

## Re: the "having" clause

**Source:**

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-03/1812.html>

---

**From:** Roji. P. Thomas (*lazydragon\_at\_nowhere.com*)

**Date:** 03/08/04

Date: Mon, 8 Mar 2004 15:54:17 +0530

Here is the original post from CELKO.

-----

Here is how

a SELECT works in SQL ... at least in theory. Real products will optimize things when they can.

- a) Start in the FROM clause and build a working table from all of the joins, unions, intersections, and whatever other table constructors are there. The table expression > AS <correlation name> option allows you give a name to this working table which you then have to use for the rest of the containing query.
- b) Go to the WHERE clause and remove rows that do not pass criteria; that is, that do not test to TRUE (reject UNKNOWN and FALSE). The WHERE clause is applied to the working set in the FROM clause.
- c) Go to the optional GROUP BY clause, make groups and reduce each group to a single row, replacing the original working table with the new grouped table. The rows of a grouped table must be group characteristics: (1) a grouping column (2) a statistic about the group (i.e. aggregate functions) (3) a function or (4) an expression made up those three items.
- d) Go to the optional HAVING clause and apply it against the grouped working table; if there was no GROUP BY clause, treat the entire table as one group.
- e) Go to the SELECT clause and construct the expressions in the list. This means that the scalar subqueries, function calls and expressions in the SELECT are done after all the other clauses are done. The "AS" operator can also give names to expressions in the SELECT list. These new names come into existence all at once, but after the WHERE clause, GROUP BY clause and HAVING clause has been executed; you cannot use them in the SELECT list or the WHERE clause for that reason.

## microsoft.public.sqlserver.programming: Re: the "having" clause

If there is a SELECT DISTINCT, then redundant duplicate rows are removed. For purposes of defining a duplicate row, NULLs are treated as matching (just like in the GROUP BY).

f) Nested query expressions follow the usual scoping rules you would expect from a block structured language like C, Pascal, Algol, etc. Namely, the innermost queries can reference columns and tables in the queries in which they are contained.

As you can see, things happen "all at once" in SQL, not from left to right as they would in a sequential file/procedural language model. In those languages, these two statements produce different results:

```
READ (a, b, c) FROM File_X;  
READ (c, a, b) FROM File_X;
```

while these two statements return the same data:

```
SELECT a, b, c FROM Table_X;  
SELECT c, a, b FROM Table_X;
```

Think about what a confused mess this statement is in the SQL model.

```
SELECT f(c2) AS c1, f(c1) AS c2 FROM Foobar;
```

That is why such non-sense is illegal syntax.

hint: Start using the AS operator for column and table aliases instead of the proprietary overloaded equal sign. It will make your code more readable and portable.

```
--  
Roji. P. Thomas  
SQL Server Programmer  
"Amy" <XXXNOSPAMXXX___l.a@usa.com> wrote in message  
news:eMMIpZPBEHA.2404@TK2MSFTNGP11.phx.gbl...  
> Hi toylet,  
>  
> Logically, The SELECT list is processed after the GROUP BY clause and the  
> HAVING filter.  
> So, When SQL Server processes the HAVING clause, the column alias is not  
yet  
> 'known'.  
> The only place you can use column aliases you define in the select list is  
> in the ORDER BY clause which is processed last.  
> This is not a SQL Server 'freak behaviour', it is by design and supports  
the  
> ANSI rules for query processing order.  
> Search this forum for an article by Celko called 'Select Sequence' in  
which  
> he explains the whole thing very clearly.  
> All you should do is change your query to:  
>  
> > select xx.pk, xx.amount, count(*) as recno  
> > from tx xx, tx yy  
> > where xx.pk>=yy.pk  
> > group by xx.pk, xx.amount
```

Re: the "having" clause

## microsoft.public.sqlserver.programming: Re: the "having" clause

```
> > having count(*) = 1
>
> HTH
>
> Amy
>
> "toylet" <toylet_at_mail.hongkong.com> wrote in message
> news:uH00lNPBEHA.3472@TK2MSFTNGP09.phx.gbl...
> > Why did SQL Server complain about "invalid column recno" in the
> > following query? It worked in another database tool I am using.
> >
> > select xx.pk, xx.amount, count(*) as recno
> > from tx xx, tx yy
> > where xx.pk>=yy.pk
> > group by xx.pk, xx.amount
> > having recno = 1
> >
> > --
> >   .~.   Might, Courage, Vision. In Linux We Trust.
> > / v \   http://www.linux-sxs.org
> > / ( _ ) \ Linux 2.4.22-xfs
> >   ^ ^   5:52pm up 2 days 2:02 load average: 0.99 0.97 0.96
>
>
```