



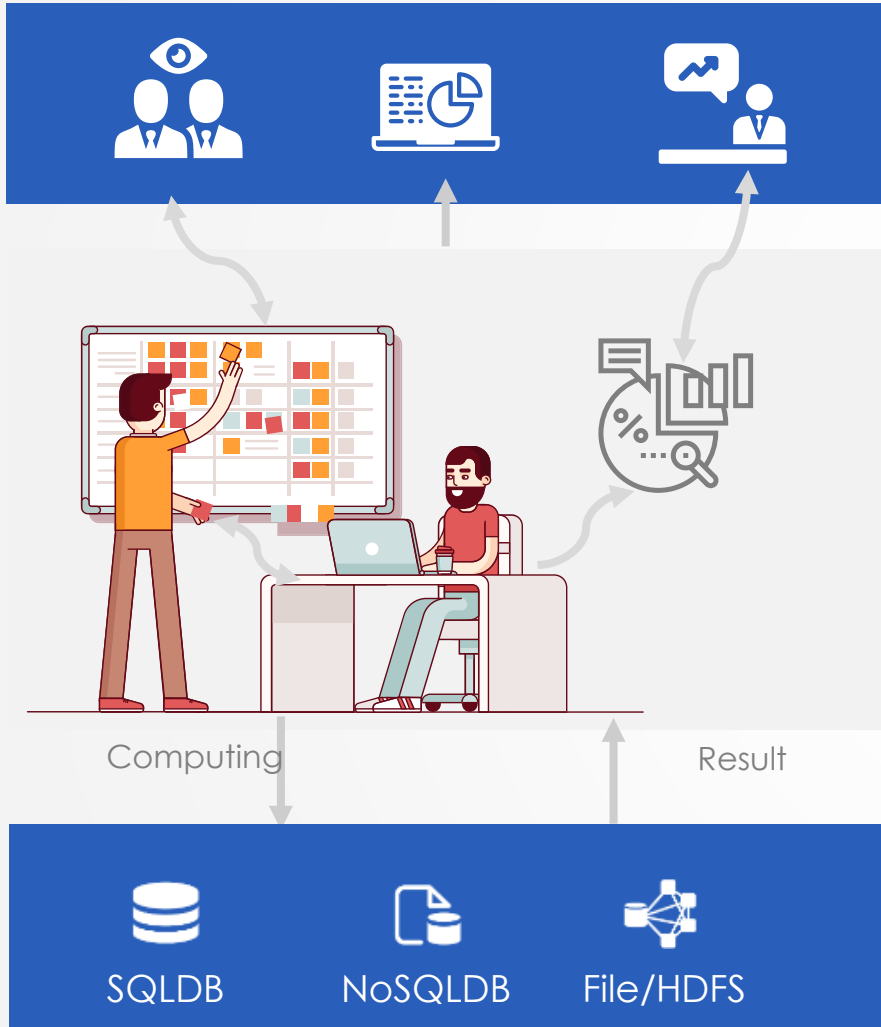
esProc

Offline Free computing

Issued by Raqsoft



widespread scenarios for offline computing



Data Cleaning and Loading (ETL)

Data preparation for reports and BI

Response to the temporary inquiry needs of business departments

Data arrangement before Artificial Intelligence/Data Mining Algorithms

Self-service analysis of data on hand

Test data generation

Experiments on optimizing scheme for big data computing

.....

Characteristics of Offline Free Computing



Demand is arbitrary and unpredictable



Diversity of raw external data



When necessary, it can be transformed into calculation in daily application.



Often done only once, lack of reusability

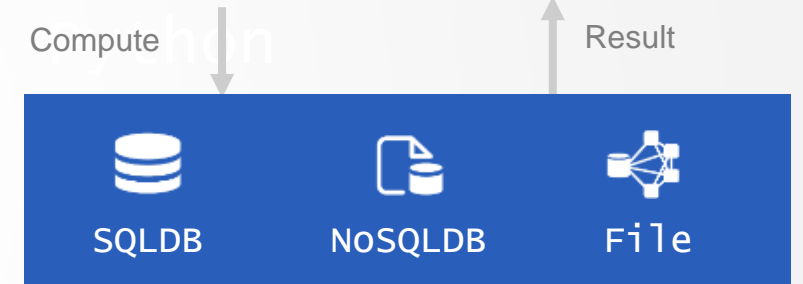
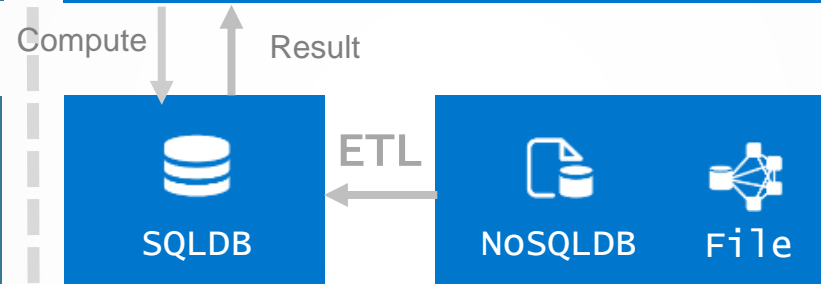
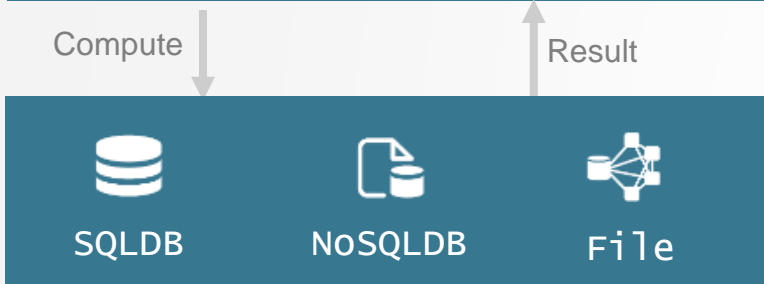
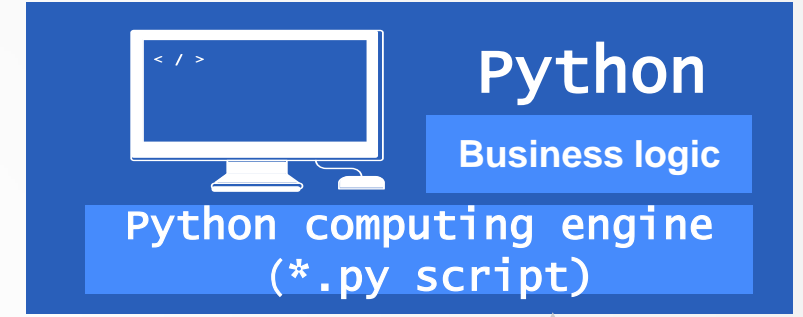
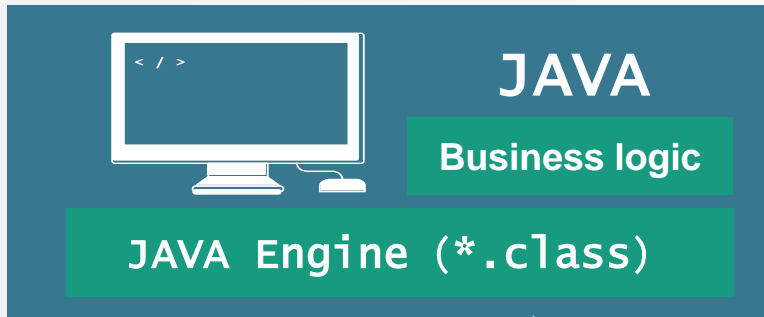


Multi-step process calculation



Mainly (Semi)structured data computing

What are the problems with the current commonly used technologies?



JAVA

- Computing class libraries are few and the code is tedious
- Professional programmers are needed
- Complex configuration of development environment
- Difficult to implement in case of diversified data sources computing

SQL

- Complex SQL and stored procedure, difficult to debug
- Tedious SQL for multi-step computing
- Computing is closed, so it is impossible to calculate data outside the database. Data sources without computational power must be loaded into database first

Python

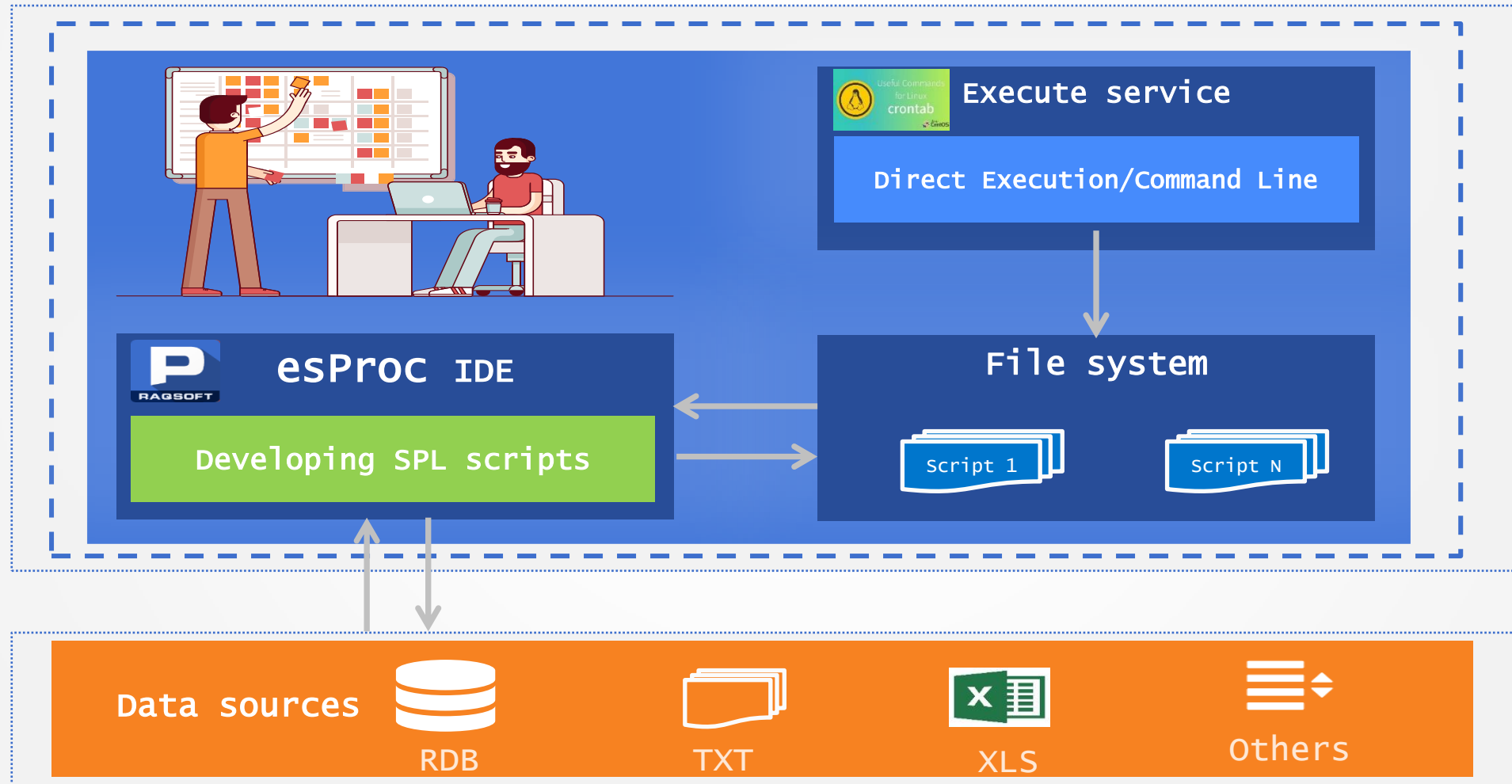
- Syntax is not designed specifically for structured data computing. Complex computing is not easy to develop.
- Incompatible between versions
- Troublesome installation and configuration of peripheral open source package
- Difficult to integrate and may be rewritten when converting to routine computing



Free Computing Using esProc

It does not depend on the computing power of the database, but directly calculates the diversity data for free analysis.

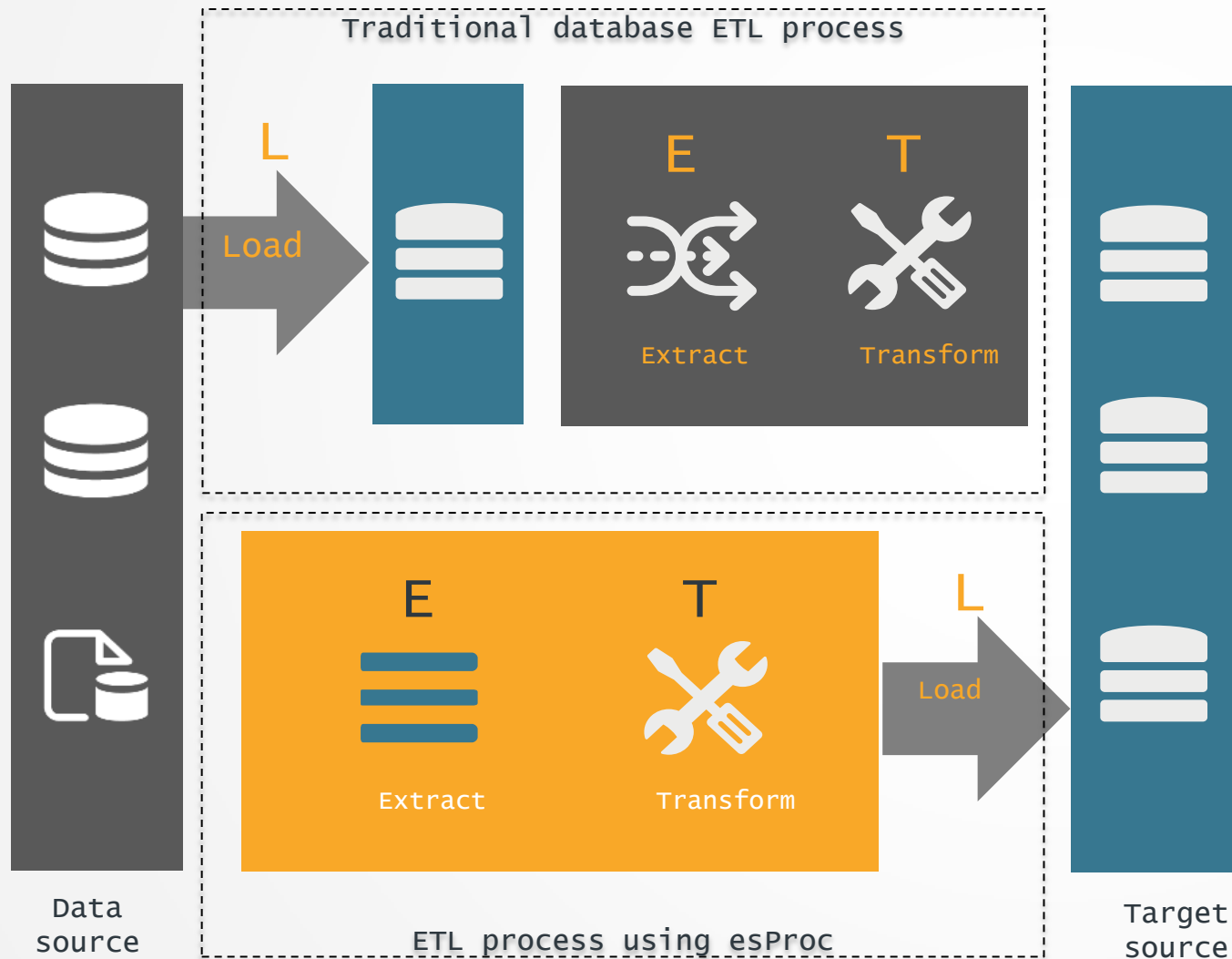
Desktop-level tools, ready-to-use, simple environment configuration!



Using esProc to implement ETL



Independent computing engine, reasonable realization of E, T, L process!

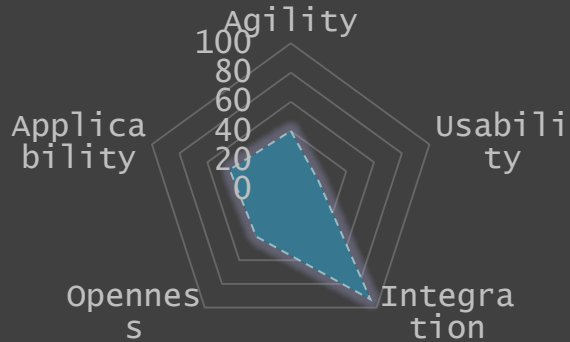


- Outside-database processing to reduce the burden of the database
- Reduce IO, shorten time window
- Simple and flexible implementation
- Allow multi-source mixed processing

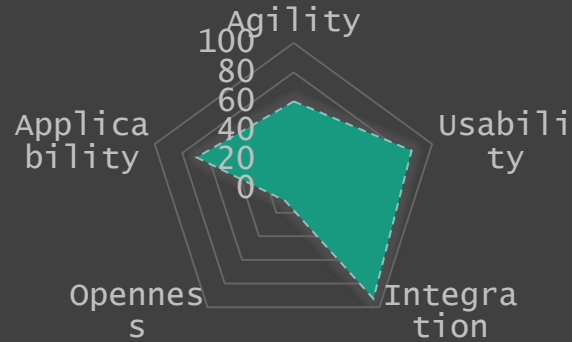
The Contrastive advantages of esProc and common languages



JAVA

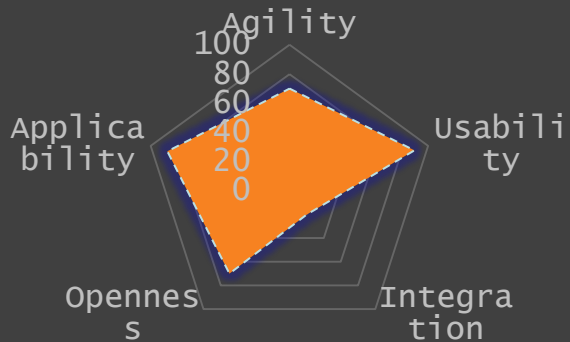


SQL

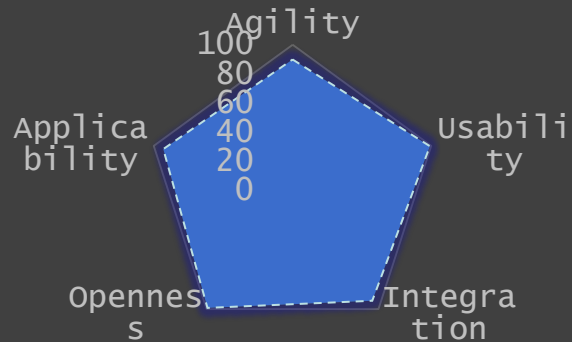


SPL Technical Advantages

- + Agility- Install-and-use, simple environment configuration, no additional plug-ins
- + Applicability - Perfect class libraries and procedure computing for complex data processing
- + Usability - Simple and intuitive syntax, easy to debug, no professional programmer required
- + Openness - Built-in multiple data source interfaces, direct computing
- + Integration- Seamlessly embedded into applications, easy to convert to daily computing



Python



SPL



Development environment

Install and use immediately, with perfect debugging function

Execute/Debug/Step

Set breakpoint

The screenshot shows a development environment with a menu bar (File, Edit, Program, Tool, Window, Help) and a toolbar with execution controls. The main window displays a code editor with the following code:

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

The console window shows the following log output:

```
System... Copy Clean  
Log level:WARNING  
Log level:INFO
```

The data table (A2) has the following columns: Index, Datetime, Commodity, and Volume. The data is as follows:

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

The code editor also shows the following code:

```
1 =file("../demo\zh\tx\Stock.txt").import@t().select(month(Datetime)==6)  
2 =file("../demo\zh\tx\Sale.txt").import@t().select(month(Datetime)==6)  
3 =file("../demo\zh\tx\Storage.txt").import@t().select(month(Date)==5)  
4 =file("../demo\zh\tx\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)
```

Real-time system info output

WYSIWYG-style interface that enables easy debugging and convenient intermediate result reference

Simple syntax, natural & intuitive computing logic



Agile syntax



Count the longest consecutively rising trading days for a stock

```
1 import pandas as pd
2 def iterate(col):
3     prev = 0;
4     res = 0;
5     val = 0;
6     for curr in col:
7         if curr - prev > 0:
8             res += 1;
9         else:
10            res = 0;
11            prev = curr;
12            if val < res:
13                val = res;
14            return val;
15 data = pd.read_excel( 'D:/Stock.xlsx' ,sheet_name=0).
    sort_values( 'Date' ).groupby( 'Company' )[ 'Price' ].apply(iterate);
```

Python

	A
1	=file("D:/Stock.xlsx").importxls@t().sort(Date).group(Company)
2	=0
3	=A1.max(A2=if(Price>Price[-1],A2+1,0))

SPL

Professional grammar in the field of set operation, code is more concise for the same process!



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	Natural & clean step-by-step computation, direct reference of cell name without					
13		=B12/B11				
14	specifically defining a variable					
14	=A5.insert(0,B6,B7,B8,B9,B11,B12,B13)					
15	result A5					



Multiple data source interfaces

- ✓ Commercial RDBMS: e, MS SQL Server, DB2, Informix
- ✓ Open source RDBMS: MySQL, PostgreSQL
- ✓ Open source NOSQL: MongoDB, Redis, Cassandra, Elasticsearch
- ✓ Hadoop: HDFS, HIVE, HBase
- ✓ Application software: SAP ECC, BW
- ✓ Files: Excel, JSON, XML, TXT
- ✓ Others: HTTP RESTful, Web Services, MDD that supports OLAP4j, AliCloud, ...

Built-in interface, ready-to-use

Free Computing Using esProc - Summary



Desktop level

The development environment is simple to configure, ready to use



Easy management

The algorithm is stored in the script file, which is easy to manage



Openness

Built-in multiple data source interfaces, direct computing



Easy integration

Frameless, lightweight architecture easy to embed and integrate



Easy development

No need for environment configuration and application layer code reference



High performance

Rich class libraries and parallel computing power

Innovation makes progress!

