

TAI Query Reporter V 1.1.0.6

New functionality...

HAVING Clause

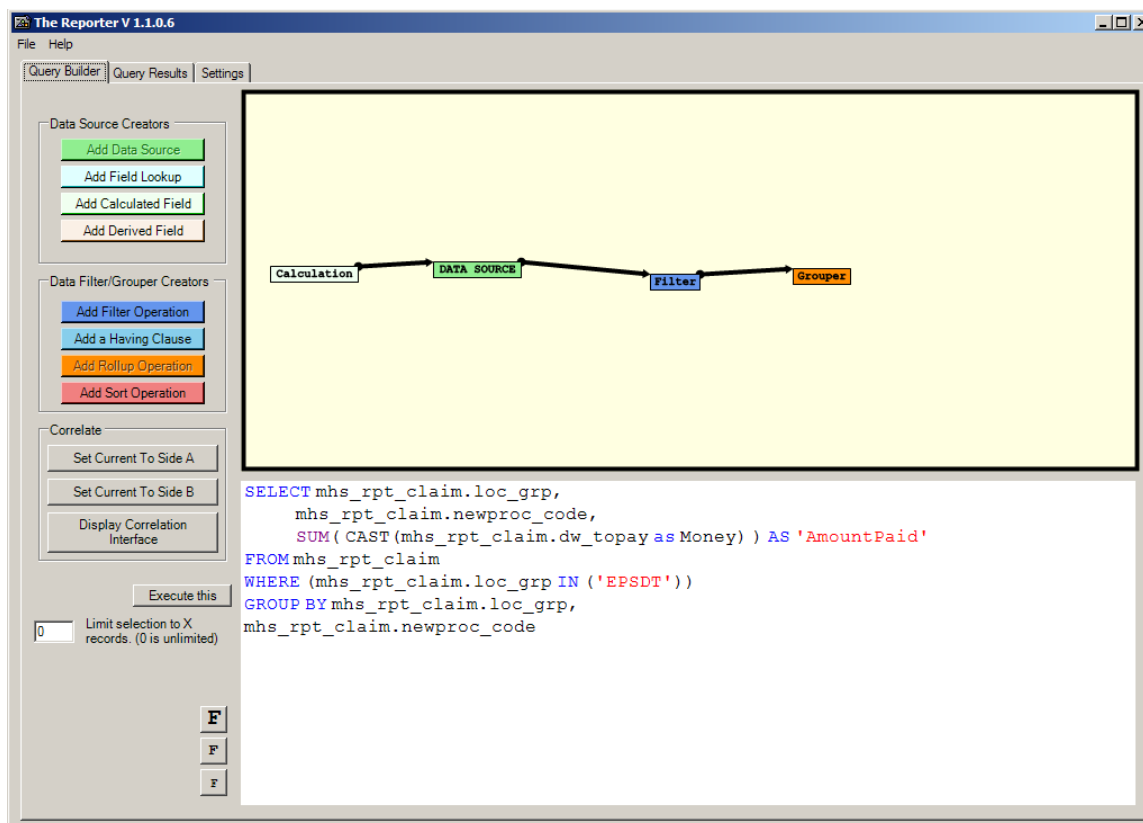
Reason:

There are often times when you have crafted a query that includes some aggregate data like SUM() and COUNT() and AVG() that are in actuality applied to whole groups of records as those records are grouped by the Grouper functionality. Meaning the SUM() or COUNT() or AVG() function is applied to more than a single record as the query gets executed in the database.

In these cases it is not possible in standard SQL syntax to place a condition on these aggregations within the where clause of the sql statement. What is necessary is to create a HAVING clause in SQL syntax and now the TAI Query Reporter can do just that.

Example Usage....

Below is a sample query that has been built to sum up claims paid for in a certain timeframe, by service code. In some cases the sum of the monies paid out yields 0 dollars. This means that a claim may have been paid but then retracted. Since the financial impact of this activity is 0 it is often desirable to remove this noise from the result set. (If you are of a financial mind that is)



The screenshot displays the 'The Reporter V 1.1.0.6' application window. The interface includes a menu bar (File, Help), a tabbed area (Query Builder, Query Results, Settings), and a central workspace. On the left, there are three panels: 'Data Source Creators' with buttons for 'Add Data Source', 'Add Field Lookup', 'Add Calculated Field', and 'Add Derived Field'; 'Data Filter/Grouper Creators' with buttons for 'Add Filter Operation', 'Add a Having Clause', 'Add Rollup Operation', and 'Add Sort Operation'; and 'Correlate' with buttons for 'Set Current To Side A', 'Set Current To Side B', and 'Display Correlation Interface'. Below these panels is an 'Execute this' button and a 'Limit selection to X records. (0 is unlimited)' input field. The central workspace contains a diagram with four boxes: 'Calculation', 'DATA SOURCE', 'Filter', and 'Grouper', connected by arrows. Below the workspace, a SQL query is displayed:

```
SELECT mhs_rpt_claim.loc_grp,
       mhs_rpt_claim.newproc_code,
       SUM( CAST(mhs_rpt_claim.dw_topay as Money) ) AS 'AmountPaid'
FROM mhs_rpt_claim
WHERE (mhs_rpt_claim.loc_grp IN ('EPSDT'))
GROUP BY mhs_rpt_claim.loc_grp,
         mhs_rpt_claim.newproc_code
```

Here we are simply summing the money paid for any claim that has been classified as 'EPSDT' and grouping by the particular service code. The results of this query yield some values that are 0 dollars, as shown below

loc_grp	newproc_code	AmountPaid
EPSDT	90801	12245.0000
EPSDT	W0613	6689.5200
EPSDT	W0987	110.0000
EPSDT	W1867	116359.5700
EPSDT	W7038	4445.0000
EPSDT	Y9600	5504.0000
EPSDT	Y9601	0
EPSDT	Y9602	0
EPSDT	Y9607	8264078.7300
EPSDT	Y9608	76035.5000
EPSDT	Y9609	2100970.2000
EPSDT	Y9610	1187234.4000
EPSDT	Y9611	0
EPSDT	Z9889	18079.0000
EPSDT	Z9892	30878.0000
EPSDT	Z9893	920.0000
EPSDT		-11352.0000
EPSDT	P5001	0
EPSDT	W7039	555.0000
EPSDT	Y9603	0
EPSDT	Y9606	399018.8200
EPSDT	Y9625	0

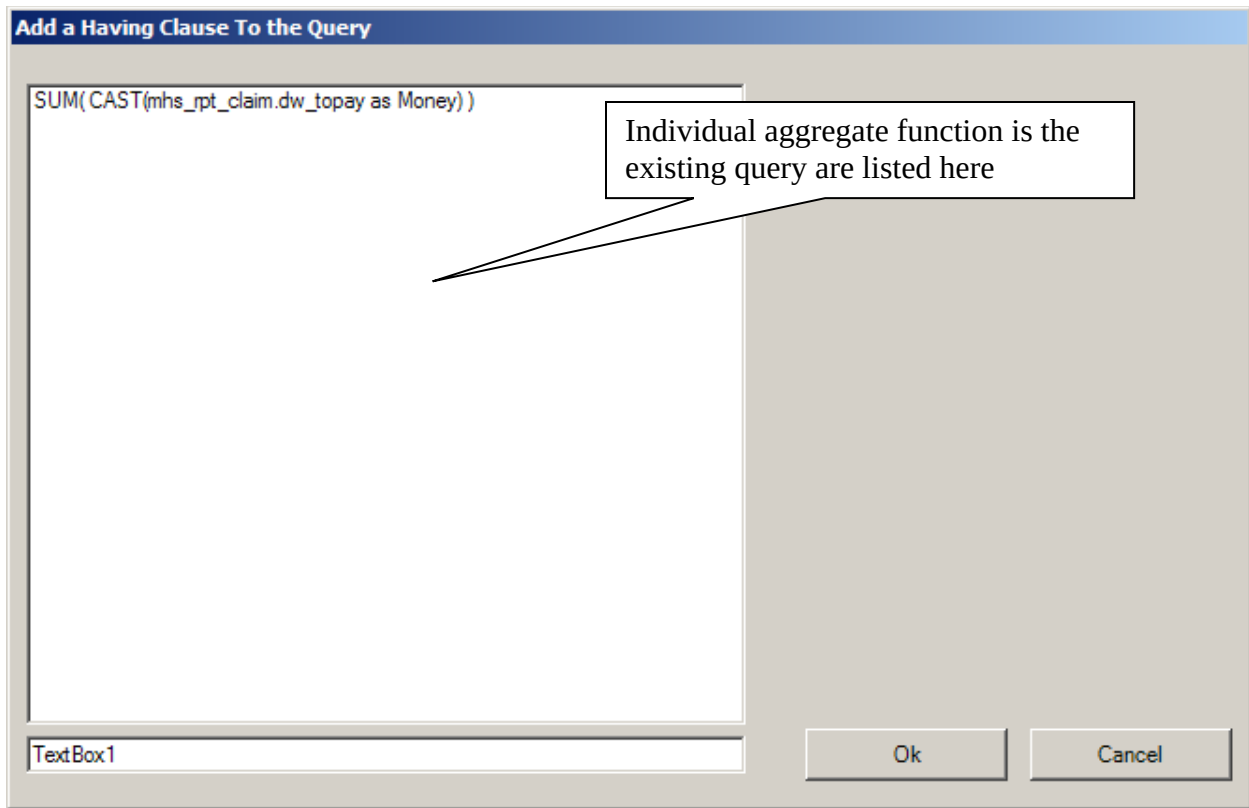
Back on the Main query screen a new button is available to Add a Having clause...

**New function
Add a Having Clause**

```

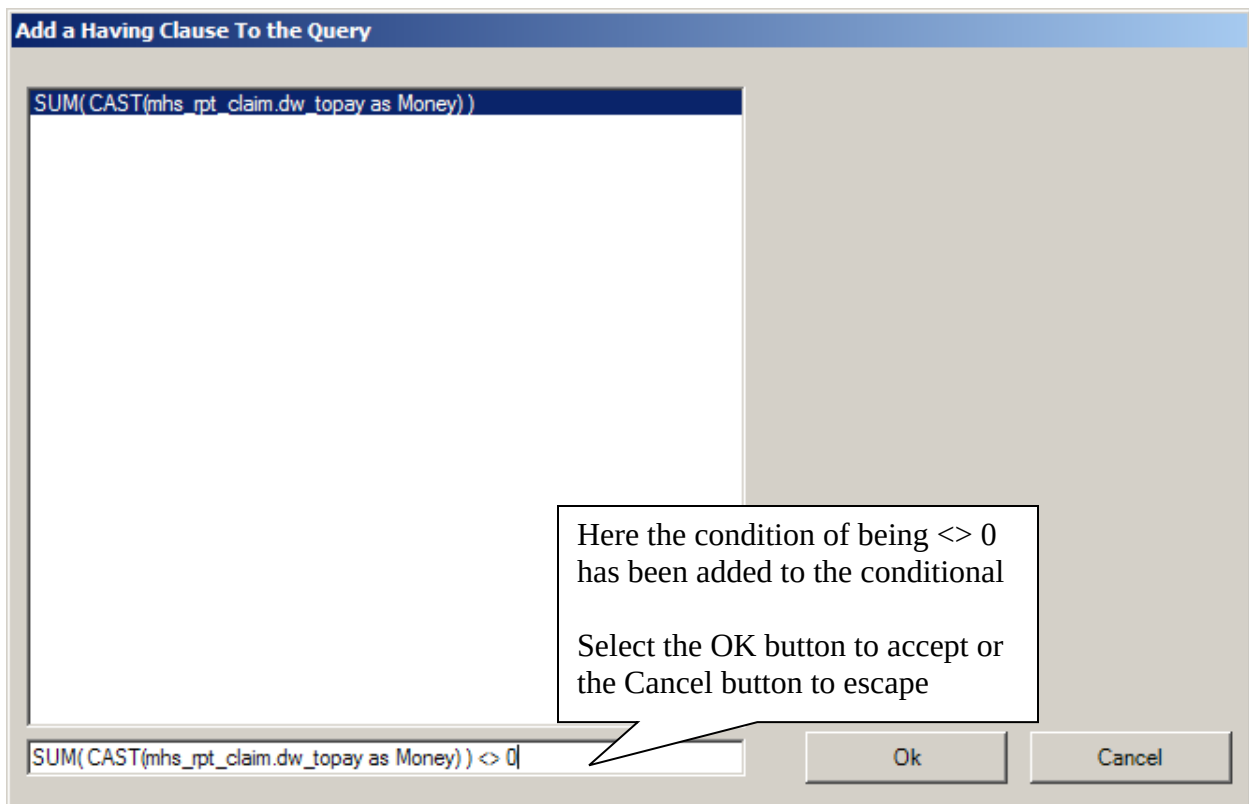
SELECT mhs_rpt_claim.loc_grp,
       mhs_rpt_claim.newproc_code,
       SUM( CAST(mhs_rpt_claim.dw_topay as Money) ) AS 'AmountPaid'
FROM mhs_rpt_claim
WHERE (mhs_rpt_claim.loc_grp IN ('EPSDT'))
GROUP BY mhs_rpt_claim.loc_grp,
         mhs_rpt_claim.newproc_code
  
```

Selecting this new button will present you with the following dialog



In the above example the simple sum() aggregate is the only one in the existing query, so it's the only one listed.

To craft a having clause simple select the desired aggregate and then type your desired conditional after the selected aggregate in the textbox below.



The result of the Having clause being added to the query is show below.

The screenshot shows the 'Query Builder' tab in 'The Reporter V 1.1.0.6'. On the left, there are panels for 'Data Source Creators', 'Data Filter/Groupers Creators', and 'Correlate'. The main area displays a query chain diagram with nodes: Calculation, DATA SOURCE, Filter, and Groupers. A 'Having' node is positioned above the chain. A callout points to this node with the text: 'The having clauses chicklet appears in the graph'. Below the diagram is a SQL query:

```
SELECT mhs_rpt_claim.loc_grp,
       mhs_rpt_claim.newproc_code,
       SUM( CAST(mhs_rpt_claim.dw_topay as Money) ) AS 'Am
FROM mhs_rpt_claim
WHERE (mhs_rpt_claim.loc_grp IN ('EPSDT'))
GROUP BY mhs_rpt_claim.loc_grp,
         mhs_rpt_claim.newproc_code
HAVING SUM( CAST(mhs_rpt_claim.dw_topay as Money) ) <> 0
```

A second callout points to the HAVING clause in the query with the text: 'The added having clause appears in the query panel'. At the bottom, there are three buttons labeled 'F' and a checkbox for 'Limit selection to X records. (0 is unlimited)'.

Note...

Because having clauses are overriding the entire query they are not connected to the chain as the are conceptually attached to the entire query rather than a single point along the query chain.

You can only have a single having clause as of this time. If you want to alter the clause. Remove the existing clause and re-add a new one.

Results...

The screenshot shows the 'Query Results' tab in 'The Reporter V 1.1.0.6'. It displays a 'Results Grid' with the following data:

loc_grp	newproc_code	AmountPaid
EPSDT		-11352.0000
EPSDT	W7039	555.0000
EPSDT	Y9606	399018.8200
EPSDT	90801	12245.0000
EPSDT	W0613	6689.5200
EPSDT	W0987	110.0000
EPSDT	W1867	116359.5700
EPSDT	W7038	4445.0000
EPSDT	Y9600	5504.0000
EPSDT	Y9607	8264078.7300
EPSDT	Y9608	76035.5000
EPSDT	Y9609	2100970.2000
EPSDT	Y9610	1187234.4000
EPSDT	Z9889	18079.0000
EPSDT	Z9892	30878.0000
EPSDT	Z9893	920.0000

Note that the 0 dollar results are now missing from the end result.

Also in this new version is the ability to directly print the end result of a query by selecting Either PRINT or PRINT PREVIEW from the result grids own context menu.

Doing a print preview will show you the grids results paginated to fit onto multiple pages if the grid results are larger than one page. The date of the print and the page numbers are also printed.

Wednesday, April 08, 2009
12:21:20 PM

loc_grp	newproc_code	AmountPaid
EPSDT		-11352.0000
EPSDT	W7039	555.0000
EPSDT	Y9605	399018.8200
EPSDT	90801	12245.0000
EPSDT	W0613	8689.5200
EPSDT	W0987	110.0000
EPSDT	W1867	116359.5700
EPSDT	W7038	4445.0000
EPSDT	Y9600	5504.0000
EPSDT	Y9607	8264078.7300
EPSDT	Y9608	76035.5000
EPSDT	Y9609	2100970.2000
EPSDT	Y9610	1187234.4000
EPSDT	Z9689	18079.0000
EPSDT	Z9692	30878.0000
EPSDT	Z9693	920.0000

Page 1

