esCalc: An Interactive Analysis Tool That Surpasses Excel

Background

Excel

In effect, Excel, instead of many of the BI tools, is the most widely used desktop data analysis tool.

Excel is simple, intuitive, easy to use and to understand, particularly suitable for the average analysts who are incapable of programming and don't have knowledge about mathematical models. Moreover, the result of each action operated on an Excel worksheet will be immediately showed to enable programmers to decide how to make the next move. This represents the typical model the analysts use to perform analytics. It is neither necessary nor possible to model the object beforehand.

But we've found some Excel defects as data analyses become increasingly complicated. Simply and briefly put, they are the "3M" problems:

1. Multi-row records
   There's no definite concept of the structured record in Excel. The single-row record is the record that corresponds to merely one row. If a record has too many data items and thus needs to occupy multiple rows, or if the record has sub-records – that is the multi-row record. It's very complicated to edit a multi-row record and to perform operations on it.
2. Multi-level tables
   Excel provides the function for data grouping, yet, unlike the ungrouped worksheet, many operations become impossible or need to be performed in a different way for the resulting hierarchical table, where consecutive operations are hard to be carried out. What's worse, the grouping generates multi-row records, on which the cross-group copying of formulas that reference the aggregate values can't be performed correctly and intelligently, and, as a result, computing errors arise.
3. Multi-table joins
   Excel isn't a relational-algebra-based product. It doesn't have specialized functions to join tables; it only provides functions, such as Lookup, which perform simple cross-page cell reference, and which are complicated to use and perform poorly.

We'll explain more about these problems later through examples.

esCalc

To solve those Excel problems, we created another model for handling the spreadsheet data, and from this model the brand-new spreadsheet software – esCalc – was born.

Instead of the improved version of Excel, esCalc bases its data and operational models directly on the relational algebra, making it have more in common with the relational database. esCalc defines the record definitely, and provides most of SQL's computational features like computed columns, sorting, filtering, grouping and performing distinct, as well as join and union between multiple tables. And because of its spreadsheet interactive interface, esCalc can be regarded as a visualized SQL calculator.

Different from the general database client software that uses field name in referencing a data item, esCalc inherits Excel's grid style, in which cells are used to name data items and describe formulas, as well as supports automatic and intelligent copying of the formulas. The more intuitive way makes it both easy to be manipulated by the layman and handy to express the order-related computations which SQL isn't good at.

Furthermore, esCalc includes the multi-level data model to increase the related computing capabilities on the basis of SQL, enabling it to support the hierarchical table containing the main table and its sub-tables, to perform operations such as filtering, sorting, re-grouping and ungrouping on the grouped worksheet, and to copy formulas automatically and intelligently between cells at different levels.

esCalc is designed for performing interactive data analysis. It doesn't support such extensive functionalities as Excel does, but it's more sophisticated and more adept with handling batched data. esCalc is intended to be a cooperator of Excel, rather than a rival, as this is unnecessary. Not only can esCalc retrieve and analyze and process an xls file, also it can export the computational result in xls format for further processing in Excel. But it's necessary to point out that esCalc isn't a plug-in for Excel, it is an independent application.

Compared with Excel, the big functionalities for data handling that esCac hasn't are VBA scripting and pivot table. We believe that VBA is too difficult for average analysts; besides, many VBA scripts are created in order to complement those functionalities that are not so convenient to be carried out in Excel and that are already better provided by esCalc, making the VBA scripting not a must. For analysts who are capable of programming, we provide esProc, another product of the RaqSoft software family, to use on the occasions where the script is needed. The software offers scripting abilities that are much more powerful than VBA. As for the pivot table, Excel has already exceled in its feature and leaves little room for improvement. So the target of esCalc is to generate files of xls format as the data sources for Excel pivot tables.

Now let's turn to discussing the above-mentioned "3M" problems of Excel in data analysis and handling through examples, and provide their esCalc solutions.

Structure

Formula copying

Records are represented by the rows in an Excel worksheet. Users can perform operations such

as filtering, sorting on the rows, and, particularly, add computed columns (its values are computed from other fields) for the rows. It's in this latter case where the formula copying becomes a problem.

To add a computed column involves all records (rows), but as Excel hasn't the concept of explicit record, the formula entered in a certain row needs to be manually copied to other rows. Excel cleverly adopts the drag-and-drop method to do this. The method is very convenient-to-use for handling single-row records (that is, each one corresponds to a single row).

But at times the worksheet data we' are handling are complicated in that one record corresponds more than one row. That's because, for example, the record has much content that needs to take up two rows, or the record includes the lower-level sub-records (the details of an order, for instance). In those cases, the cells to which the formula is copied aren't continuous any more, and the drag-and-drop method becomes powerless. We can imagine how much hassle there will be if all the copying is done manually row by row.

esClac solves the problem by both retaining Excel's intuitive way of naming data items after cells and by introducing the concept of the explicit record. This combines the strongest points of Excel and database client software. The formula entered to a certain cell will be automatically and correctly copied to its homo-cells (cells of the desired field in other records) without the specialized copying actions, even if there are multi-row records and records with sub-records.

Here's an *order* table:

0	1		A	B	C	D	E	F
1-		1	**ID**	**SalesID**	**PType**	**Date**	**Amount**	**158191.31**
	1	2	S0201	2	Books	2013-01-01	1479.53	=round(E2/F1,4)
	1	3	S0202	1	Books	2013-01-01	2449.75	
	1	4	S0203	3	Foods	2013-01-01	15522.0	
	1	5	S0205	2	Foods	2013-01-01	2295.0	
	1	6	S0206	3	Books	2013-01-02	665.88	
	1	7	S0207	3	Foods	2013-01-02	50318.0	
	1	8	S0210	2	Foods	2013-01-02	1240.0	
	1	9	S0211	3	Foods	2013-01-03	67826.0	
	1	10	S0212	3	Books	2013-01-03	10547.0	
	1	11	S0213	1	Books	2013-01-03	5848.15	

F1 calculates the total order amount using the formula ={E2}.sum(). We then enter the formula =round(E2/F1,4) in F2 to calculate the percentage of the amount of the current order in the total amount, that is – dividing the total value in F1 by the amount of the current order. At the same time, we set the display format of F2 as #0.00%, which means representing the value in percentage. After entering the formula in F2, here's what we get:

0	1		A	B	C	D	E	F
1-		1	**ID**	**SalesID**	**PType**	**Date**	**Amount**	**158191.31**
	1	2	S0201	2	Books	2013-01-01	1479.53	0.94%
	1	3	S0202	1	Books	2013-01-01	2449.75	1.55%
	1	4	S0203	3	Foods	2013-01-01	15522.0	9.81%
	1	5	S0205	2	Foods	2013-01-01	2295.0	1.45%
	1	6	S0206	3	Books	2013-01-02	665.88	0.42%
	1	7	S0207	3	Foods	2013-01-02	50318.0	31.81%
	1	8	S0210	2	Foods	2013-01-02	1240.0	0.78%
	1	9	S0211	3	Foods	2013-01-03	67826.0	42.88%
	1	10	S0212	3	Books	2013-01-03	10547.0	6.67%
	1	11	S0213	1	Books	2013-01-03	5848.15	3.70%

Check the homo-cells (F3~F11) of F2 and we'll find that they've all finished the computations. This shows that esCalc can copy the formula and display format in one cell to its homo-cells automatically and correctly.

It can also copy the formula in handling multi-row records as conveniently as in handling the single-row records, for example:

0	1		A	B	C	D
1-		1				
	1	2	**Pineapple**		Unit Price	$2.00
	2	3	Quantity	5	Amount	**=floor(D2*B3,2)**
	1	4	**Butternut squash**		Unit Price	$0.99
	2	5	Quantity	1.85	Amount	
	1	6	**Tomato**		Unit Price	$0.99
	2	7	Quantity	2.4	Amount	
	1	8	**Cucumber**		Unit Price	$0.49
	2	9	Quantity	3	Amount	
	1	10	**Apple**		Unit Price	$0.99
	2	11	Quantity	0.88	Amount	

The worksheet contains the unit prices and quantities of vegetables and fruits purchased. D3 calculates the purchasing amount of the pineapple with =floor(D2*B3,2). Here's the result after the formula is entered:

0	1		A	B	C	D
1-		1				
	1	2	**Pineapple**		Unit Price	$2.00
	2	3	Quantity	5	Amount	**$10.00**
	1	4	**Butternut squash**		Unit Price	$0.99
	2	5	Quantity	1.85	Amount	**$1.83**
	1	6	**Tomato**		Unit Price	$0.99
	2	7	Quantity	2.4	Amount	**$2.37**
	1	8	**Cucumber**		Unit Price	$0.49
	2	9	Quantity	3	Amount	**$1.47**
	1	10	**Apple**		Unit Price	$0.99
	2	11	Quantity	0.88	Amount	**$0.87**

As soon as the formula is entered, it is copied to the cells, corresponding to all products, i.e. D3's homo-cells, to calculate their total purchasing amount.

Data editing

Editing multi-row records is another shortcoming of Excel.

Excel doesn't handle a record as a whole. Inserting, deleting and moving a record are operations performed based on the rows and columns of the worksheet. There's almost no problem about processing single-row records. But the operations on rows become complicated in handling multi-row records and records with sub-records, and inserting and deleting fields based on columns are almost non-executable.

Because Excel isn't good at handling the multi-row records, it generally prevents them from appearing when generating the original data. So there're not many chance for the Excel users to encounter them. In many real-world businesses, however, the multi-level worksheet or multi-level data items truly exist, and they are not uncommon. By the way, group operations will generate multi-level tables, as we'll mention later.

Even if there're the single-row records, Excel will still make mistakes in copying formulas for inter-row calculations (such as the calculation of YOY rate and the accumulated value) when rows are inserted or deleted. There's the same problem in moving records through the copy. Both cases require modifying the results manually or recopying by drag-and-drop. In addition, since Excel doesn't stress the concept of record, it offers no hot keys for record processing, making the modification and recopying not that easy.

But it's easy for esCalc, which defines the record, to perform those operations. It also provides the convenient hot keys to trigger the actions in a shortcut way. The records (including their sub-records) can be deleted and moved as a whole with just one click, after which the inter-row calculation formulas will remain correct. To insert and delete fields based on columns is to change the data structure. esCalc will automatically copy these operations on one cell to all its homo-cells.

Here's the *employee* table:

0	1	2		A	B	C	D	E
1-			1					
	1-		2	R&D		2	40.0	
		1	3	R&D	Rebecca Moore	1974-11-20	40	
		1	4	R&D	Ashley Smith	1975-05-13	40	
	1-		5	Sales		3	39.3	
		1	6	Sales	Rachel Johnson	1970-12-17	44	
		1	7	Sales	Matthew Johnson	1984-07-07	31	
		1	8	Sales	Alexis Smith	1972-08-16	43	

There's the formula =age(C3) in D3. The formula, as well as those in its homo-cells, is used to calculate the age of each employee. Meanwhile C2's formula ={B3}.count() calculates the number of employees in each department, and D2's formula =round({D3}.avg(),1) calculates the average age of the employees in each department. Suppose we want to delete the duplicate department values in the first field of the *employee* table without affecting the other data items. To do this we select B3 and press Ctrl+Backspace to delete A3 and its homo-cells. Here's what we get:

0	1	2		A	B	C	D	E
1-			1					
	1-		2	R&D		2	40.0	
		1	3	Rebecca Moore	1974-11-20	40		
		1	4	Ashley Smith	1975-05-13	40		
	1-		5	Sales		3	39.3	
		1	6	Rachel Johnson	1970-12-17	44		
		1	7	Matthew Johnson	1984-07-07	31		
		1	8	Alexis Smith	1972-08-16	43		

All the homo-rows will change their structure at the same time. In the meantime formulas in C3 and its homo-cells will adapt themselves intelligently to the new structure. For instance, C3's formula becomes =age(B3) automatically.

In esCalc we can merely change the structure of the summary rows. For instance, to delete the blank cells in the second column of the department summary rows, we select C2 and press Ctrl+Backspace to delete A2 and its homo-cells. Here's what we get:

0	1	2		A	B	C	D	E
1-			1					
	1-		2	R&D	2	40.0		
		1	3	Rebecca Moore	1974-11-20	40		
		1	4	Ashley Smith	1975-05-13	40		
	1-		5	Sales	3	39.3		
		1	6	Rachel Johnson	1970-12-17	44		
		1	7	Matthew Johnson	1984-07-07	31		
		1	8	Alexis Smith	1972-08-16	43		

This is a *membership management* table:

0	1		A	B	C	D
1-		1	**Month**	**Join in**	**Leave**	**Remain**
	1	2	1	500		500
	1	3	2	96	35	561
	1	4	3	1	15	547
	1	5	6	7	31	523
	1	6	7	60	34	549
	1	7	8	65	48	566
	1	8	9	15	39	542

The worksheet table records the number of new members and those who leave each month. Enter ==D7+B8-C8 in D8 to calculate the number of members in the current month according to the number in the last month and the number of withdrawals in this month. Here the related calculation expression starting with two equal signs is used and the data in the corresponding cells will adjust intelligently according to any change of the table.

Now we insert the records of the missing months April and May in the table and enter data to them. Here's the complete table:

0	1		A	B	C	D
1-		1	**Month**	**Join in**	**Leave**	**Remain**
	1	2	1	500		500
	1	3	2	96	35	561
	1	4	3	1	15	547
	1	5	4	6	13	540
	1	6	5	15	21	534
	1	7	6	7	31	510
	1	8	7	60	34	536
	1	9	8	65	48	553
	1	10	9	15	39	529

Because esCalc stores the inserted data also in the form of homo-rows, the calculations in column D will still be correctly done and the membership statistics will be automatically updated along with the change of the data. If the formulas are changed according to positions of the cells instead of their structure, errors will occur when new rows are inserted.

Non-related calculations

Cells in Excel calculate in a related way. That means once the value of a referenced cell changes, the calculation cell will re-calculate; and if the referenced cell is deleted, error will occur to the calculation cell.

But the more commonly seen scenarios are these: After the values of a computed column are obtained, the values of cells referenced by the formulas become useless and deletable; or we may change the original value to be referenced by a computed column and then compare the

new value and the old value, in which case the old value is expected to remain what it was. For instance, the original data contains persons' birthdays. Sometimes only birthdays during a certain time period are needed to compute the ages in the subsequent computations. Thus the birthday values can be deleted after the ages are obtained; other times the birthday values are changed and ages are calculated according to the changed birthdays. It's not easy to deal with both scenarios in Excel.

esCalc offers two types of calculation cells: related calculation cell and non-related calculation cell. The value of a related calculation cell will change along with the change of the referenced cell, as with in the Excel; a non-related calculation cell becomes irrelevant to the referenced cell once it finishes the calculation, and either the change or the deletion of the referenced cell can't affect its value anymore.

In reality, there are more non-related calculations than related calculations during interactive data analysis.

This is the *population* table of the state of Alaska:

0	1		A	B	C	D
1-		1	**Year**	**Population of Alaska**		
	1	2	1950	128643		
	1	3	1960	226187	75.8%	
	1	4	1970	300382	32.8%	
	1	5	1980	401851	33.8%	
	1	6	1990	550043	36.9%	
	1	7	2000	626932	14.0%	
	1	8	2010	710231	13.3%	

C8 and its homo-cells calculate the growth rate of every census for the state of Alaska. The formula in C8 is =round((B8-B7)/B7,3) and the display format is #0.0%. Now we select C2 and sort the records by the growth rate in descending order. Here's the result:

0	1		A	B	C	D
1-		1	**Year**	**Population of Alaska**		
	1	2	1960	226187	75.8%	
	1	3	1990	550043	36.9%	
	1	4	1980	401851	33.8%	
	1	5	1970	300382	32.8%	
	1	6	2000	626932	14.0%	
	1	7	2010	710231	13.3%	
	1	8	1950	128643		

Since the formulas in column C are headed by a single equal sign, C2 and its homo-cells are non-related calculation cells which will keep their values unchanged, rather than re-calculate to get the wrong growth rates according to the new order.

Grouping

Formulas and their copying

We mentioned in the preceding part that there are records with their sub-records. But in many cases the hierarchical records are generated from group operations.

The Excel data model doesn't support the multi-level worksheet. Though the group operation is provided, it is treated specially. The aggregate operation on the summary level after data grouping uses SUBTOTAL, which is difficult to memorize, instead of the more familiar functions like SUM/COUNT; otherwise the group members won't be correctly located.

As mentioned previously, for the formulas in the cells at the level of detail data, on one hand we can't simply use the drag-and-drop method to perform the batched copying (because the detail data is inconsecutive data areas separated by summary rows) but we can only perform the cross-group copying manually; on the other hand, when formulas reference cells at the summary level or involve cross-row calculations (such as the calculation of percentages and YOY rate), even the manual operation can't guarantee a correct copying according to the Excel rule of formula copying, and, moreover, manual modification of the mistakenly copied formulas is needed. All these work is too tedious to bear when there are a lot of groups.

Excel provides the sign $ to reference the summary cell in a one-level worksheet, but it is helpless when facing the multi-level worksheet.

esCalc has a data model that supports the multi-level table. The aggregate operations performed on the summary level after data grouping still use the common functions sum/count. Particularly, esCalc distinguishes the levels to which the cells belong, and handles the copying of formulas that reference cells both at detail data level (including inter-row reference) and at summary data level according to different situations. The intra-group copying only adjusts cells at the detail data level, while cross-group copying changes the cells at summary data level. What the esCalc users need to do is to reference the desirable cell intuitively, without having to distinguish different levels themselves using the sign $ (actually the sign can merely reference data from one level and falls short of the need). With esCalc, formulas can be correctly copied even the calculations involve multiple levels of summary data.

To calculate the average temperature difference in each month, for example, based on the following sheet:

0	1	2		A	B	C	D	E
1-			1	**Climate data**				
	1-		2	**Quarter 1**	**High ℉**	**Low ℉**	**Precipitation inches**	
		1	3	Jan	30	15	1.91	**=B3-C3**
		1	4	Feb	35	18	1.93	
		1	5	Mar	47	27	2.46	

	1-		6	**Quarter 2**	**High ℉**	**Low ℉**	**Precipitation inches**	
		1	7	Apr	60	37	3.55	
		1	8	May	71	46	4.07	

esCalc stores the same type of data in homo-rows. Thus as E3 calculates the average temperature difference in January, its homo-cells corresponding to other months calculate their respective average temperature differences at the same time, saving users the trouble of copying formulas. Here's the result:

0	1	2		A	B	C	D	E
1-			1	**Climate data**				
	1-		2	**Quarter 1**	**High ℉**	**Low ℉**	**Precipitation inches**	
		1	3	Jan	30	15	1.91	**15**
		1	4	Feb	35	18	1.93	**17**
		1	5	Mar	47	27	2.46	**20**
	1-		6	**Quarter 2**	**High ℉**	**Low ℉**	**Precipitation inches**	
		1	7	Apr	60	37	3.55	**23**
		1	8	May	71	46	4.07	**25**

In the above data handling, the month data is sub-rows and their parent row is the quarter data. Calculations performed on the sub-rows won't affect the parent row; and similarly, data handling in the parent row won't affect the sub-rows. For instance, enter the formula ={A3}.count() in E2 to calculate the number of records in each quarter. Here's the result:

0	1	2		A	B	C	D	E
1-			1	**Climate data**				
	1-		2	**Quarter 1**	**High ℉**	**Low ℉**	**Precipitation inches**	**3**
		1	3	Jan	30	15	1.91	**15**
		1	4	Feb	35	18	1.93	**17**
		1	5	Mar	47	27	2.46	**20**
	1-		6	**Quarter 2**	**High ℉**	**Low ℉**	**Precipitation inches**	**2**
		1	7	Apr	60	37	3.55	**23**
		1	8	May	71	46	4.07	**25**

The esCalc formulas can be intelligently copied according to different data structures, instead of being mechanically copied according to the positions of the cells. The esCalc copying rule is more reasonable.

Another example is to calculate the precipitation based on the *climate* table:

0	1	2		A	B	C	D	E
1-			1	**Climate data**				
	1-		2	**Quarter 1**	**High ℉**	**Low ℉**	**Precipitation inches**	**2.1**
		1	3	Jan	30	15	1.91	
		1	4	Feb	35	18	1.93	
		1	5	Mar	47	27	2.46	
	1-		6	**Quarter 2**	**High ℉**	**Low ℉**	**Precipitation inches**	**3.81**
		1	7	Apr	60	37	3.55	
		1	8	May	71	46	4.07	

Of which E2 and E6 respectively calculate the average precipitation of the current quarter, as E2's formula ={D3}.avg(). To calculate the difference between the precipitation in each month and the average precipitation in the corresponding quarter, just enter the formula =D3-E2 in E3. Here's the result:

0	1	2		A	B	C	D	E
1-			1	**Climate data**				
	1-		2	**Quarter 1**	**High ℉**	**Low ℉**	**Precipitation inches**	**2.1**
		1	3	Jan	30	15	1.91	**-0.19**
		1	4	Feb	35	18	1.93	**-0.17**
		1	5	Mar	47	27	2.46	**0.36**
	1-		6	**Quarter 2**	**High ℉**	**Low ℉**	**Precipitation inches**	**3.81**
		1	7	Apr	60	37	3.55	**-0.26**
		1	8	May	71	46	4.07	**0.26**

Formulas have been intelligently adjusted during the copying according to the hierarchical level to which the target cell belongs. For instance, we click on E6 and know that the formula has been adjusted as ={D7}.avg(), which calculates the average precipitation of the current quarter; click on E8 and see the formula have been adapted as =D8-E6, which means subtracting the average precipitation value of the current quarter from the precipitation of the current month. So we can see that esCalc can correctly copy the formula to both a cell that sits on a group's summary row and one that sits on a group's detail row.

Post-grouping operations

That data grouping in Excel is special is also reflected by the difficulty in handling the post-grouping operations. We can't perform operations such as sorting and filtering freely on the grouped worksheet table as what we do with a single-level table.

For example, in order to find out the ranks of sellers on performances, we want to group and aggregate the order records by sellers and then sort the groups according to the aggregate amounts. To do that we need to first perform group and aggregate and then the sort by aggregate values; during the sorting, members of the group need to move together with the aggregate value. But we can't perform this kind of sorting automatically in Excel. In a modified version of this example, for each seller we want to delete the small orders, each of which makes up less than 1% of the seller's total sales amount, and then re-calculate the total amount. This

requires grouping data and calculating the percentage of each member in each group, and performing filtering on all groups by the percentages (here the non-related calculations discussed above will also be used). Excel can't make it all at once due to its lack of support to operations on the multi-level worksheet; it can only handle the groups separately one by one.

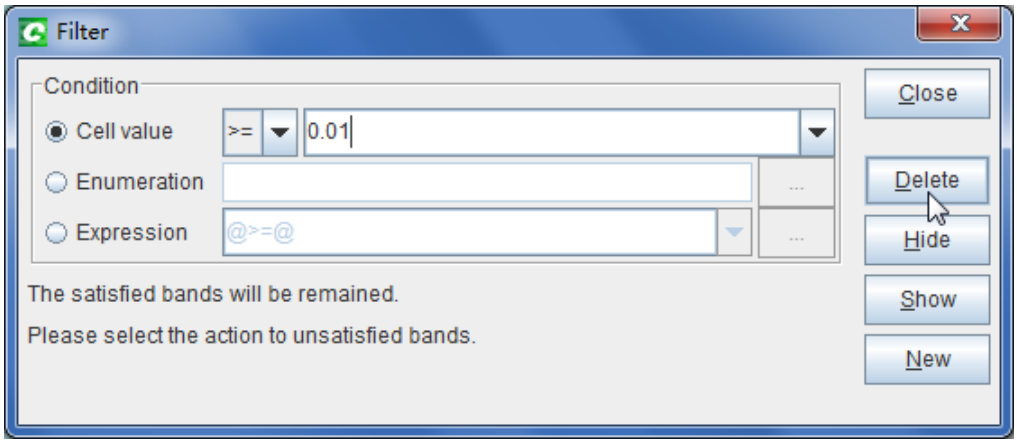
esCalc sees the multi-level worksheet as normal, and makes it open to all operations. So it's easy for esCalc to handle the above scenarios. In esCalc, during the sorting by aggregate values after data grouping, the detail rows of a group will move together with their summary row, which is again the application of esCalc record conception (a group with its members as a whole can be regarded as a record). The post-grouping filtering on detail rows will be performed once and for all by copying all groups at one time.

Here's the *order* table:

0	1	2		A	B	C	D	E	F
1-			1	**ID**	**SalesID**	**PType**	**Date**	**Amount**	**158191.31**
	1-		2		**1**				
		1	3	S0202	1	Books	2013-01-01	2449.75	0.0155
		1	4	S0213	1	Books	2013-01-03	5848.15	0.037
	1-		5		**2**				
		1	6	S0201	2	Books	2013-01-01	1479.53	0.0094
		1	7	S0205	2	Foods	2013-01-01	2295.0	0.0145
		1	8	S0210	2	Foods	2013-01-02	1240.0	0.0078
	1-		9		**3**				
		1	10	S0203	3	Foods	2013-01-01	15522.0	0.0981
		1	11	S0206	3	Books	2013-01-02	665.88	0.0042
		1	12	S0207	3	Foods	2013-01-02	50318.0	0.3181
		1	13	S0211	3	Foods	2013-01-03	67826.0	0.4288
		1	14	S0212	3	Books	2013-01-03	10547.0	0.0667

F1 calculates the total sales amount of the orders with the formula ={E3}.sum(). F3 calculates the percentage of each order's amount in the total amount with the formula =round(E3/F1,4).

esCalc permits various operations on a grouped worksheet, such as filtering. To delete every order whose amount makes up less than 1% of the total sales amount, we select F3 to do the filtering:



Here's what we get through data filtering:

0	1	2		A	B	C	D	E	F
1-			1	**ID**	**SalesID**	**PType**	**Date**	**Amount**	**158191.31**
	1-		2		**1**				
		1	3	S0202	1	Books	2013-01-01	2449.75	0.0155
		1	4	S0213	1	Books	2013-01-03	5848.15	0.037
	1-		5		**2**				
		1	6	S0205	2	Foods	2013-01-01	2295.0	0.0145
	1-		7		**3**				
		1	8	S0203	3	Foods	2013-01-01	15522.0	0.0981
		1	9	S0207	3	Foods	2013-01-02	50318.0	0.3181
		1	10	S0211	3	Foods	2013-01-03	67826.0	0.4288
		1	11	S0212	3	Books	2013-01-03	10547.0	0.0667

Structure editing

Excel doesn't support inserting or deleting the data level based on a grouped worksheet. To change the existing data structure, we need to clear the groups and re-group the data, making the work we did on the summary level (the calculated cells) a waste. Sometimes it is the summary values, instead of the details, that we desired.

In the worksheet in which the small orders have been removed, for example, we need to group the records by the ordered products to see which products are more popular among each seller's non-small orders. To do this we need to insert another level of groups into the double-level grouped worksheet, and to aggregate and sort each group. As we are only interested in the group and aggregate results, we want to delete the detail level of data. But we can't perform these operations automatically in Excel. We have to copy the intermediate results out into a new worksheet for further handling. Even worse is that since the grouped data is not continuous, even the copying action can't be carried out automatically.

In esCalc, we re-group the preceding worksheet table by products, and here's the result:

0	1	2	3		A	B	C	D	E	F
1-				1	**ID**	**SalesID**	**PType**	**Date**	**Amount**	**158191.31**
	1-			2		**1**				
		1-		3			Books			
			1	4	S0202	1	Books	2013-01-01	2449.75	0.0155
			1	5	S0213	1	Books	2013-01-03	5848.15	0.037
	1-			6		**2**				
		1-		7			Foods			
			1	8	S0205	2	Foods	2013-01-01	2295.0	0.0145
	1-			9		**3**				
		1-		10			Books			
			1	11	S0212	3	Books	2013-01-03	10547.0	0.0667
		1-		12			Foods			
			1	13	S0203	3	Foods	2013-01-01	15522.0	0.0981
			1	14	S0207	3	Foods	2013-01-02	50318.0	0.3181
			1	15	S0211	3	Foods	2013-01-03	67826.0	0.4288

We can do further computations based on the re-grouped worksheet. To calculate the total sales amount for each seller, for instance, we enter the same formula ={E4}.sum() in both E2 and E3. Here's the result:

0	1	2	3		A	B	C	D	E	F
1-				1	**ID**	**SalesID**	**PType**	**Date**	**Amount**	**158191.31**
	1-			2		**1**			**8297.9**	
		1-		3			Books		**8297.9**	
			1	4	S0202	1	Books	2013-01-01	2449.75	0.0155
			1	5	S0213	1	Books	2013-01-03	5848.15	0.037
	1-			6		**2**			**2295.0**	
		1-		7			Foods		**2295.0**	
			1	8	S0205	2	Foods	2013-01-01	2295.0	0.0145
	1-			9		**3**			**144213.0**	
		1-		10			Books		**10547.0**	
			1	11	S0212	3	Books	2013-01-03	10547.0	0.0667
		1-		12			Foods		**133666.0**	
			1	13	S0203	3	Foods	2013-01-01	15522.0	0.0981
			1	14	S0207	3	Foods	2013-01-02	50318.0	0.3181
			1	15	S0211	3	Foods	2013-01-03	67826.0	0.4288

We entered the same formulas in F1, E2 and E3 to calculate the total sales amount, but we get different results because they are entered in cells that sit at different levels.

Now we select E2 to perform a sorting in descending order to sort the worksheet data by the total sales amounts of the sellers. The result is as follows:

0	1	2	3		A	B	C	D	E	F
1-				1	**ID**	**SalesID**	**PType**	**Date**	**Amount**	**158191.31**
	1-			2		**3**			**144213.0**	
		1-		3			Books		**10547.0**	
			1	4	S0212	3	Books	2013-01-03	10547.0	0.0667
		1-		5			Foods		**133666.0**	
			1	6	S0203	3	Foods	2013-01-01	15522.0	0.0981
			1	7	S0207	3	Foods	2013-01-02	50318.0	0.3181
			1	8	S0211	3	Foods	2013-01-03	67826.0	0.4288
	1-			9		**1**			**8297.9**	
		1-		10			Books		**8297.9**	
			1	11	S0202	1	Books	2013-01-01	2449.75	0.0155
			1	12	S0213	1	Books	2013-01-03	5848.15	0.037
	1-			13		**2**			**2295.0**	
		1-		14			Foods		**2295.0**	
			1	15	S0205	2	Foods	2013-01-01	2295.0	0.0145

In esCalc, when the grouping rows move because of sorting or other operations, their sub-rows will follow suit.

There's nothing particular for esCalc to carry out these operations. Because it defines the hierarchical level as a nature of the worksheet, enabling free insertion or deletion of a level and automatic copying of an action operated on a row to all its homo-rows (similar concept to homo-cells). So all detail rows will be deleted simultaneously if we execute the action on a certain detail row.

The data model for esCalc spreadsheet encompasses a hierarchical structure, making grouping and ungrouping the normal operations that can be still performed on the same worksheet as filtering and sorting. Here's an analogy between the spreadsheet data models and the numerical system. Within the range of integers, we are free to do addition, subtraction and multiplication but we can't do division at will, because the quotient isn't necessarily an integer. But if we expand the range into the rational numbers, the division operation becomes naturally as well, though we need to redefine the rules for the other operations in the expanded scope. Likewise, when esCalc extends the data model for the worksheets to include a multi-level structure, it also redefines rules for carrying out sorting, filtering and generating computed columns (to support the smart copying of formulas across different levels, for instance). By doing so, related operations can be performed consecutively, ensuring the interactive data analysis to proceed smoothly.

Join

The JOIN is one of the most important SQL operations. It is used in scenarios such as obtaining

attributes through codes (like using the product codes to get the producing areas and the unit prices) and multiple table alignments (like aligning both the *allowance* table and *attendance* table with the *employee* table)。

Excel uses Lookup functions to associate tables. They are similar to the SQL left join. SQL also has inner join, right join and full join, among which the inner join is implemented through filtering after the left join and the right join is the opposite operation of the left join with joining direction changed. The full join, however, can't be performed automatically in Excel.

The biggest problem of the Lookup functions is their complicated usage. They need to specify the joining column, the joining scope and the referenced columns, with only one referenced column for each look-up, and multiple Lookup statements using the same query condition for referencing multiple columns. Not only is the writing troublesome, but also the method has a poor performance due to the repeated operations. In fact as a traversal-style query method, the Lookup function is very inefficient in searching associated data.

Based on SQL model, esCalc supports the whole set of join operations including inner join, left join and full join, with multiple columns referenced at once from the associated table by specifying the associated cells in the two worksheet to be joined. This is much simpler than using the Excel method. To join the *performance* table and the *attendance* table, for instance, set the master cells (i.e. the joining cells) and copy the to-be-referenced cells in the *attendance* table and paste them on the *employee performance* table using the JOIN operation.

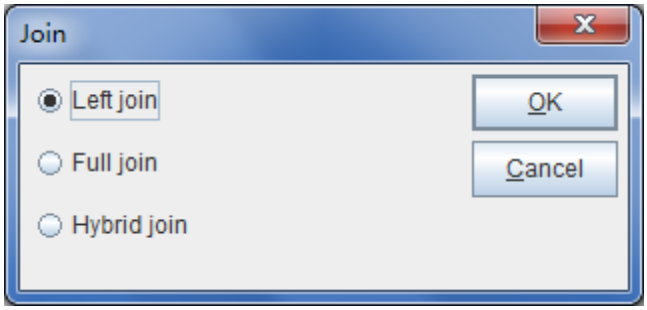
Here's the *performance* table, in which A2 and its homo-cells are set as the master cells where the employee numbers are stored:

0	1		A	B	C	D	E
1-		1	**EID**	**Name**	**Base Wage**	**Bonus**	
	1	2	1	Rebecca Moore	2000	0	
	1	3	2	Ashley Wilson	2200	0	
	1	4	3	Rachel Johnson	1800	1100	
	1	5	4	Emily Smith	1200	0	
	1	6	5	Ashley Scott	2000	200	
	1	7	6	Matthew Jones	1600	500	
	1	8	7	Alexis Smith	1300	0	
	1	9	8	Megan Wilson	3000	500	
	1	10	9	Victoria Green	2300	900	
	1	11	10	Ryan Jackson	2600	0	
	1	12	11	Jacob Moore	1250	0	
	1	13	12	Jessica Davis	2000	300	

Here's the *attendance* table, which contains only the employees who have had absences, and in which A2 and its homo-cells are the master cells holding the employee numbers:

0	1		A	B
1-		1	**EID**	**Absence**
	1	2	2	6
	1	3	3	24
	1	4	5	4
	1	5	7	10
	1	6	11	16
	1	7	12	8

To perform the join operation, select B2 in the *attendance* table and press Ctrl+C to copy, and then select E2 in the *employee performance* table and press Ctrl+Alt+J to choose and execute the Left join:



After that the resulting *employee performance* table is as follows:

0	1		A	B	C	D	E
1-		1	**EID**	**Name**	**Base Wage**	**Bonus**	
	1	2	1	Rebecca Moore	2000	0	
	1	3	2	Ashley Wilson	2200	0	6
	1	4	3	Rachel Johnson	1800	1100	24
	1	5	4	Emily Smith	1200	0	
	1	6	5	Ashley Scott	2000	200	4
	1	7	6	Matthew Jones	1600	500	
	1	8	7	Alexis Smith	1300	0	10
	1	9	8	Megan Wilson	3000	500	
	1	10	9	Victoria Green	2300	900	
	1	11	10	Ryan Jackson	2600	0	
	1	12	11	Jacob Moore	1250	0	16
	1	13	12	Jessica Davis	2000	300	8

The esCalc join operation also supports multi-level worksheet tables. For example, the employees are stored in groups according to their states, and the attendances are recorded in the same way. The multi-level join will first align tables according to the groups and then find the joining rows in each group. This way error won't occur even there are employees with the same names under different states and the result set will be obtained with the detail group data kept neatly and completely.

Here's the *duty* table, in which the master cells hold the state names and employee numbers:

0	1	2		A	B	C	D
1-			1	**Day**	**State/ID**		
	1-		2		**California**		
		1	3	Saturday	1		
		1	4	Sunday	2		
	1-		5		**New York**		
		1	6	Monday	1		
		1	7	Wednesday	2		
		1	8	Friday	1		
	1-		9		**Texas**		
		1	10	Monday	1		
		1	11	Tuesday	2		
		1	12	Wednesday	3		
		1	13	Thursday	1		
		1	14	Friday	4		

Here's the *employee* table, in which the master cells also hold the state names and employee numbers:

0	1	2		A	B	C	D
1-			1	**EID**	**Name**	**DEPT**	**STATE**
	1-		2				California
		1	3	1	Rebecca Moore	R&D	California
		1	4	2	Matthew Johnson	Sales	California
		1	5	3	Megan Wilson	Marketing	California
	1-		6				New York
		1	7	1	Ashley Wilson	Finance	New York
		1	8	2	Jessica Davis	Sales	New York
	1-		9				Texas
		1	10	1	Emily Smith	HR	Texas
		1	11	2	Ashley Smith	R&D	Texas
		1	12	3	Victoria Davis	HR	Texas
		1	13	4	Jacob Moore	Sales	Texas

In this *employee* table, select B3 and C3 at the same time and copy the employee information, and then select C3 in the *duty* table and perform left join. Here's the result:

0	1	2		A	B	C	D
1-			1	**Day**	**State/ID**		
	1-		2		**California**		
		1	3	Saturday	1	Rebecca Moore	R&D
		1	4	Sunday	2	Matthew Johnson	Sales
	1-		5		**New York**		
		1	6	Monday	1	Ashley Wilson	Finance
		1	7	Wednesday	2	Jessica Davis	Sales
		1	8	Friday	1	Ashley Wilson	Finance
	1-		9		**Texas**		
		1	10	Monday	1	Emily Smith	HR
		1	11	Tuesday	2	Ashley Smith	R&D
		1	12	Wednesday	3	Victoria Davis	HR
		1	13	Thursday	1	Emily Smith	HR
		1	14	Friday	4	Jacob Moore	Sales