SQL Enhancer

esProc · Issued by Raqsoft

www.raqsoft.com

Do this kind of query often?

SELECT ... FROM ...

- Which stocks have risen for five consecutive days this month?
- What is the median revenue per region?
- What is the growth in the three months with the best performance?
- List the students who are in Top 10 in every subject?
- By what month will sales of each salesman reach 500000?

......

Obsessed with these problems?

Dozens or hundreds of lines of SQL, with layers of nesting. Hard to understand even by the writer later on.

It will take several steps to solve the problem, but cannot write stored procedure for the database.

SQL debugging is hard, and it can't be tracked step by step. The only way is to print out temporary results. The efficiency is too low.

The window function and exists seem to be easy to use, but the calculation result is different from the expectation.

......

Why is SQL not suitable for writing complex queries?

Data is order-less

It's difficult to realize order-related calculation, and the sequence number need to be calculated.

Incompletely set-oriented

Do not have explicit set, unable to keep set data, and force aggregation while grouping.

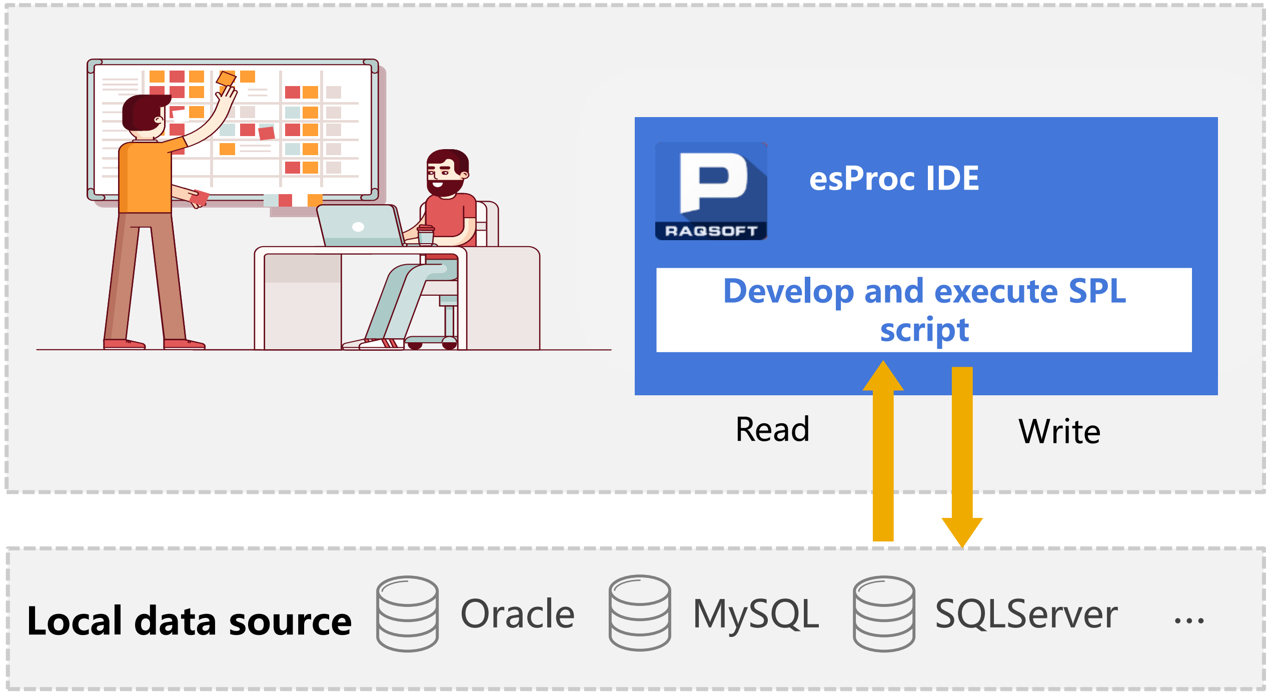
Simple definition of join

The join operation is very confusing and error prone when it involves many tables.

Do not advocate step by step

It does not directly support step-by-step calculation, and stored procedures cannot be used in many scenarios.

Free computing with esProc



Fast and free implementation of temporary computation with unified syntax

Desktop level tool, ready to use, simple environment configuration!

Agile syntax

Count the longest consecutively rising trading days for a stock.

Complicated and difficult to understand

SQL

```
SELECT max(continuousDays)-1 FROM
        (SELECT count(*) continuousDays FROM
                (SELECT SUM(changeSign) OVER ( ORDER BY tradeDate) unRiseDays FROM
                ( SELECT tradeDate,
                        CASE WHEN  closePrice>LAG(closePrice) OVER( ORDER BY tradeDate THEN 0 ELSE 1 END
                        changeSign
                FROM stock ))
        GROUP BY unRiseDays)
```

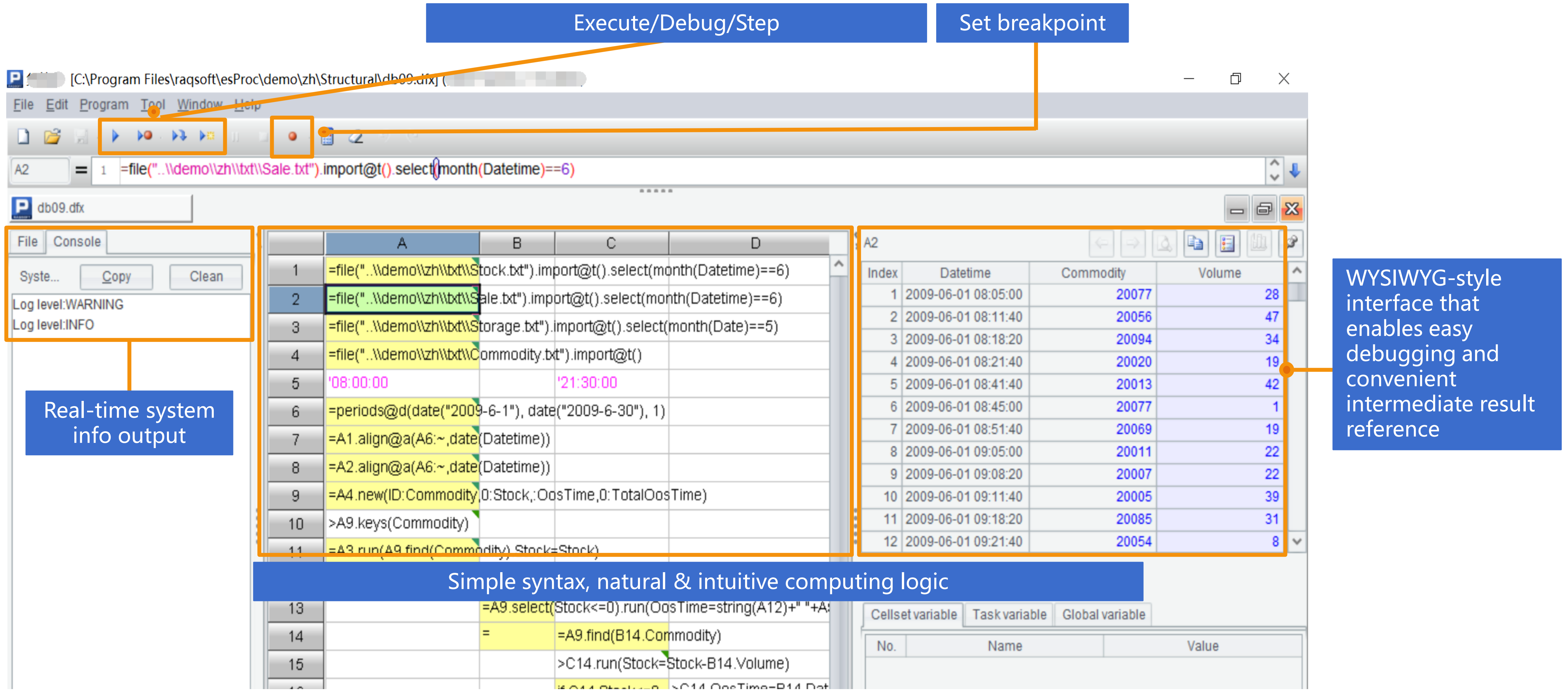
SPL

	A
1	=orcl.query("select * from stock order by trading day")
2	=A1.group@i(closePrice<closePrice[-1]).max(~.len())

Simple and intuitive

Convenient development environment

Install and use immediately, with perfect debugging function


Execute/Debug/Step
Set breakpoint
Real-time system info output
Simple syntax, natural & intuitive computing logic
WYSIWYG-style interface that enables easy debugging and convenient intermediate result reference
[C:\Program Files\raqsoft\esProc\demo\zh\Structural\db09.dfx]
File Edit Program Tool Window Help
db09.dfx
File Console
Syste... Copy Clean
Cellset variable Task variable Global variable

```
A2 = 1 =file("..\\demo\\zh\\txt\\Sale.txt").import@t().select(month(Datetime)==6)
```

```
Log level:WARNING
Log level:INFO
```

	A	B	C	D
1	=file("..\\demo\\zh\\txt\\Stock.txt").import@t().select(month(Datetime)==6)			
2	=file("..\\demo\\zh\\txt\\Sale.txt").import@t().select(month(Datetime)==6)			
3	=file("..\\demo\\zh\\txt\\Storage.txt").import@t().select(month(Date)==5)			
4	=file("..\\demo\\zh\\txt\\Commodity.txt").import@t()			
5	'08:00:00		'21:30:00	
6	=periods@d(date("2009-6-1"), date("2009-6-30"), 1)			
7	=A1.align@a(A6:~,date(Datetime))			
8	=A2.align@a(A6:~,date(Datetime))			
9	=A4.new(ID:Commodity,0:Stock,:OosTime,0:TotalOosTime)			
10	>A9.keys(Commodity)			
11	=A3.run(A9.find(Commodity).Stock=Stock)			
13		=A9.select(Stock<=0).run(OosTime=string(A12)+" "+A		
14		=	=A9.find(B14.Commodity)	
15			>C14.run(Stock=Stock-B14.Volume)	

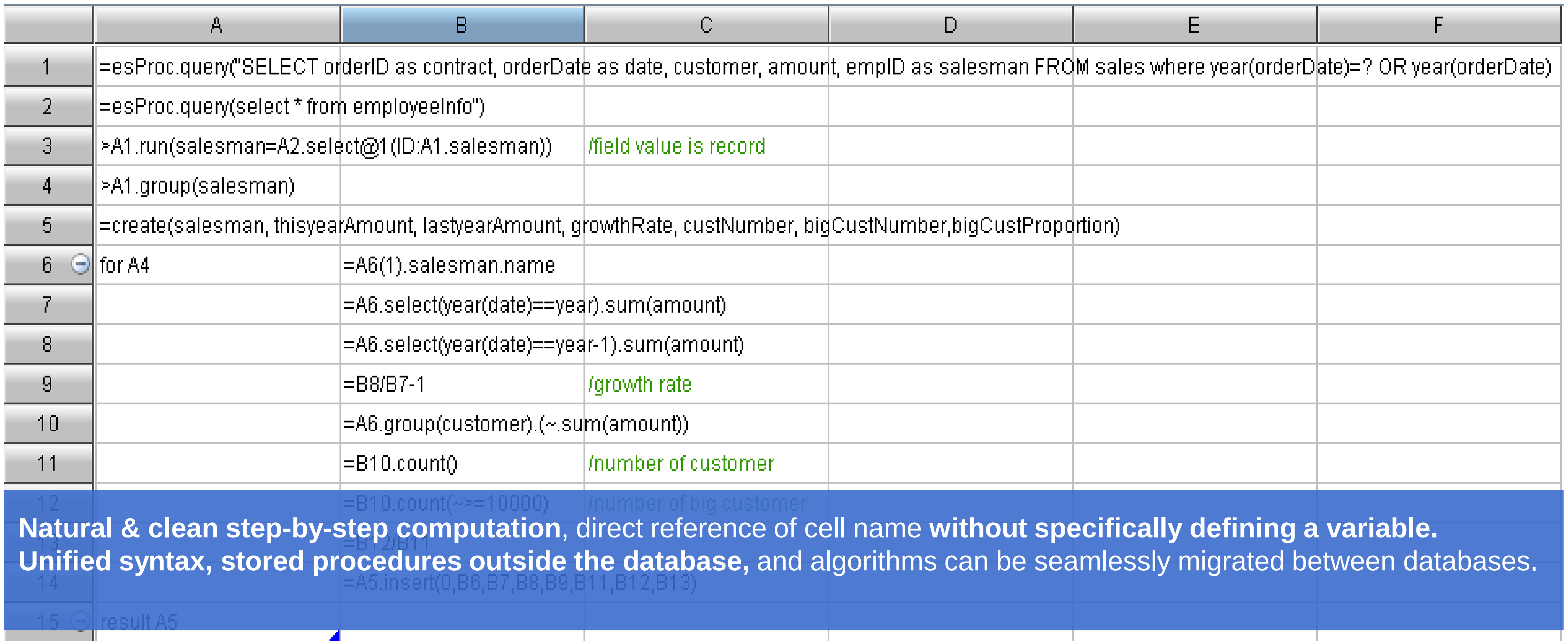
A2

Index	Datetime	Commodity	Volume
1	2009-06-01 08:05:00	20077	28
2	2009-06-01 08:11:40	20056	47
3	2009-06-01 08:18:20	20094	34
4	2009-06-01 08:21:40	20020	19
5	2009-06-01 08:41:40	20013	42
6	2009-06-01 08:45:00	20077	1
7	2009-06-01 08:51:40	20069	19
8	2009-06-01 09:05:00	20011	22
9	2009-06-01 09:08:20	20007	22
10	2009-06-01 09:11:40	20005	39
11	2009-06-01 09:18:20	20085	31
12	2009-06-01 09:21:40	20054	8

No.	Name	Value

Procedure-oriented computing

Reliable loop branch control



	A	B	C	D	E	F
1	=esProc.query("SELECT orderID as contract, orderDate as date, customer, amount, empID as salesman FROM sales where year(orderDate)=? OR year(orderDate)					
2	=esProc.query(select * from employeeInfo")					
3	>A1.run(salesman=A2.select@1(ID:A1.salesman))		/field value is record			
4	>A1.group(salesman)					
5	=create(salesman, thisyearAmount, lastyearAmount, growthRate, custNumber, bigCustNumber,bigCustProportion)					
6	for A4	=A6(1).salesman.name				
7		=A6.select(year(date)==year).sum(amount)				
8		=A6.select(year(date)==year-1).sum(amount)				
9		=B8/B7-1	/growth rate			
10		=A6.group(customer).(~.sum(amount))				
11		=B10.count()	/number of customer			
12		=B10.count(~>=10000)	/number of big customer			
13		=B12/B11				
14		=A5.insert(0,B6,B7,B8,B9,B11,B12,B13)				
15	result A5					

Natural & clean step-by-step computation, direct reference of cell name **without specifically defining a variable.**
Unified syntax, stored procedures outside the database, and algorithms can be seamlessly migrated between databases.

Example:

Calculate the moving average of monthly sales (one month before and one month after).

	A
1	=Sales.sort(month)
2	=A1.derive(Amount{-1,1}.avg():Moving average)

Find stocks that rise three days in a row.

	A
1	=Stock.sort(trading day)
2	=A1.group(code)
3	=A2.select((a=0,~.pselect(a=if(price>price[-1],a+1,0):3))>0)
4	=A3.(code)

Example:

Algorithm

For a company's organization table, query the subordinate organizations of the specified branch and list the names of its superior organizations. Multiple levels are separated by commas.

Data

ID	ORG_NAME	PARENT_ID
1	Head Office	0
2	Beijing Branch Office	1
3	Shanghai Branch Office	1
4	Chengdu Branch Office	1
5	Beijing R&D Center	2
…	…	…

Code

	A	B
1	=connect("db")	/Connect to database
2	=A1.query("select * from Organization")	/Query organization table
3	>A2.switch(PARENT_ID,A2:ID)	/The foreign key Parent_ID is mapped to the record where the ID is located to realize self join
4	=A2.select@1(ORG_NAME=="Beijing Branch Office")	/Select the record of Beijing Branch
5	=A2.new(ID,ORG_NAME,~.prior(PARENT_ID,A4) :PARENT)	/Create a new table consisting of ID, department name, and parent. The parent is obtained by recursively searching the records under Beijing branch through prior function.
6	=A5.select(PARENT!=null)	/Select the members whose parent exists, otherwise they are not subordinates of Beijing Branch.
7	=A6.run(PARENT=PARENT.(PARENT_ID.ORG_NAME).concat@c())	/concatenate all the parent names in the parent field, separated by commas.

Resource link

- SPL codes of common calculation http://doc.raqsoft.com/
- SPL / SQL syntax comparison http://www.raqsoft.com/dissociative-record.html

- Complex computing logic
  - Transposition http://c.raqsoft.com/article/1580980857594
  - Recursion http://c.raqsoft.com/article/1583481673878
  - Alignment Grouping http://c.raqsoft.com/article/1583484727988
  - TopN and variants http://c.raqsoft.com/article/1583485219299
  - Grouped subsets http://c.raqsoft.com/article/1583482802281

- Installation and free authorization http://c.raqsoft.com.cn/article/1571895350771