	/Merge result
4	=file("d:\\result.txt").export@a(A3)	/Export in an appended way
5	=directory@dp(path)	/List directories under the directory
6	>A5.(call("E:/esproc_test/readfiles.dfx",~))	/Call this script recursively

Merge result

FF_file1	1
FF_file1	2
FF_file1	3
FF_file2	4
FF_file2	5
FF_file2	6
FF_file3	7
FF_file3	8
FF_file3	9
FF_file1/F_file1	1
FF_file1/F_file1	2
FF_file1/F_file1	3
FF_file1/F_file2	4

A1

Index	Member
1	D:\file\FF_file1.txt
2	D:\file\FF_file2.txt
3	D:\file\FF_file3.txt

A2

Index	Member
1	[[FF_file1,1],[FF_file1,2],[FF_file1,3]]
2	[[FF_file2,4],[FF_file2,5],[FF_file2,6]]
3	[[FF_file3,7],[FF_file3,8],[FF_file3,9]]

A3

Index	_1	_2
1	FF_file1	1
2	FF_file1	2
3	FF_file1	3
4	FF_file2	4
5	FF_file2	5
6	FF_file2	6
7	FF_file3	7
8	FF_file3	8
9	FF_file3	9

A5

Index	Member
1	D:\file\FF_2017
2	D:\file\FF_2018
3	D:\file\FF_2019

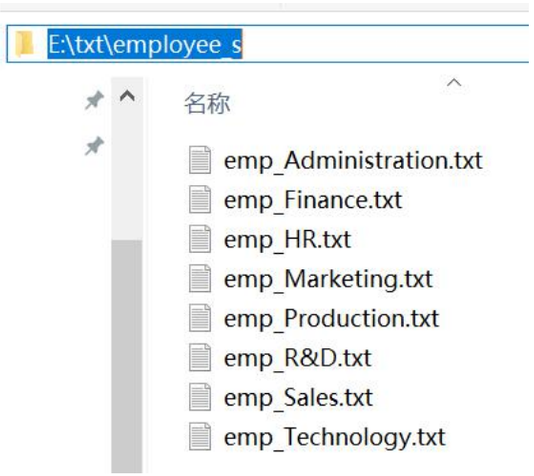
Merge and split

Small file split 1

Example： Write employee information into different files by department.

	A	B
1	=file("E:/txt/EMPLOYEE.txt").import@t()	/Read employee information
2	=A1.group(DEPT)	/Group by department
3	=A2.(file("E:/txt/employee_s/emp_"+~.DEPT+".txt").export@t(~))	/Name files and export

File directory



A1

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
6	6	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000
7	7	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000
8	8	Megan	Wilson	F	California	1979-04-19	1984-04-19	Marketing	11000

A2

Index	Member
1	[[18,Jonathan,Moore, ...],[20,Alexis,Allen, ...],[26,...
2	[[2,Ashley,Wilson, ...],[13,Daniel,Davis, ...],[23,J...
3	[[4,Emily,Smith, ...],[9,Victoria,Davis, ...
4	[[8,Megan,Wilson, ...],[17,Hannah,Joh...
5	[[16,Christopher,Hernandez, ...],[19,Samantha,...
6	[[1,Rebecca,Moore, ...],[5,Ashley,Smith, ...],[10,...
7	[[3,Rachel,Johnson, ...],[6,Matthew,Johnson, ...]...
8	[[55,Olivia,Anderson, ...],[56,Jacob,Smith, ...],[8...

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	18	Jonathan	Moore	M	Florida	1971-03-07	2000-03-07	Administrat...	7000
2	20	Alexis	Allen	F	Florida	1977-08-07	2007-08-07	Administrat...	16000
3	26	Timothy	Miller	M	Florida	1977-12-24	2007-12-24	Administrat...	5000
4	42	Michael	Jones	M	Pennsylva...	1978-08-20	2008-08-20	Administrat...	12000

Merge and split

Small file split 2

Example: Data with and without missing values is split into two files.

	A	B
1	=file("E:/txt/EMPLOYEE_nan.txt").import@t()	/Import data
2	=[true,false]	/Make sure two groups are divided
3	=A1.align@a(A2,~.array().pos(null)>0)	/Divide the data with or without missing values into two groups
4	=A3.(file("E:/txt/employee_N_s/employee_"+["NA","NO_NA"](#)+".txt").export@t(~))	
	/Export the two groups of data respectively	

A1

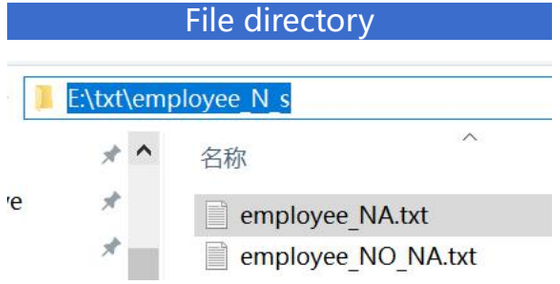
Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
175	175	Jasmine	Smith	F	Pennsylva...	1976-03-23	2005-07-01	(null)	7000.0
176	176	Joshua	Miller	M	Mississippi	1979-07-24	2004-05-01	HR	10000.0
177	177	Megan	Johnson	F	Missouri	1978-03-11	(null)	HR	5000.0

A3

Index	Member
1	[[16,Christopher,, ...],[17,Hannah,Johnson, ...],[23,Joseph,, ...], ...
2	[[1,Rebecca,Moore, ...],[2,Ashley,Wilson, ...],[3,Rachel,Johnso...

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	16	Christopher	(null)	M	Florida	1979-06-27	2007-06-27	Production	9000.0
2	17	Hannah	Johnson	F	Texas	(null)	2006-07-19	Marketing	4000.0
3	23	Joseph	(null)	M	California	1983-08-27	2003-08-27	Finance	6000.0

File directory



Merge and split

Large file split 1

Example： Write employee information into different files by department.

	A	B
1	=file("E:/txt/EMPLOYEE.txt").cursor@t()	
2	for A1,100	=A2.group(DEPT)
3		=B2.(file("E:/txt/EMPLOYEE/EMP_"+~.DEPT+".txt").export@at(~))
	/Read the file with cursor, fetch data by loop, and process the data retrieved each time according to the processing method of small file, but use @a by appending when exporting.	

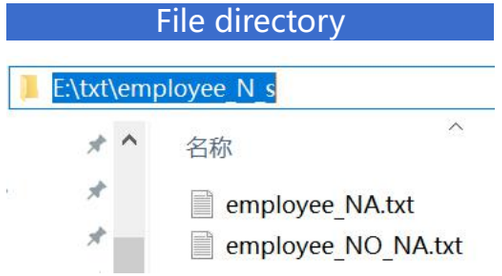
A2 and B2 in the first loop

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	Member
1	[[18,Jonathan,Moore, ...],[20,Alexis,Allen, ...],[26,Timothy,Miller, ...], ...]
2	[[2,Ashley,Wilson, ...],[13,Daniel,Davis, ...],[23,Joseph,Turner, ...], ...]
3	[[4,Emily,Smith, ...],[9,Victoria,Davis, ...],[51,Madison,Willia
4	[[8,Megan,Wilson, ...],[17,Hannah,Johnson, ...],[21,Jacob,I
5	[[16,Christopher,Hernandez, ...],[19,Samantha,Williams, .

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	18	Jonathan	Moore	M	Florida	1971-03-07	2000-03-07	Administrat...	7000
2	20	Alexis	Allen	F	Florida	1977-08-07	2007-08-07	Administrat...	16000

File directory



Merge and split

Large file split 2

Example： Data with and without missing values is split into two files.

	A	B
1	=file("E:/txt/EMPLOYEE_nan.txt").cursor@t()	
2	=[true,false]	/Make sure two groups are divided each time
3	for A1,100	=A3.align@a(A2,~.array().pos(null)>0)
4		=B2.(file("E:/txt/EMPLOYEE_N/EMPLOYEE_"+["NA","NO_NA"](#)+".txt").export@at(~))
	/Read the file with cursor, fetch data by loop, and process the data retrieved each time according to the processing method of small file, but use @a by appending when exporting.	

A3 and B3 in the first loop

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000.0
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000.0
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000.0
4	4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000.0
5	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000.0

File directory

Index	Member
1	[[16,Christopher,, ...],[17,Hannah,Johnson, ...],[23,Joseph,, ...] ...
2	[[1,Rebecca,Moore, ...],[2,Ashley,Wilson, ...],[3,Rachel,Johnso...

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	16	Christopher	(null)	M	Florida	1979-06-27	2007-06-27	Production	9000.0
2	17	Hannah	Johnson	F	Texas	(null)	2006-07-19	Marketing	4000.0
3	23	Joseph	(null)	M	California	1983-08-27	2003-08-27	Finance	6000.0
4	27	Alexis	Jones	F	California	1983-12-27	(null)	Marketing	10000.0

THANKS

Innovation makes progress!

www.raqsoft.com