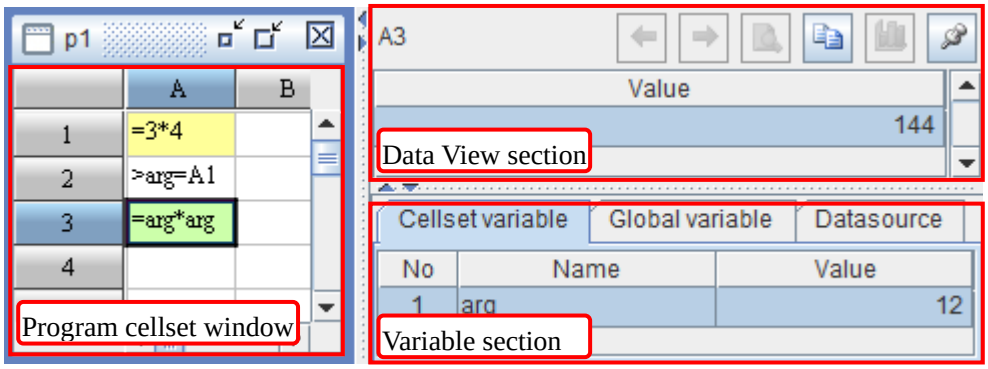
Getting started

1 Code in Cellset

esProc is a programming tool coding in the cellset. Every statement is contained in a cell, which is similar to **Excel**.



On the left is the **Program Cellset** window, the active program cellset you are operating on is shown here; The top right area is the **Data View** section and the bottom right area is the **Variable** section.

The esProc expression follows the Java conventions to the utmost, such as **%** for modulus, **==** and **!=** for equaling and not equaling, the double quotation marks are required to enclose the strings, the **** is used as the conversion marks, and the **&&**, **||**, and **!** are logic operator of **AND**, **OR**, and **NOT**.

Depending on the code in the cells, the cell may be in various colors.

	A	B	C
1	5	esProc	2013-11-5
2	'D:\file\dfx		
3	=A1*7	=interval(C1,now())	
4	>arg1=4	/Comment Info	

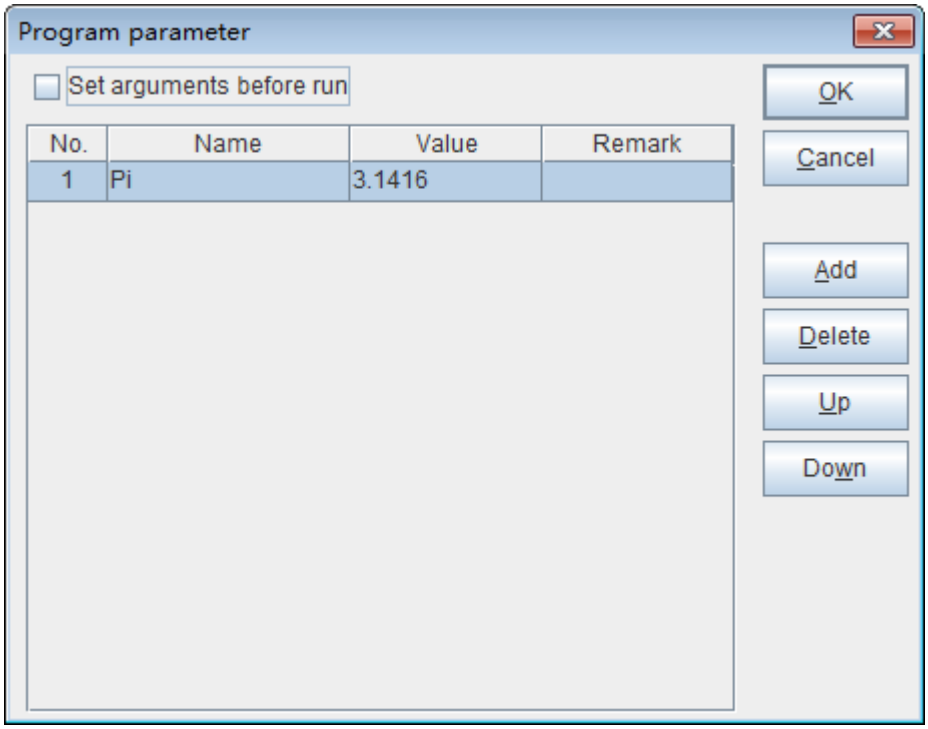
The pink indicates the text is of constant type, the black indicates the text is executable code, and the green indicates the text is comments.

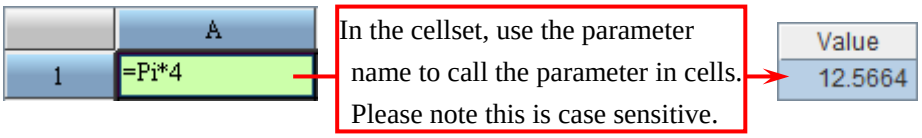
If the cell contains value after computing, then the background color will turn to light yellow. You can check the cell value in the Data View section.

In esProc, if any value is in the cell, then you can use the **Cell Name** directly as the variable name to call the cell value. This is similar to the Excel.

You can also use the named variable. The variable of this type is called **Cellset Variable**. You can check the cellset variable in the Cellset Variable section.

In the cellset file, you can define the **Cellset Parameter** by clicking the **Program** -> **Parameter** menu item. This can be defined in the Cellset Parameter window.





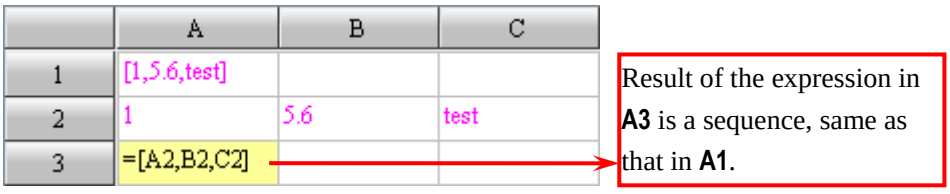
When setting the cellset parameters, you can also check options and set parameters before computing. Please note that the names of parameter and variable are **case sensitive** in esProc.

2 Introduction to Sequence and TSeq

Sequence is an ordered set consisting of some data, and the constitutive data of a sequence is called Member. Sequence is equivalent to the array in the high level language. The difference is that the member of sequence **does not** have to be of the same type. The sequence is the most common unit in esProc.

Some members enclosed in a **[]** forms a constant sequence. The sequence can be defined in an expression in a format similar to the constant sequence, and the cell can be used in an expression.

If no member exists in sequence, then you can use **[]** to define the empty sequence.



The cell values of **A1** and **A3** can be viewed in the Data View section.

Member
1
5.6
test

esProc inherits the data table concept of relational database, and defines it as Table Sequence or TSeq for short. Consistent with the concept of rational database, in esProc, every Table Sequence also has its own Data Structure, which consists of several Fields. The member of table sequence is also called as Record. TSeq is a sequence whose members are all records.

TSeq can be retrieved from database or converted from the data of text file. When using the data of text file, you can separate the data of the same row with Tab.

	A
1	=file("D:/files/txt/students.txt")
2	=A1.import@t()

Import the data in the text file as TSeq and the 1st row is the title

ID	Name	Gender	Age
1	Emily	F	17
2	Elizabeth	F	17
6	Zachary	M	19
8	Megan	F	16

Sequence consisting of records from table sequence is called **Record Sequence** or **RSeq** for short. The members of record sequence are not definitely from a same table sequence. The record sequence whose members are from a same table sequence is called as **Pure Record Sequence**.

3 Introduction to ISeq

The sequence making up of integers is **Integer Sequence** or **ISeq** for short. When defining the ISeq, you can also use the **[]** to enclose the member for defining the ISeq or use the expression to define it.

Member
1
2
3
4
5

	A
1	[1,2,3,4,5]
2	[5,3,-1,3,0]
3	=[10,3*4,5-2,A1.len()]

Generate ISeq with the expression in which the **A1.len()** is the length of sequence in **A1**

Member
5
3
-1
3
0

Member
10
12
3
5

In esProc, **to()** function is often used to define an ISeq of continuous increase or decrease.

Member
5
6
7
8

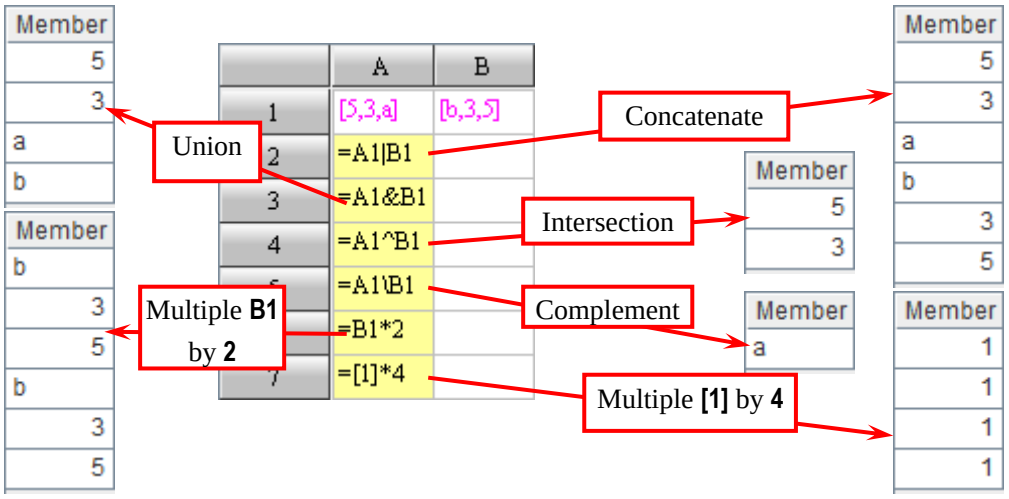
	A
1	=to(5,8)
2	=to(4)

Member
1
2
3
4

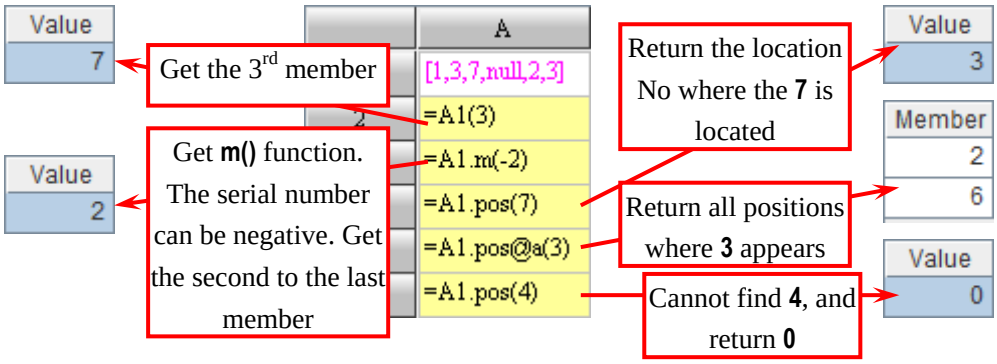
4 Use Sequence to Compute

In esProc, not only the ISeq is a sequence, but also the TSeq and RSeq are all sequence. The sequence computation is the basic computation of esProc.

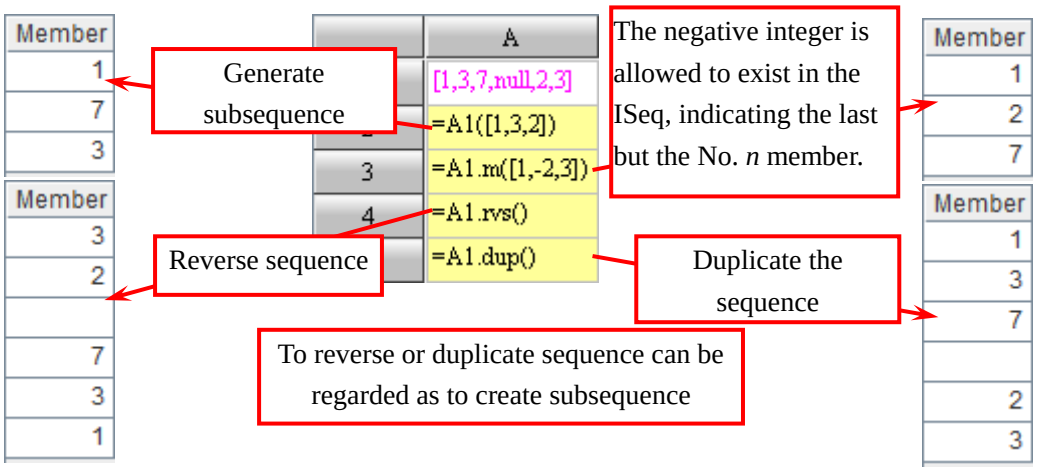
As a type of set, sequence can be used for common operations with the **|**, ****, **^**, **&**, and other operators. You can compute the concatenate, subtraction, intersection, and union of two sequences:



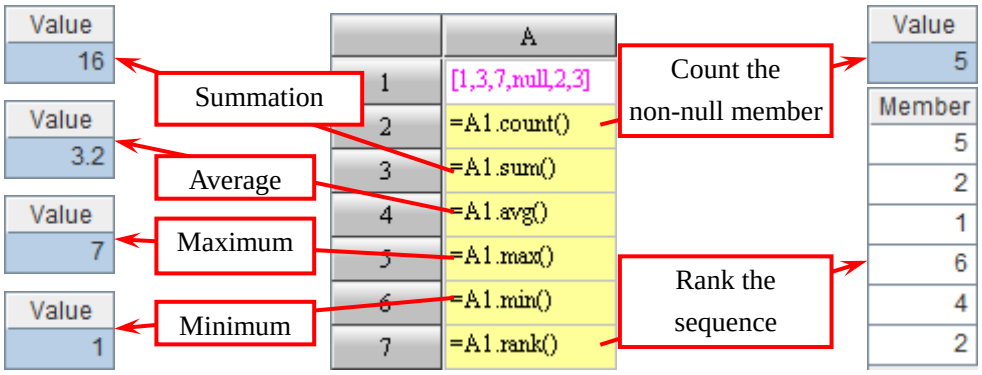
The member in a sequence is ordered. We can not only get or set the value of a specified member according to the member location, but also get the location of specified members:



The sequence composed of the members of another sequence is called the Subset of the latter sequence, which is similar to the definition of this Subset. In esProc, you can take the ISeq as the location-indicating serial number to generate a subsequence of this sequence.



In the sequence, you can perform various converge operations:



5 Retrieve TSeq from Database

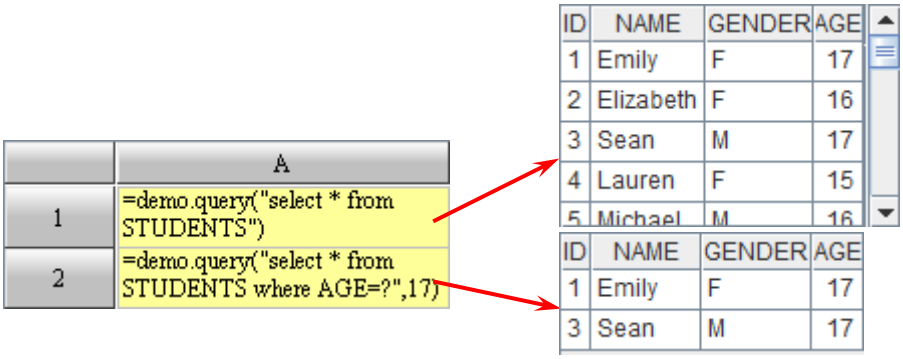
To retrieve TSeq from database, you will have to connect with the relational database in the Datasource Manager of esProc:

1) Click the **Tool** -> **Datasource connection** menu item, open the Datasource Manager.
2) Create a new datasource, and select the type.
3) Edit the datasource connection parameters and name the datasource.
4) Connect, and the Datasource Manager will display if connected successfully. Multiple databases can be connected at the same time in the Datasource Manager.

In esProc, you can connect the HSQL database demo for testing.

After the connections to the database are established successfully in the data source manager, you can simply use the datasource name in the cellset to access them directly.

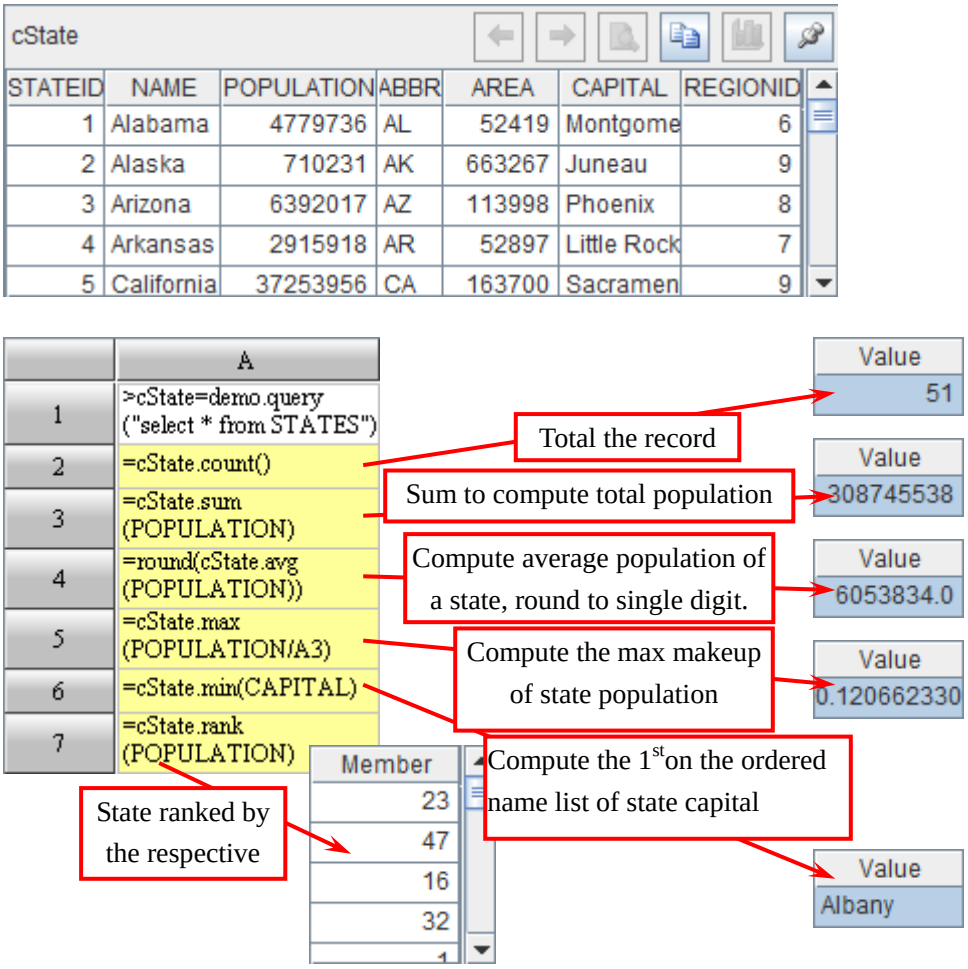
For example, you can retrieve the data composition TSeq from the database with the SQL statement.



6 Collect Statistics on TSeq

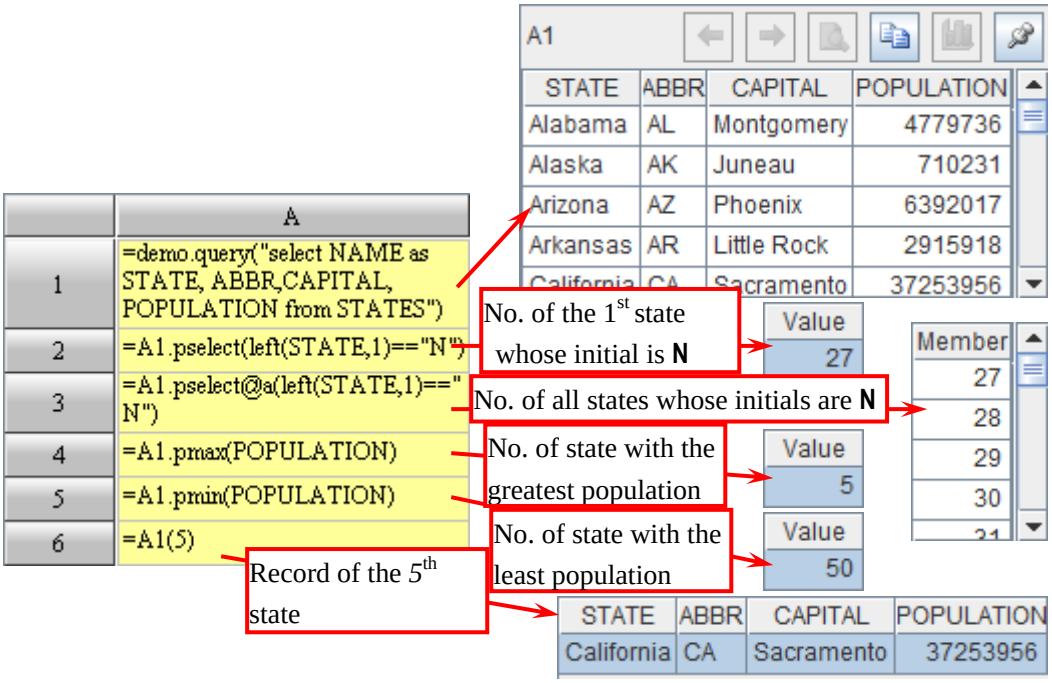
To collect statistics based on the data from database, we often perform the count, sum, min value computations, and other operations on the database record. In esProc, you can perform various converge computation in the TSeq, such as **count**, **sum**, **avg**, **max**, and **min**.

The cellset variable cState is a TSeq for storing information of various states. You can complete several converge computations like count and sum with the **count()**, **sum()**, and other functions.

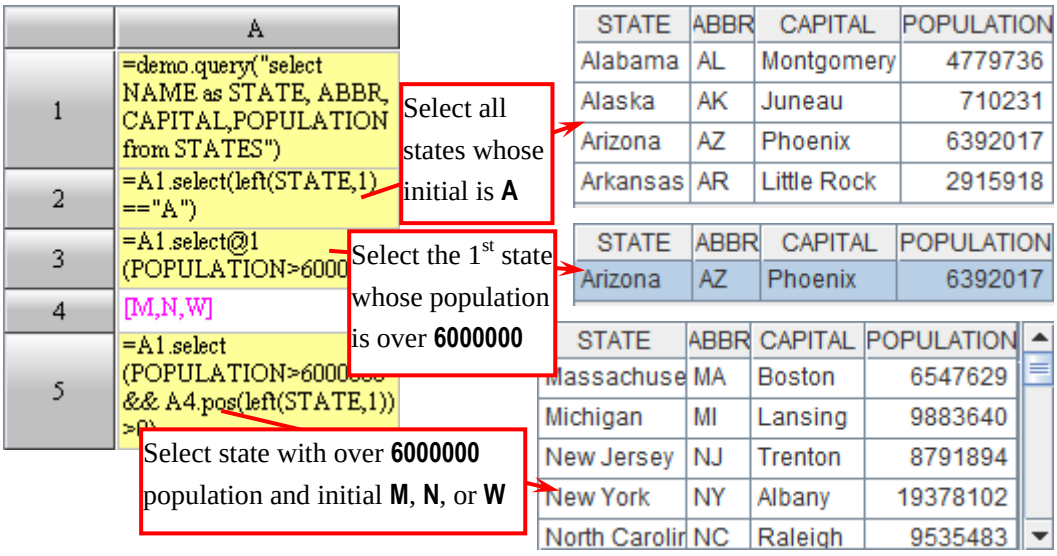


7 Locate and Filter the Record in TSeq

When analyzing the data in the TSeq, you will usually need to search for the records satisfying the conditions according to the requirements. In esProc, if there is a value in a cell, then the value of this cell can be referenced with the cell name.



In esProc, you can either directly select the record from the TSeq according to its position, or compute the position of records satisfying the conditions with the **pselect()** function, or specify the position of the max or min record as a result of a specified expression with the **pmax()**, **pmin()**, and other functions.



In the TSeq, you can use **select()** function to filter the data and pick out the record satisfying the specified conditions. After filtering, an RSeq composed of the corresponding records will be

returned.

In esProc, to generate a new TSeq, you can choose the desired column to compute based on the data from TSeq or RSeq.

With the **new()** function, you can select the necessary information from TSeq or RSeq to form a new TSeq.

	A
1	=demo.query("select EID,NAME,GENDER,BIRTHDAY from EMPLOYEE")
2	=A1.new(NAME:EName,if(GENDER=="M","Male","Female"):Gender)
3	=A1.new(EID,NAME,interval@y(BIRTHDAY,now()):Age)

EID	NAME	GENDER	BIRTHDAY
1	Rebecca	F	1974-11-20
2	Ashley	F	1980-07-19
3	Rachel	F	1970-12-17
4	Emily	F	1985-03-07
5	Ashley	F	1975-05-13

Retrieve the name and gender data to create a TSeq. The **NAME** field is named after **EName**, and the abbreviation will not be used for the gender.

Retrieve the employee No, and name. Compute the age of each employee, and then create the TSeq.

Of **A2** and **A3** the results are as follows:

EName	Gender
Rebecca	Female
Ashley	Female
Rachel	Female
Emily	Female
Ashley	Female

EID	NAME	Age
1	Rebecca	39
2	Ashley	33
3	Rachel	43
4	Emily	28
5	Ashley	38

In esProc, you can choose to add the computed column directly to the TSeq. The value in the computed column can be computed based on the data from other columns of TSeq. With **derive()** function, the computed column can be added for the TSeq.

	A
1	=demo.query("select EID,NAME,GENDER,BIRTHDAY from EMPLOYEE")
2	>A1.derive(interval@y(BIRTHDAY,now()):Age)

Add the **Age** field for the TSeq, and the age can be derived from **BIRTHDAY**

After using **derive()** function to add the computed column, the existing TSeq is changed. As you can see in **A1**, the TSeq has been changed:

EID	NAME	GENDER	BIRTHDAY	Age
1	Rebecca	F	1974-11-20	39
2	Ashley	F	1980-07-19	33
3	Rachel	F	1970-12-17	43
4	Emily	F	1985-03-07	28
5	Ashley	F	1975-05-13	38

8 Reference and Sort on Record in TSeq

Since the field of record has no data type requirement, you can assign any value. To assign the value to another record, you can conveniently implement the foreign key reference.

	A
1	=demo.query("select * from EMPLOYEE")
2	=demo.query("select NAME as STATE,ABBR,CAPITAL, POPULATION from STATES")
3	>A1.derive(A2.select@1 (STATE==A1.STATE):StateInfo)

Add **StateInfo** field to TSeq, and the field value is the corresponding records in the TSeq **A2**

After executing the derive command, in the TSeq of EMPLOYEE, the value of **StateInfo** field is the record of TSeq of A2 on which you can double click to view the **StateInfo** field:

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY	StateInfo
1	Rebecca	Moore	F	Californ	1974-11-2(	2005-03-11	R&D	7000	Californ
2	Ashley	Wilson	F	New Yo	1980-07-19	2008-03-16	Finan(	11000	New Yo
3	Rachel	Johnson	F	New Me	1970-12-17	2010-12-01	Sales	9000	New Me
4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000	Texas
5	Ashley	Smith	F	Texas	1975-05-1(	2004-07-30	R&D	10000	Texas

STATE	ABBR	CAPITAL	POPULATION
Texas	TX	Austin	25145561

In the TSeq or RSeq, you can sort the record according to the specified conditions. With the **sort()** function, you can re-sort the records in the TSeq, and return the RSeq after sorting.

	A
1	=demo.query("select * from EMPLOYEE").derive(interval@y (BIRTHDAY,now()):Age)
2	=A1.sort(STATE)
3	=A1.sort(Age:-1)
4	=A1.sort(Age:-1,NAME)

Sort the records by **STATE** ascendingly

Sort the records by **Age** descendingly

Sort the records by **Age** descendingly. Then sort those of the same age by **NAME** ascendingly

9 Grouping Record in TSeq

To collect statistics and analyze data often requires grouping the records of a TSeq or RSeq according to certain conditions.

The commonest group is the **Equal Group**. Distribute the records from a TSeq or RSeq to several groups according to the same field or expression.

With **group()** function, you can specify one or more conditions to group the values from TSeq or RSeq.

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.group(STATE)
3	=A1.group(STATE,GENDER)

Group **EMPLOYEE** records by its **STATE**

Group **EMPLOYEE** records by its **STATE** and **GENDER**

The **group()** function for the TSeq or RSeq is similar to the operation of group by in SQL. A sequence composed of multiple groups will be returned. By default, when using **group()** function, the record will be sorted before grouping according to the results of group expression.

In esProc, you can proceed with the group and summarization computation based on the grouping result.

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.group(STATE)
3	=A1.group(STATE,GENDER)
4	=A2.new(~.STATE:State,~.count():Count)
5	=A3.new(~.STATE:State,~.GENDER:Gender,~.count():Count)

Create a TSeq. Return the total employee amount of each state

State	Count
Alabama	4
Arizona	9
Arkansas	1
California	55
Colorado	6

Create a TSeq. Return respective total male and female employee amount of each state

State	Gender	Count
Alabama	F	3
Alabama	M	1
Arizona	F	7
Arizona	M	2
Arkansas	M	1

This differs with that of SQL greatly. SQL does not provide the explicitly data type of sets and the group results cannot be saved either. The SQL users have to group and summarize immediately after the group by action, and then the group result will be dropped, not allowing for any reuse.

Summarize after grouping and you can use **groups()** function to represent it briefly.

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.groups(STATE,GENDER;count(~):Count)

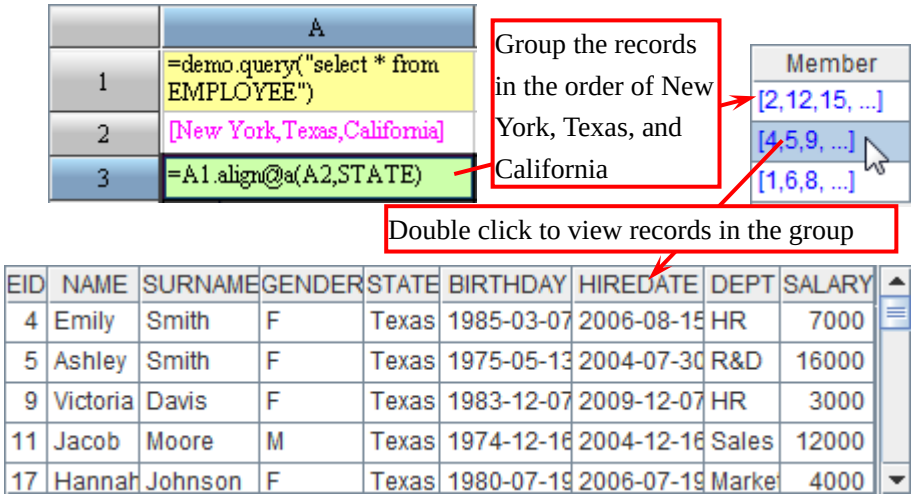
STATE	GENDER	Count
Alabama	F	3
Alabama	M	1
Arizona	F	7
Arizona	M	2
		1

After grouping the employees' data by **STATE** and **GENDER**, create a TSeq of total female and male employee amount of each state. This is similar to the result of **A5** in above example.

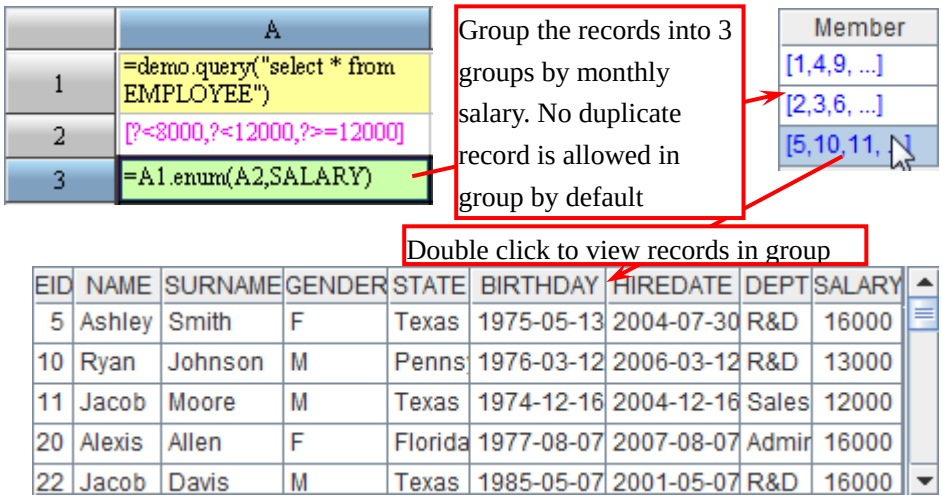
In esProc, the record in the TSeq can be grouped in a rather complicated way.

If you need to perform the equal group on the records in the TSeq or sequence in a specified

order, then the **align group** will be frequently used. With **align()** function, you can conveniently implement the align group.

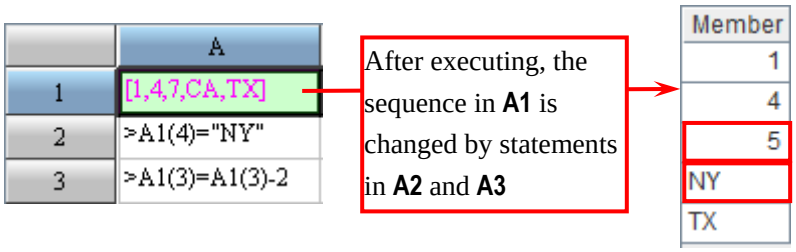


Enum group is to group the records according to various conditions. You can implement the enum group with the **enum()** function.

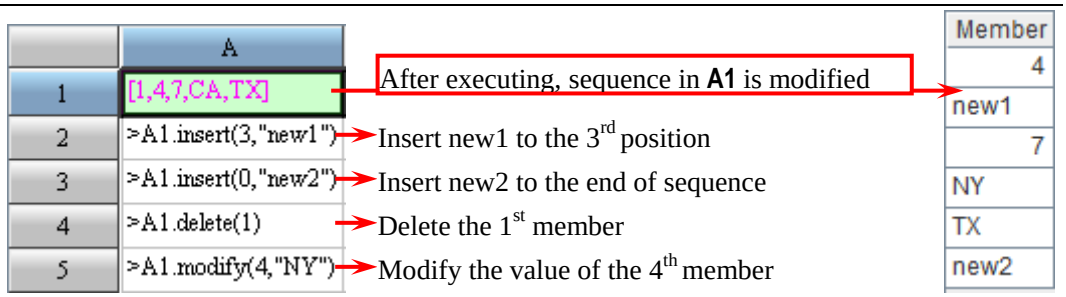


10 Modify Sequence and TSeq

The same syntax for array assignment in Java also applies to the sequence modification in esProc, that is, you can just assign directly to the member at appointed position to implement it.

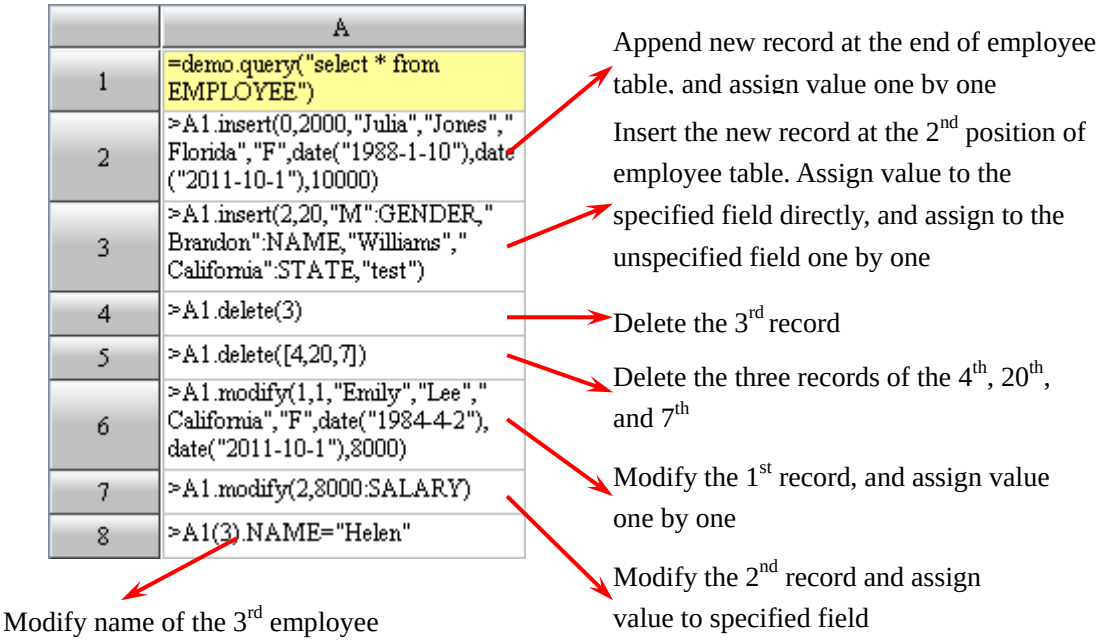


Besides, similar to SQL statement, you can also use the **insert, delete, modify,** and other functions to perform the insert, delete, modify, and other operations on the sequence.

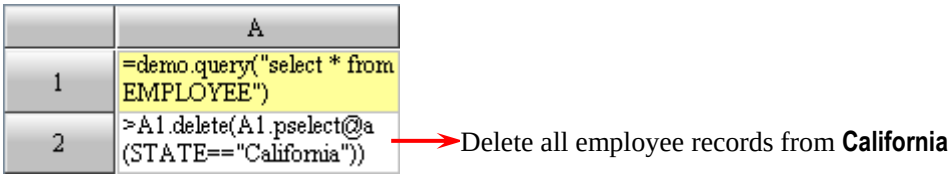


Unlike the other Sequence, the member of record in TSeq cannot be assigned directly. For example, if T is a TSeq, then the *T*(3)=*r* operation is wrong.

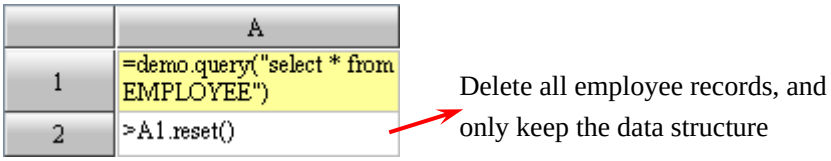
When modifying TSeq, you can only use **insert( ), delete( ), modify( )**, and other functions to modify the TSeq, or directly modify the record of this field.



As the expression in the **A5** indicates that you can use ISeq to indicate the position when deleting the records in the TSeq. Therefore, with the pickout function of **pselect()** in the TSeq, you can delete the records according to the specified conditions.



Use **reset()** function to delete all records in the TSeq and only keep the data structure.

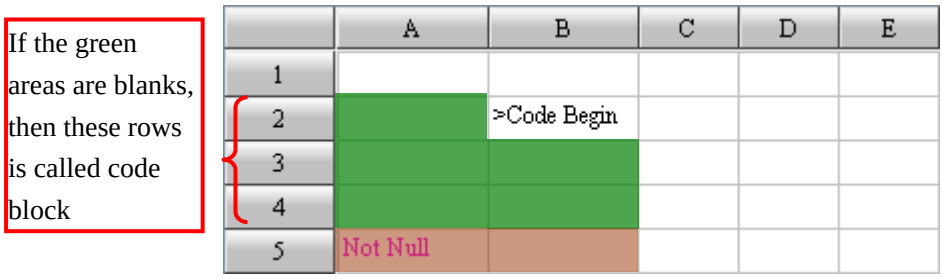


11 Flow of Program in esProc

In esProc, you can use various judgment and loop statements just as in the normal program language to implement the program design of choice structure and repetitive structure.

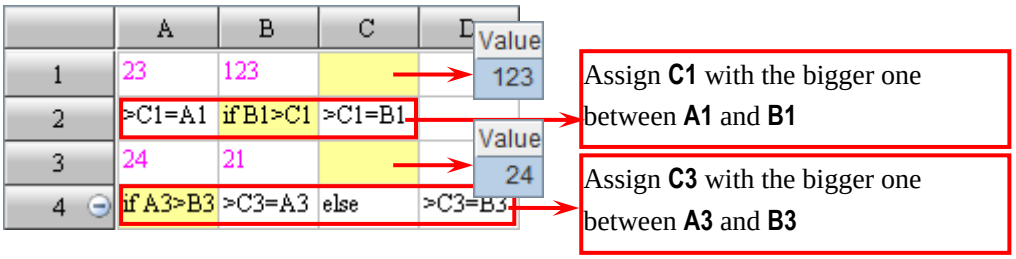
Unlike JAVA and other high level languages, esProc use the straightforward format of code block to specify the working range of statement, instead of using the symbol like **{}** or reserved words (BEGIN/END) to enclose the working range.

In the program cellset, a certain range of indented cells is called as **Code Block**, and the starting cell is **Master cell** of the code block.

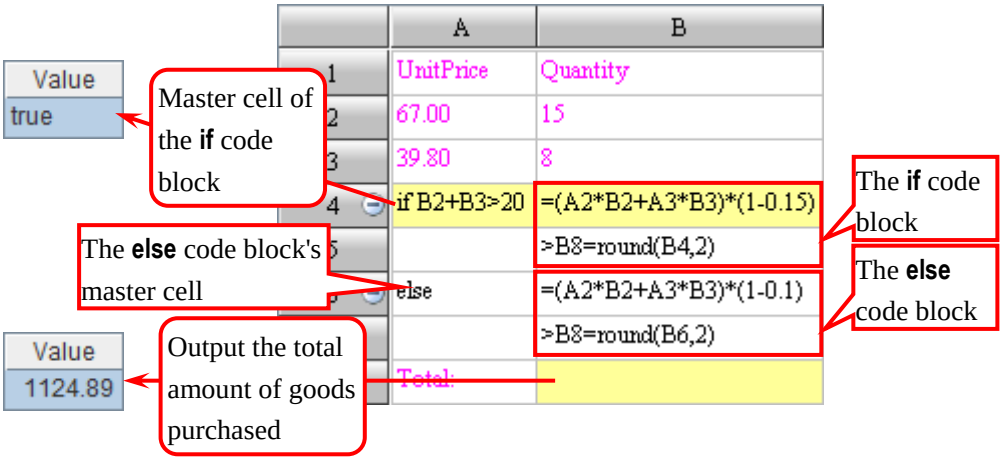


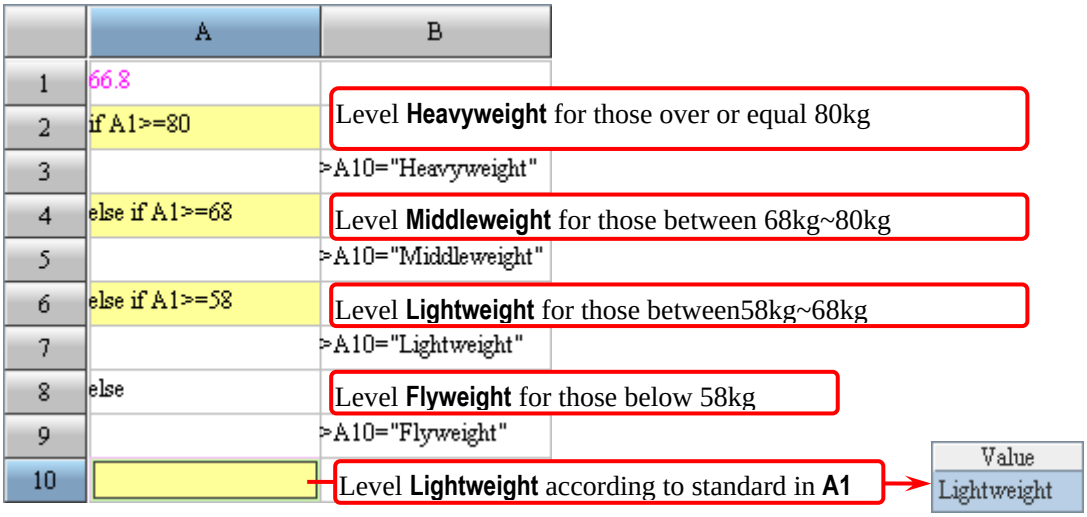
In esProc, the commonest statement of choice structure is the **if/else** judgment. There are 3 common styles of judgment statement: use **if** separately, **if**...**else**..., and **if**...**else if**...**else if**.....**else**. In esProc, you can use these structures according to your practical needs.

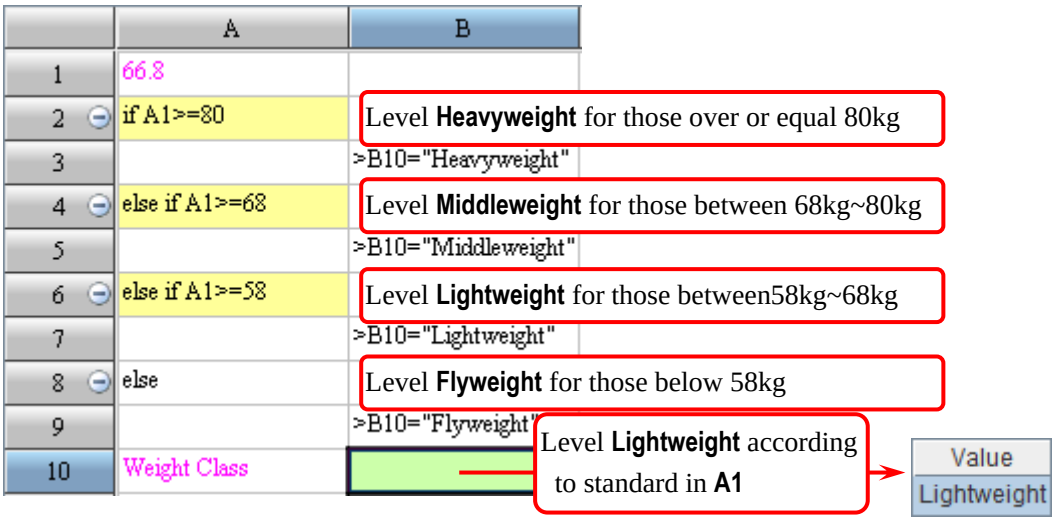
The judgment statement can be used in an individual line.



The judgment statement can be composed in multiple lines, and put the statement in the code block.

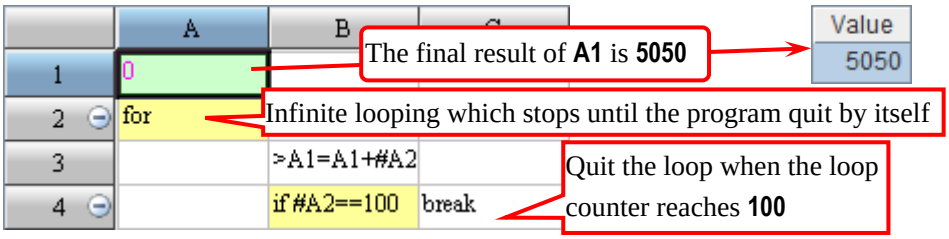




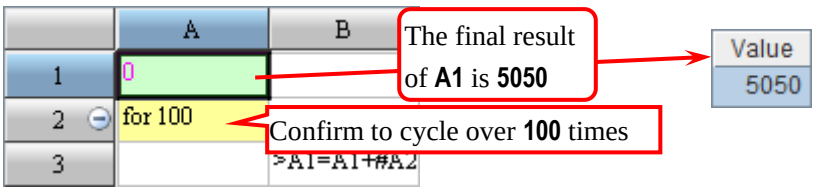


The most common loop statements in esProc is the **for** loop. If conditions in the **for** statement is met, then the statement in the code block will be executed repeatedly. The **for** loop statement can be used to implement the functionality of while statement and for statement of the normal program language.

If using the **for** statement individually, then it equals to for true. The code in the code block will be cycled non-conditionally. If using the **break** statements, then you can break out of the **for** loop, and stop the code block.



The **for** *n* statement can be used to specify the loops.



The for *A* statement enables you to cycle every member in the sequence *A*:

	A	B	C	D
1	=demo.query ("select * from EMPLOYEE")			
2	for A1	if A2.GENDER=="F"	>A4=A4+1	
3			if B4<A2.SALARY	>B4=A2.SALARY
4				

Assign the value in **C2** and compute the total female employees

Assign the value in **D3**, and compute the highest salary of female employees