esProc

Some examples of solving BIRT special layout

Issued by Raqsoft

Contents

Horizontal columns

Some special layout is difficult to realize directly through the functions provided by the reporting tool itself, but if we prepare a suitable data source, we can greatly reduce the difficulty of report design!

Birt and other reporting tools only support vertical column division, so it is difficult to realize the layout of horizontal and column division of records, as follows:

Report format

Eld	Name	Dept
4	Emily	HR
7	Alexis	Sales
10	Ryan	R&D
13	Daniel	Finance
16	Christopher	Production
19	Samantha	Production

Eld	Name	Dept
5	Ashley	R&D
8	Megan	Marketing
11	Jacob	Sales
14	Alyssa	Sales
17	Hannah	Marketing
20	Alexis	Administration

Eld	Name	Dept
6	Matthew	Sales
9	Joseph	Production
12	Jessica	Sales
15	Alexis	Sales
18	Jonathan	Administration

Source data table

EID	NAME	DEPT
4	Emily	HR
5	Ashley	R&D
6	Matthew	Sales
7	Alexis	Sales
8	Megan	Marketing
9	Victoria	HR
10	Ryan	R&D
11	Jacob	Sales
12	Jessica	Sales
13	Daniel	Finance
14	Alyssa	Sales
15	Alexis	Sales
16	Christopher	Production
17	Hannah	Marketing
18	Jonathan	Administration
19	Samantha	Production
20	Alexis	Administration

Horizontal columns – Example

The original 3-field data can be converted into 9-field data with esProc, and the horizontal columns can be divided with the reporting tool. The code example is as follows:

Index	EID	NAME	DEPT
1	4	Emily	HR
2	7	Alexis	Sales
3	10	Ryan	R&D
4	13	Daniel	Finance
5	16	Christopher	Production
6	19	Samantha	Production

Index	EID	NAME	DEPT
1	5	Ashley	R&D
2	8	Megan	Marketing
3	11	Jacob	Sales
4	14	Alyssa	Sales
5	17	Hannah	Marketing
6	20	Alexis	Administration

Index	EID	NAME	DEPT
1	6	Matthew	Sales
2	9	Victoria	HR
3	12	Jessica	Sales
4	15	Alexis	Sales
5	18	Jonathan	Administration

A2: Take the first record from A1 every 3 records to form a new two-dimensional table. B2, C2 are similar.

Sample code

	A	B	C
1	=myDB.query("select EId,Name,Dept from emp where EId>=? and EId<=? order by EId ",begin,end)		
2	=A1.step(3,1)	=A1.step(3,2)|[null]	=A1.step(3,3)|[null]
3	=A2.derive(B2(#).EID:EID2,B2(#).NAME:NAME2,B2(#).DEPT:DEPT2,C2(#).EID:EID3,C2(#).NAME:NAME3,C2(#).DEPT:DEPT3)		
4	return A3		

Divide the data into three parts according to the index number, and stored in A2, B2 and C2. Then add the fields in B2 and C2 to A2 in turn, as shown in the right figure:

Index	EID	NAME	DEPT	EID2	NAME2	DEPT2	EID3	NAME3	DEPT3
1	4	Emily	HR	5	Ashley	R&D	6	Matthew	Sales
2	7	Alexis	Sales	8	Megan	Marketing	9	Victoria	HR
3	10	Ryan	R&D	11	Jacob	Sales	12	Jessica	Sales
4	13	Daniel	Finance	14	Alyssa	Sales	15	Alexis	Sales
5	16	Christopher	Production	17	Hannah	Marketing	18	Jonathan	Administr...
6	19	Samantha	Production	20	Alexis	Administr...	(null)	(null)	(null)

Divided into columns by mixed rows

The source data table is horizontally divided into two columns, the second column of each row is the same as the first column of the next row, and the report layout is as follows:

Report format

EID	NAME	DEPT	EID2	NAME2	DEPT2
4	Emily	HR	5	Ashley	R&D
5	Ashley	R&D	6	Matthew	Sales
6	Matthew	Sales	7	Alexis	Sales
7	Alexis	Sales	8	Megan	Marketing
8	Megan	Marketing	9	Victoria	HR
9	Victoria	HR	10	Ryan	R&D
10	Ryan	R&D	11	Jacob	Sales
11	Jacob	Sales	12	Jessica	Sales
12	Jessica	Sales	13	Daniel	Finance
13	Daniel	Finance	14	Alyssa	Sales
14	Alyssa	Sales	15	Alexis	Sales
15	Alexis	Sales	16	Christopher	Production
16	Christopher	Production	17	Hannah	Marketing
17	Hannah	Marketing	18	Jonathan	Administration
18	Jonathan	Administration	19	Samantha	Production
19	Samantha	Production	20	Alexis	Administration

Source data table

EID	NAME	DEPT
4	Emily	HR
5	Ashley	R&D
6	Matthew	Sales
7	Alexis	Sales
8	Megan	Marketing
9	Victoria	HR
10	Ryan	R&D
11	Jacob	Sales
12	Jessica	Sales
13	Daniel	Finance
14	Alyssa	Sales
15	Alexis	Sales
16	Christopher	Production
17	Hannah	Marketing
18	Jonathan	Administration
19	Samantha	Production
20	Alexis	Administration

Divided into columns by mixed rows – Example

A2: staggered splicing: splice the previous record and the current record in turn, and take the splicing result from the third row, as shown in the right figure:

Index	EID	NAME	DEPT
1	4	Emily	HR
2	5	Ashley	R&D
3	5	Ashley	R&D
4	6	Matthew	Sales
5	6	Matthew	Sales
6	7	Alexis	Sales
7	7	Alexis	Sales
8	8	Megan	Marketing
9	8	Megan	Marketing
10	9	Victoria	HR
11	9	Victoria	HR
12	10	Ryan	R&D
13	10	Ryan	R&D
14	11	Jacob	Sales
15	11	Jacob	Sales

	A	B
1	=myDB.query("select EId,Name,Dept from emp where EId>=? and EId<=? order by EId ",begin,end)	
2	=A1.conj([~[-1],~]).to(3,)	
3	=A2.step(2,1)	=A2.step(2,2)|[null]
4	=A3.derive(B3(#).EID:EID2,B3(#).NAME:NAME2,B3(#).DEPT:DEPT2)	

EID	NAME	DEPT
4	Emily	HR
5	Ashley	R&D
6	Matthew	Sales
7	Alexis	Sales
8	Megan	Marketing
9	Victoria	HR
10	Ryan	R&D
11	Jacob	Sales
12	Jessica	Sales
13	Daniel	Finance

EID	NAME	DEPT
5	Ashley	R&D
6	Matthew	Sales
7	Alexis	Sales
8	Megan	Marketing
9	Victoria	HR
10	Ryan	R&D
11	Jacob	Sales
12	Jessica	Sales
13	Daniel	Finance
14	Alyssa	Sales

Wide table horizontal printing

Given a wide table of any database, and there are too many columns to be printed on a piece of paper. The report requires each paper to print column head and column number. Columns 1 to n are printed on the first paper, and columns n + 1 to 2n are printed on the second paper, and so on. The diagram is as follows:

Report format

page1

ORDERID	CLIENT	AMOUNT	ORDERDATE
10248	VINET	428.0	2012-07-04
10249	TOMSP	1842.0	2012-07-05
10250	HANAR	1523.49998...	2012-07-08
10251	VICTE	624.949999...	2012-07-08
10252	SUPRD	3559.49999...	2012-07-09
10253	HANAR	1428.0	2012-07-10
10254	CHOPS	545.399997...	2012-07-11
10255	RICSU	2450.0	2012-07-12
10256	WELLI	510.0	2012-07-15
10257	HILAA	1109.0	2012-07-16

page2

SELLERID	POSTALCODE	DELIVERYDATE	SALAY
3	111080	2012-07-16	32.38
1	440876	2012-07-10	11.61
2	754546	2012-07-12	65.83
1	690047	2012-07-15	41.34
2	567889	2012-07-11	51.30000000...
2	545486	2012-07-16	58.17
2	301256	2012-07-23	22.98
3	120477	2012-07-15	148.33
2	873763	2012-07-17	13.97
3	502234	2012-07-22	81.91

page3

ORDERID	CLIENT	AMOUNT	ORDERDATE
10258	ERNSH	1603.99999...	2012-07-17
10259	CENTC	100.0	2012-07-18
10260	OTTIK	1461.75	2012-07-19
10261	QUEDE	440.0	2012-07-19
10262	RATTC	583.199999...	2012-07-22
10263	ERNSH	1785.0	2012-07-23
10264	FOLKO	673.749998...	2012-08-20
10265	BLONP	1170.0	2012-07-25
10266	WARTH	341.999999...	2012-07-26
10267	FRANK	3496.49998...	2012-07-29

page4

SELLERID	POSTALCODE	DELIVERYDATE	SALAY
1	801009	2012-07-23	140.51
3	705022	2012-07-19	3.25
1	140739	2012-07-29	55.09
2	238967	2012-07-30	3.050000000...
3	871610	2012-07-25	48.29
3	101043	2012-07-31	146.06
3	144567	2012-08-23	3.67
1	670005	2012-08-12	55.28
3	101103	2012-07-31	25.73
1	380805	2012-08-06	208.58

Source data table

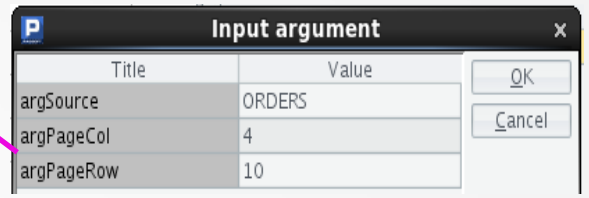
ORDERID	CLIENT	AMOUNT	ORDERDA...	SELLERID	POSTALC...	DELIVERYDATE	SALAY
10248	VINET	428.0	2012-07...	3	111080	2012-07-16	32.38
10249	TOMSP	1842.0	2012-07...	1	440876	2012-07-10	11.61
10250	HANAR	1523.499...	2012-07...	2	754546	2012-07-12	65.83
10251	VICTE	624.9499...	2012-07...	1	690047	2012-07-15	41.34
10252	SUPRD	3559.499...	2012-07...	2	567889	2012-07-11	51.30000...
10253	HANAR	1428.0	2012-07...	2	545486	2012-07-16	58.17
10254	CHOPS	545.3999...	2012-07...	2	301256	2012-07-23	22.98
10255	RICSU	2450.0	2012-07...	3	120477	2012-07-15	148.33
10256	WELLI	510.0	2012-07...	2	873763	2012-07-17	13.97
10257	HILAA	1109.0	2012-07...	3	502234	2012-07-22	81.91
10258	ERNSH	1603.999...	2012-07...	1	801009	2012-07-23	140.51
10259	CENTC	100.0	2012-07...	3	705022	2012-07-19	3.25
10260	OTTIK	1461.75	2012-07...	1	140739	2012-07-29	55.09
10261	QUEDE	440.0	2012-07...	2	238967	2012-07-30	3.050000...
10262	RATTC	583.1999...	2012-07...	3	871610	2012-07-25	48.29
10263	ERNSH	1785.0	2012-07...	3	101043	2012-07-31	146.06
10264	FOLKO	673.7499...	2012-08...	3	144567	2012-08-23	3.67
10265	BLONP	1170.0	2012-07...	1	670005	2012-08-12	55.28
10266	WARTH	341.9999...	2012-07...	3	101103	2012-07-31	25.73
10267	FRANK	3496.499...	2012-07...	1	380805	2012-08-06	208.58

Wide table horizontal printing – Example

Input parameter: Database table name

Input parameter: Number of columns per page

Input parameter: Number of rows per page



	A	
1	=myDB.query("select * from "+argSource)	/argSource is the name of database table
2	=create(${argPageCol.(concat("Col",~)).string()})	/Dynamically generate an empty 2D table with argPageCol columns
3	=A1.group((#-1)\argPageRow)	/Divide A1 into groups every argpagerow row
4	=(fn=A1.fno()).step(argPageCol,1).(to(~,if((t=~+argPageCol-1)>fn,fn,t)).(A1.fname(~)))	/Divide the field names in A1 into groups of every argpagecol
5	=A4.("["+~.string@q()+"]|~.conj(["+~.string()+"])").string("|")	/Splice string, which can be executed dynamically in A6
6	=A3.run(A2.record(${A5}))	/Loop each group of data of A3, insert the field name and field value of each page into A2 in turn
7	return A2	/Return A2 to reporting tool

Index	Member
1	[[10248,VINET,428.0, ...],[10249,TOMSP,1842.0, ...],[10250,HANAR,1523.4999898076057, ...
2	[[10258,ERNSH,1603.9999940246344, ...],[10259,CENTC,100.0, ...],[10260,OTTIK,1461.75, ...
3	[[10268,GROSR,1098.0, ...],[10269,WHITC,626.9999995082617, ...],[10270,WARTH,1350.0, ...
4	[[10278,BERGS,1480.0, ...],[10279,LEHMS,348.75, ...],[10280,BERGS,596.0, ...], ...]
5	[[10288,REGGC,71.99999988079071, ...],[10289,BSBEV,474.0, ...],[10290,COMMI,2165.0, ...]...
6	[[10298,BLONP,2625.0, ...],[10299,RICAR,345.0, ...],[10300,MAGAA,590.0, ...], ...]

Index	ORDERID	CLIENT	AMOUNT	ORDERDA...	SELLERID	POSTALCODE	DELIVERYDATE	SALAY
1	10248	VINET	428.0	2012-07...	3	111080	2012-07-16	32.38
2	10249	TOMSP	1842.0	2012-07...	1	440876	2012-07-10	11.61
3	10250	HANAR	1523.499...	2012-07...	2	754546	2012-07-12	65.83
4	10251	VICTE	624.9499...	2012-07...	1	690047	2012-07-15	41.34
5	10252	SUPRD	3559.499...	2012-07...	2	567889	2012-07-11	51.30000...
6	10253	HANAR	1428.0	2012-07...	2	545486	2012-07-16	58.17
7	10254	CHOPS	545.3999...	2012-07...	2	301256	2012-07-23	22.98
8	10255	RICSU	2450.0	2012-07...	3	120477	2012-07-15	148.33
9	10256	WELLI	510.0	2012-07...	2	873763	2012-07-17	13.97
10	10257	HILAA	1109.0	2012-07...	3	502234	2012-07-22	81.91

Index	Member
1	[ORDERID,CLIENT,AMOUNT, ...]
2	[SELLERID,NAME,STATE, ...]

Index	Member
1	ORDERID
2	CLIENT
3	AMOUNT
4	ORDERDATE

Index	Member
1	SELLERID
2	NAME
3	STATE
4	SALAY

Index	Col1	Col2	Col3	Col4
1	ORDERID	CLIENT	AMOUNT	ORDERDATE
2	10248	VINET	428.0	2012-07-04
3	10249	TOMSP	1842.0	2012-07-05
4	10250	HANAR	1523.499989...	2012-07-08
5	10251	VICTE	624.9499997...	2012-07-08
6	10252	SUPRD	3559.499998...	2012-07-09
7	10253	HANAR	1428.0	2012-07-10
8	10254	CHOPS	545.3999973...	2012-07-11
9	10255	RICSU	2450.0	2012-07-12
10	10256	WELLI	510.0	2012-07-15
11	10257	HILAA	1109.0	2012-07-16
12	SELLERID	POSTALCODE	DELIVERYDATE	SALAY
13	3	111080	2012-07-16	32.38
14	1	440876	2012-07-10	11.61
15	2	754546	2012-07-12	65.83
16	1	690047	2012-07-15	41.34
17	2	567889	2012-07-11	51.30000000...
18	2	545486	2012-07-16	58.17
19	2	301256	2012-07-23	22.98
20	3	120477	2012-07-15	148.33
21	2	873763	2012-07-17	13.97
22	3	502234	2012-07-22	81.91

Page1

Page2

Row replication – Example

Make 3 copies of the records in order, and show them in a report. The diagram is as follows:

Report format

Index	ORDERID	CLIENT	AMOUNT	ORDERDATE	SELLERID	POSTALCODE	DELIVERYDATE	SALAY
1	10248	VINET	428.0	2012-07-04	3	111080	2012-07-16	32.38
2	10248	VINET	428.0	2012-07-04	3	111080	2012-07-16	32.38
3	10248	VINET	428.0	2012-07-04	3	111080	2012-07-16	32.38
4	10249	TOMSP	1842.0	2012-07-05	1	440876	2012-07-10	11.61
5	10249	TOMSP	1842.0	2012-07-05	1	440876	2012-07-10	11.61
6	10249	TOMSP	1842.0	2012-07-05	1	440876	2012-07-10	11.61
7	10250	HANAR	1523.4999...	2012-07-08	2	754546	2012-07-12	65.83
8	10250	HANAR	1523.4999...	2012-07-08	2	754546	2012-07-12	65.83
9	10250	HANAR	1523.4999...	2012-07-08	2	754546	2012-07-12	65.83
10	10251	VICTE	624.94999...	2012-07-08	1	690047	2012-07-15	41.34
11	10251	VICTE	624.94999...	2012-07-08	1	690047	2012-07-15	41.34
12	10251	VICTE	624.94999...	2012-07-08	1	690047	2012-07-15	41.34
13	10252	SUPRD	3559.4999...	2012-07-09	2	567889	2012-07-11	51.300000...
14	10252	SUPRD	3559.4999...	2012-07-09	2	567889	2012-07-11	51.300000...
15	10252	SUPRD	3559.4999...	2012-07-09	2	567889	2012-07-11	51.300000...
16	10253	HANAR	1428.0	2012-07-10	2	545486	2012-07-16	58.17
17	10253	HANAR	1428.0	2012-07-10	2	545486	2012-07-16	58.17
18	10253	HANAR	1428.0	2012-07-10	2	545486	2012-07-16	58.17

Source data table

Index	ORDERID	CLIENT	AMOUNT	ORDERDATE	SELLERID	POSTALCODE	DELIVERYDATE	SALAY
1	10248	VINET	428.0	2012-07-04	3	111080	2012-07-16	32.38
2	10249	TOMSP	1842.0	2012-07-05	1	440876	2012-07-10	11.61
3	10250	HANAR	1523.4999...	2012-07-08	2	754546	2012-07-12	65.83
4	10251	VICTE	624.94999...	2012-07-08	1	690047	2012-07-15	41.34
5	10252	SUPRD	3559.4999...	2012-07-09	2	567889	2012-07-11	51.300000...
6	10253	HANAR	1428.0	2012-07-10	2	545486	2012-07-16	58.17
7	10254	CHOPS	545.39999...	2012-07-11	2	301256	2012-07-23	22.98
8	10255	RICSU	2450.0	2012-07-12	3	120477	2012-07-15	148.33
9	10256	WELLI	510.0	2012-07-15	2	873763	2012-07-17	13.97
10	10257	HILAA	1109.0	2012-07-16	3	502234	2012-07-22	81.91
11	10258	ERNSH	1603.9999...	2012-07-17	1	801009	2012-07-23	140.51
12	10259	CENTC	100.0	2012-07-18	3	705022	2012-07-19	3.25
13	10260	OTTIK	1461.75	2012-07-19	1	140739	2012-07-29	55.09
14	10261	QUEDE	440.0	2012-07-19	2	238967	2012-07-30	3.0500000...
15	10262	RATTC	583.19999...	2012-07-22	3	871610	2012-07-25	48.29

	A
1	=myDB.query("select * from orders")
2	=A1.conj([~]*3)

[] represents a sequence (ordered set), [~] represents the A1 current record as a single member sequence, [~] * 3 makes 3 copies of the current record, function conj performs calculation on each record of A1, and finally merges.

Using condition to control the format of grouping table

A grouping table is displayed with the grouping field sellerid, detailed fields client and amount. The final format of the report is as follows:

1. In each group of details, the "+" sign needs to be displayed from item 2 to the end, but not for item 1.
2. If there is more than one detail in each group, the sum of amount will be displayed at the end of the group. If there is only one detail, the sum will not be displayed.

Index	item	value
1	Florida	16000
2	+Pennsylvania	12000
3	Administration SUBTOTAL:	28000
4	New York	11000
5	+Texas	11000
6	Finance SUBTOTAL:	22000
7	California	12000
8	Illinois	12000
9	+New York	25000
10	+Texas	12000
11	+New Jersey	12000
12	+Nevada	12000
13	+Pennsylvania	12000
14	Production SUBTOTAL:	85000
15	Texas	32000
16	+Pennsylvania	24000
17	+New York	24000
18	R&D SUBTOTAL:	80000
19	California	62000
20	+Texas	36000

Sum of the group

There is only one detail in the group, and the sum is not displayed.

Index	SELLERID	CLIENT	AMOUNT
1	Finance	New York	11000
2	R&D	Texas	32000
3	Sales	California	62000
4	R&D	Pennsylvania	24000
5	Sales	Texas	36000
6	Administration	Florida	16000
7	Finance	Texas	11000
8	Production	Illinois	12000
9	Administration	Pennsylvania	12000
10	Technology	Pennsylvania	13000
11	Sales	Pennsylvania	12000
12	Production	New York	25000
13	Production	Texas	12000
14	Sales	Alabama	12000
15	Sales	Illinois	12000
16	Sales	Kansas	12000
17	HR	California	12000
18	Sales	Michigan	12000
19	R&D	New York	24000
20	Sales	New York	12000

Using condition to control the format of grouping table – Example

Query the database, group by sellerid, and access each group of data by loop. In the loop, append the details of the current group to the empty sequence table A2. If the sequence number is greater than 1, then add "+" before the client. If the number of records in the current group is greater than 1, then append subtotal to A2.

	A	B	C
1	=myDB.query("SELECT SELLERID,CLIENT,AMOUNT FROM ORDER WHERE AMOUNT>?",arg)		
2	=create(item,value)		
3	for A1.group(SELLERID)	>A2.insert(0:A3,if(#>1,"+")+CLIENT,AMOUNT)	
4		if A3.len()>1	>A2.insert(0,A3.SELLERID+" SUBTOTAL:",A3.sum(AMOUNT))
5	return A2		

A2: Create empty sequence table

Index	item	value

A3: Group by sellerid

Index	SELLERID	CLIENT	AMOUNT
1	Administration	Florida	16000
2	Administration	Pennsylvania	12000

B3: Append the details to A2 according to requirement.

Index	item	value
1	Florida	16000
2	+Pennsylvania	12000

B4, C4: Append subtotal according to requirement.

Index	item	value
1	Florida	16000
2	+Pennsylvania	12000
3	Administration SUBTOTAL:	28000

Insert a sub table into the main table dynamically

The report needs to query the main table according to the ApplicationName and present the data in the form of a list. There are multiple status field values corresponding to each record in the main table, but no more than 5. You need to insert them horizontally between the phone and decline fields in the main table, and then name them questionno1 and questionno2 till questionNo5. If all data in a column is empty, the column is not displayed. The sample is as follows:

Report format

Index	Applicatio...	User	Phone	QuestionNo1	QuestionNo2	QuestionNo3	QuestionNo4	QuestionNo5	Decline
1	mfc	Bill	+ 700000...	yes	yes	no			1
2	mfc	John	+ 187612...	yes					2
3	mfc	Tim	+ 008613...	no	no				4

Filtering condition: ApplicationName='mfc'

Main table

Index	tID	ApplicationName	User	Phone	Decline
1	A01	mfc	Bill	+ 70000000	1
2	A02	mfc	John	+ 18761221	2
3	A03	java	Jack	+ 8014001231	6
4	A04	mfc	Tim	+ 008613133...	4
5	A05	db	John	+ 18761221	8

Foreign key
1:N

Sub table

Index	qID	tID	status
1	1	A01	yes
2	2	A01	no
3	3	A01	yes
4	4	A02	yes
5	5	A03	no
6	6	A04	no
7	7	A04	no
8	8	A05	yes

Insert a sub table into the main table dynamically – Example

A1: Execute SQL, take the join data of main and sub table. Arg1 is the parameter from report. If arg1="mfc", the calculation result of A1 is as the right figure:

Index	tID	ApplicationName	User	Phone	Decline	qID	status
1	A01	mfc	Bill	+ 70000000	1	1	yes
2	A01	mfc	Bill	+ 70000000	1	3	yes
3	A01	mfc	Bill	+ 70000000	1	2	no
4	A02	mfc	John	+ 18761221	2	4	yes
5	A04	mfc	Tim	+ 00861313...	4	6	no
6	A04	mfc	Tim	+ 00861313...	4	7	no

	A	B
1	=myDB.query@x("select * from dColThread t,dColQuestion q where t.tID=q.tID and t.ApplicationName=?",arg1)	
2	=A2.group(tID)	
3	=create(ApplicationName,User,Phone,QuestionNo1,QuestionNo2,QuestionNo3,QuestionNo4,QuestionNo5,Decline)	
4	for A2	=A4.(status)|["","","","",""]
5		=A3.record(A4.ApplicationName|A4.User|A4.Phone|B4.to(5)|A4.Decline)
6	return A3	

Index	Member
1	[[A01,mfc,Bill, ...],[A01,mfc,Bill, ...],[A01,mfc,Bill, ...
2	[[A02,mfc,John, ...]]
3	[[A04,mfc,Tim, ...],[A04,mfc,Tim, ...]]

A2: Group by tID, each group is one piece of main table record and the corresponding sub table records, as the right figure:

Index	tID	Applicatio...	User	Phone	Decline	qID	status
1	A01	mfc	Bill	+ 70000...	1	1	yes
2	A01	mfc	Bill	+ 70000...	1	3	yes
3	A01	mfc	Bill	+ 70000...	1	2	no

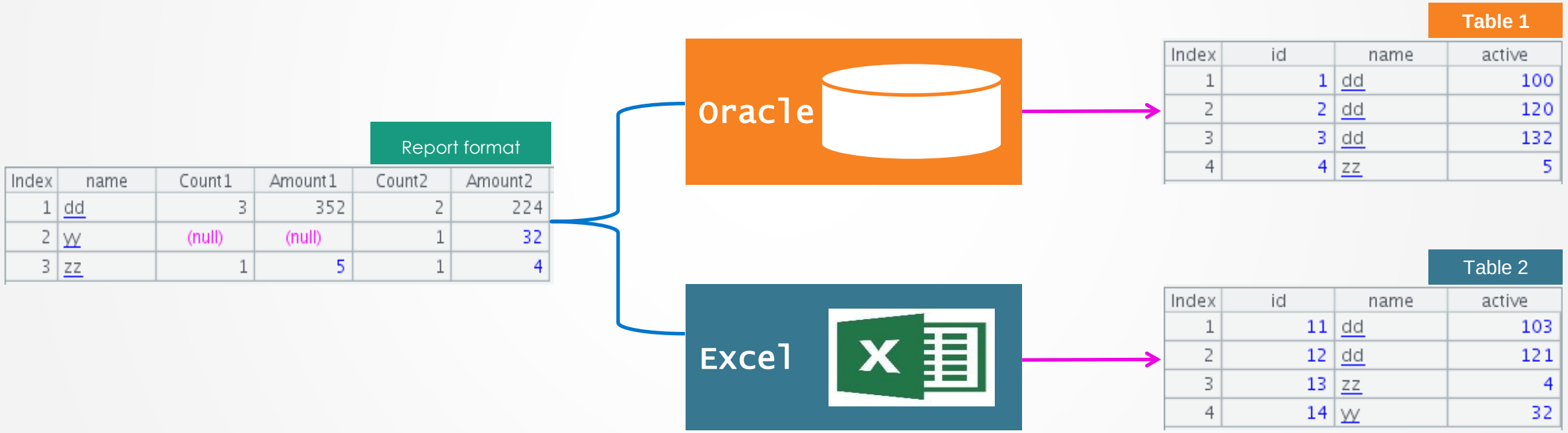
Index	tID	Applicatio...	User	Phone	Decline	qID	status
1	A02	mfc	John	+ 18761...	2	4	yes

Index	tID	Applicatio...	User	Phone	Decline	qID	status
1	A04	mfc	Tim	+ 00861...	4	6	no
2	A04	mfc	Tim	+ 00861...	4	7	no

Each loop of A2 inserts a record into the empty sequence table A3, the main table field is inserted directly, the sub table field is inserted after column to row conversion, and makes up at least 5 fields. A3 result is the sample table.

Horizontally splice columns

Table 1 is from the Oracle database, and table 2 is from the excel file, both of which have the same structure. You need to group table 1 and table 2 by name, count the members of each group and sum the active field of each group, and finally display them side by side in the report. The ideal sample is as follows:



Horizontally splice columns – Example

Idea: Take data from database and excel file respectively, make a full join of the two, and combine the required fields into a dataset. A5 stores the results of the merge.

Index	name	Count	Amount
1	dd	3	352
2	zz	1	5

Index	name	Count	Amount
1	dd	2	224
2	W	1	32
3	zz	1	4

	A
1	=myDB.query@x("select name,count(*) Count,sum(active) Amount from table1 group by name")
2	=file("D:\\table2.xlsx").importxls@t()
3	=A2.groups(name;count(~):Count,sum(active):Amount)
4	=join@f(A1,name;A3,name)
5	=A4.new(ifn(_1,_2).name:name,_1.Count:Count1,_1.Amount:Amount1,_2.Count:Count2,_2.Amount:Amount2)

Index	_1	_2
1	dd	dd
2	(null)	W
3	zz	zz

name	Count	Amount
dd	3	352

name	Count	Amount
dd	2	224

Index	name	Count1	Amount1	Count2	Amount2
1	dd	3	352	2	224
2	W	(null)	(null)	1	32
3	zz	1	5	1	4

Calculation between columns of crosstab

The database table store stores the sales volume of various products in 2014 and 2015. A cross table is needed to present the annual sales volume of each product and calculate the annual growth rate of each product. The ideal sample table and some source data are as follows:

Report format

	2014	2015	Growth Rate
Book	35	67	0.914285714
Pencil	56	50	-0.10714285

Source data table

Index	year	item	quantity
1	2014	Book	35
2	2015	Pencil	50
3	2014	Pencil	56
4	2015	Book	67

Calculation between columns of crosstab – Example

The columns of the crosstab are generated dynamically. When calculating between columns, they need to be referenced dynamically twice. It is difficult to use the report script to implement. At this time, the result of calculation between columns can be added to the source data in advance by esProc, and then the requirements can be realized only by designing a simple crosstab.

Fetch data and sort

Index	year	item	quantity
1	2014	Book	35
2	2015	Book	67
3	2014	Pencil	56
4	2015	Pencil	50

	A
1	=myDB.query@x("select * from store order by item,year")
2	=A1.group(item).run(A1.record(["Growth Rate",item,~(2).quantity/~(1).quantity-1]))

Index	year	item	quantity
1	2014	Book	35
2	2015	Book	67

Index	Member
1	[[2014,Book,35],[2015,Book,67]]
2	[[2014,Pencil,56],[2015,Pencil,50]]

Index	year	item	quantity
1	2014	Pencil	56
2	2015	Pencil	50

The calculation result of A2 is as follows:

Index	year	item	quantity
1	2014	Book	35
2	2015	Book	67
3	2014	Pencil	56
4	2015	Pencil	50
5	Growth Rate	Book	0.9142857142...
6	Growth Rate	Pencil	-0.107142857...

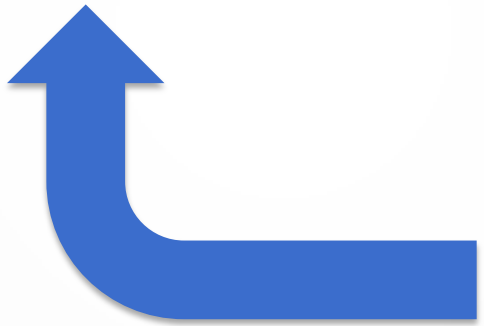
Transposition

The database table sales stores the order data, which needs to be expanded horizontally according to the month, vertically according to the aggregate value, and finally presented as a cross table or grouping table. Generally, such requirements need to be processed by row and column transposition. The ideal sample table and some source data are as follows:

Report format

Index	SUBTOTAL	1	2	3	4	5	6	7	8	9	10	11	12
1	OSUM	61258.06...	38483.63...	38547.21...	53032.95...	53781.28...	36362.80...	51020.85...	47287.66...	55629.24...	66749.22...	43533.80...	71398.42...
2	OMAX	11188.4	4924.134...	10495.6	9921.299...	10191.7	2944.399...	6475.400...	5510.592...	5256.5	10164.8	4529.8	6635.274...
3	OMIN	49.8	174.9	147.0	136.8	110.0	155.0	23.79999...	55.79999...	45.0	93.5	52.34999...	12.5
4	OCOUNT	33	29	30	31	32	30	33	33	37	38	34	48

Horizontal expansion: from January to December
Vertical expansion: according to sum, Max, min, count



Detailed data

Index	ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE
1	10248	VINET	3	428.0	2012-07-04
2	10249	TOMSP	1	1842.0	2012-07-05
3	10250	HANAR	2	1523.499...	2012-07-08
4	10251	VICTE	1	624.9499...	2012-07-08
5	10252	SUPRD	2	3559.499...	2012-07-09
6	10253	HANAR	2	1428.0	2012-07-10
7	10254	CHOPS	2	545.3999...	2012-07-11
8	10255	RICSU	3	2450.0	2012-07-12
9	10256	WELLI	2	510.0	2012-07-15
10	10257	HILAA	3	1109.0	2012-07-16
11	10258	ERNSH	1	1603.999...	2012-07-17
12	10259	CENTC	3	100.0	2012-07-18
13	10260	OTTIK	1	1461.75	2012-07-19
14	10261	QUEDE	2	440.0	2012-07-19
15	10262	RATTC	3	583.1999...	2012-07-22

Transposition – Example

First, calculate the total amount, the maximum order amount, the minimum order amount and the total order number of each month in the specified year, and transpose the data into 13 columns and 4 rows, that is, the four aggregations are the first column, the column name is subtotal, each month occupies one column, the column names are 1, 2, 3, 4…

	A
1	=myDB.query("select month(ORDERDATE) as MONTH,sum(AMOUNT) as OSUM,max(AMOUNT) as OMAX, min(AMOUNT) as OMIN ,count(ORDERID) as OCOUNT from sales where year(ORDERDATE)=? group by MONTH order by MONTH",argYear)
2	=["OSUM","OMAX","OMIN","OCOUNT"].new(~:SUBTOTAL,${to(A1.len()).string()})
3	=A1.run(A2.field(#+1,OSUM|OMAX|OMIN|OCOUNT))
4	return A2

Index	MONTH	OSUM	OMAX	OMIN	OCOUNT
1	1	61258.0699...	11188.4	49.8	33
2	2	38483.6349...	4924.13496...	174.9	29
3	3	38547.2199...	10495.6	147.0	30
4	4	53032.9524...	9921.29997...	136.8	31
5	5	53781.2899...	10191.7	110.0	32
6	6	36362.8024...	2944.39998...	155.0	30
7	7	51020.8574...	6475.40000...	23.7999998...	33
8	8	47287.6699...	5510.59246...	55.7999999...	33
9	9	55629.2424...	5256.5	45.0	37
10	10	66749.2259...	10164.8	93.5	38
11	11	43533.8089...	4529.8	52.3499999...	34
12	12	71398.4284...	6635.27499...	12.5	48

Index	SUBTOTAL	1	2	3	4	5	6	7	8	9	10	11	12
1	OSUM	61258.06...	38483.63...	38547.21...	53032.95...	53781.28...	36362.80...	51020.85...	47287.66...	55629.24...	66749.22...	43533.80...	71398.42...
2	OMAX	11188.4	4924.134...	10495.6	9921.299...	10191.7	2944.399...	6475.400...	5510.592...	5256.5	10164.8	4529.8	6635.274...
3	OMIN	49.8	174.9	147.0	136.8	110.0	155.0	23.79999...	55.79999...	45.0	93.5	52.34999...	12.5
4	OCOUNT	33	29	30	31	32	30	33	33	37	38	34	48

Innovation makes progress!