

# Cummulative Measure

by Sudheer Sharma - Friday, February 13, 2009

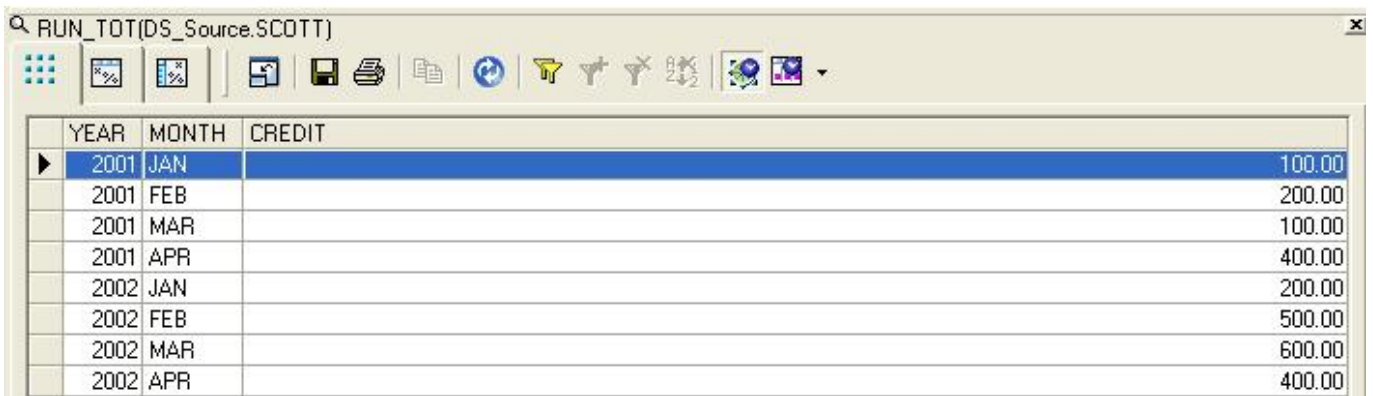
<http://dwhnotes.com/data-integrator/cm>

What do you think of this post?

[Awesome \(1\)](#) [Interesting \(0\)](#) [Useful \(2\)](#)

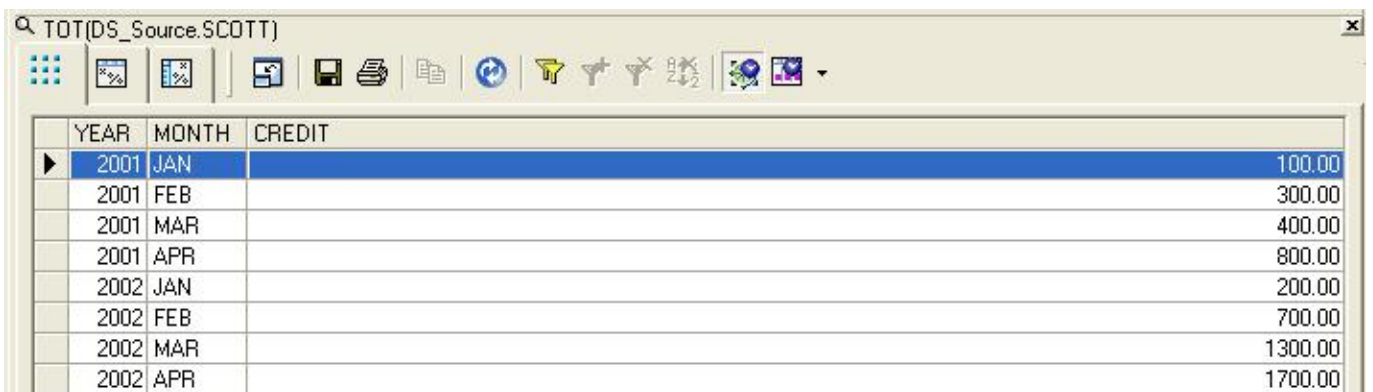
Calculating a cummulative measure is common query in DW, but there is no straight forward way in DI. Either you have to write a custom function or you have to design using query transform.

I have a source data like this,



YEAR	MONTH	CREDIT
2001	JAN	100.00
2001	FEB	200.00
2001	MAR	100.00
2001	APR	400.00
2002	JAN	200.00
2002	FEB	500.00
2002	MAR	600.00
2002	APR	400.00

I want target data like this,



YEAR	MONTH	CREDIT
2001	JAN	100.00
2001	FEB	300.00
2001	MAR	400.00
2001	APR	800.00
2002	JAN	200.00
2002	FEB	700.00
2002	MAR	1300.00
2002	APR	1700.00

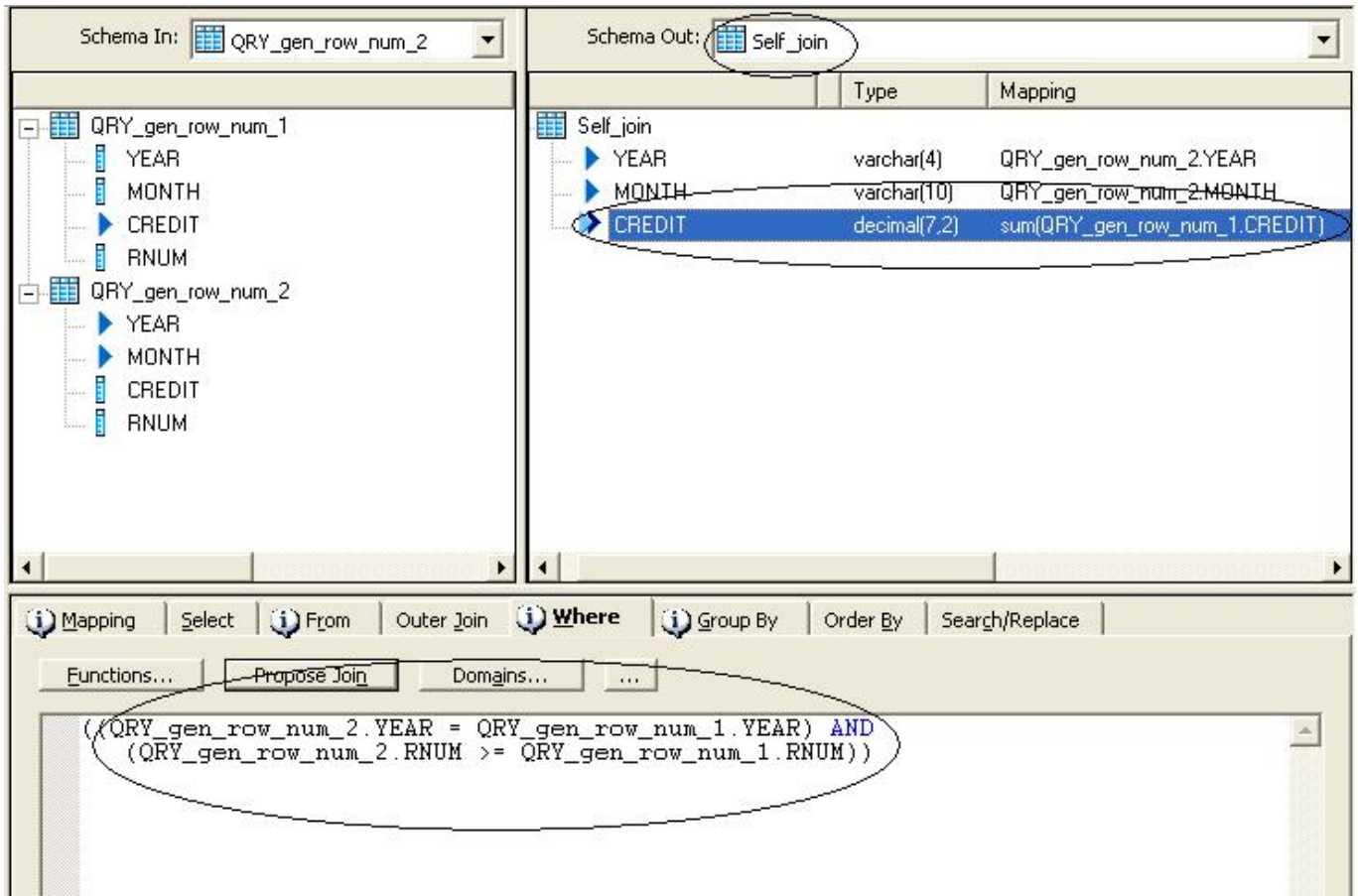
Well will see the design, here i have two QRY instances and generating a row number in each instance

	Type	Mapping
YEAR	varchar(4)	RUN_TOT.YEAR
MONTH	varchar(10)	RUN_TOT.MONTH
CREDIT	decimal(7,2)	RUN_TOT.CREDIT
RNUM	int	gen_row_num()

for: QRY\_gen\_row\_num\_1.RNUM

gen\_row\_num()

will do a self join on the basis of RNUM and then summing up the credit value



and grouping the values based on Year & Month in the same QRY instance



and then apply order by on the basis of Year and Credit in another QRY instance

The screenshot shows a SQL query editor interface. At the top, there are two dropdown menus: 'Schema In:' set to 'Self\_join' and 'Schema Out:' set to 'Order\_by'. Below these are two tree views. The left tree view shows the 'Self\_join' schema with columns 'YEAR', 'MONTH', and 'CREDIT'. The right tree view shows the 'Order\_by' schema with columns 'YEAR', 'MONTH', and 'CREDIT'. Below the tree views is a toolbar with buttons for 'Mapping', 'Select', 'From', 'Outer Join', 'Where', 'Group By', 'Order By', and 'Search/R'. At the bottom, there is a table with the following content:

Column	Sort
Self_join.YEAR	Ascending
Self_join.CREDIT	Ascending

Execute the job, you'll get the desired output.

What do you think of this post?

[Awesome \(1\)](#) [Interesting \(0\)](#) [Useful \(2\)](#)