

## Re: GROUP BY and performance

**Source:**

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-07/0742.html>

---

**From:** Jim Clark (*Reply\_To\_Group\_at\_NotAnAddress.com*)

**Date:** 07/02/04

Date: Fri, 2 Jul 2004 11:17:20 -0400

If you use the sample I provided and execute the query the result set will contain the person information for "Jim Bo" twice since he is a member of two families. If I want to get a list of all the people that are in any family for which include = 1 but do not want duplicates for those that are in two families then I must use DISTINCT or GROUP BY.

Essentially I want to GROUP BY personid. I know that there is only one value for every other column in the select list (firstname and lastname) for a given personid so it isn't necessary to do DISTINCT or GROUP BY every column in the select list or put every column other than personid in an aggregate function. I can get the desired results with any of the following queries:

```
SELECT DISTINCT person.personid, firstname, lastname
FROM person
  JOIN familymembership
    ON person.personid = familymembership.personid
  JOIN family
    ON family.familyid = familymembership.familyid
WHERE include = 1
```

or

```
SELECT person.personid, MAX(firstname), MAX(lastname)
FROM person
  JOIN familymembership
    ON person.personid = familymembership.personid
  JOIN family
    ON family.familyid = familymembership.familyid
WHERE include = 1
GROUP BY person.personid
```

or

```
SELECT person.personid, firstname, lastname
FROM person
  JOIN familymembership
```

```
ON person.personid = familymembership.personid
JOIN family
ON family.familyid = familymembership.familyid
WHERE include = 1
GROUP BY person.personid, firstname, lastname
```

But I think SQL Server will be doing unnecessary work looking at firstname and lastname. If the select list were much longer would one of the above be preferable to the others? Is there a commonly agreed upon standard for this type of query. I encounter it often and usually add DISTINCT.

"Adam Machanic" <amachanic@hotmail.\_removetoemail\_.com> wrote in message news:uOLY1PEYEHA.212@TK2MSFTNGP12.phx.gbl...

```
> Jim,
>
> Perhaps you can provide a better example? Given the presence of your
> person.personid in the SELECT list, unless a person can have multiple
names
> (or multiple values for other attributes given that you say the SELECT
list
> will be quite long), you shouldn't have a problem. If the person does
have
> multiple values possible, you will have to decide what to do with them...
> For instance, if you had a DATETIME value that could be duplicated:
>
> SELECT person.personid, firstname, lastname, MAX(SomeDateTime) AS
> LatestVisit
> FROM person
> JOIN familymembership
> ON person.personid = familymembership.personid
> JOIN family
> ON family.familyid = familymembership.familyid
> GROUP BY person.personid, firstname, lastname
> WHERE include = 1
>
> As for performance implications, it's difficult to evaluate that without
> testing it! A technique that sometimes helps (but is by no means always
> necessary) is to do something like:
>
> SELECT x.personid, firstname, lastname
> FROM person
> JOIN (SELECT DISTINCT personid
> FROM familymembership
> JOIN family
> ON family.familyid = familymembership.familyid
> WHERE include = 1) x (personid)
> ON (x.personid = person.personid)
>
> Or you could try using IN:
>
> SELECT x.personid, firstname, lastname
```

> *FROM person*  
> *WHERE personid IN (SELECT DISTINCT personid*  
> *FROM familymembership*  
> *JOIN family*  
> *ON family.familyid = familymembership.familyid*  
> *WHERE include = 1)*  
>  
>  
> *Sometimes one of these two techniques will yeild better execution plans*  
*than*  
> *using DISTINCT on the entire SELECT list. And sometimes performance will*  
*be*  
> *worse. It all depends on your data, indexes, server, and probably the*  
*phase*  
> *of the moon. Again, test it... There is no one right answer.*  
>  
>  
> *"Jim Clark" <Reply\_To\_Group@NotAnAddress.com> wrote in message*  
> *news:O\$\$fLCEYEHA.3112@tk2msftngp13.phx.gbl...*  
>> *This is a pretty contrived example but I think it is sufficient for my*  
>> *question. The script at the end of this message sets up the example.*  
>>  
>> *Consider the following query:*  
>>  
>> *SELECT person.personid, firstname, lastname*  
>> *FROM person*  
>> *JOIN familymembership*  
>> *ON person.personid = familymembership.personid*  
>> *JOIN family*  
>> *ON family.familyid = familymembership.familyid*  
>> *WHERE include = 1*  
>>  
>> *Assume I only want each person to show up one time in the result set. I*  
>> *have worked with some databases that would let me group by*  
*person.personid*  
>> *without having to group on the entire select list or use distinct while*  
>> *acknowledging that the query could return unexpected data for columns*  
*that*  
>> *are not in the GROUP BY or in an aggregate function. Obviously I cannot*  
*do*  
>> *that in SQL Server.*  
>>  
>> *Assume that my SELECT list was actually quite long and included columns*  
*from*  
>> *other tables but only tables that had 0 or 1 row for each person row.*  
> *What*  
>> *is the optimal way to write such a query for SQL Server? Should I use*  
>> *DISTINCT, GROUP BY the entire select list or put every column except*  
>> *personid in an aggregate function, like MAX? Or is there some other,*  
>> *preferrable method?*  
>>

> > *With a large select list and many rows I worry about the performance of  
> any  
> of the options I mentioned. Are my concerns warranted or does the  
> optimizer  
> know to ignore the DISTINCT or GROUP BY where constraints assure that it  
> is  
> unnecessary?  
>>  
>> Thanks in advance for your thoughts on this.  
>>  
>> Jim*

> > -----SAMPLE SCRIPT-----  
> > *CREATE TABLE person*  
> > (  
> > *personid int,*  
> > *firstname varchar(15),*  
> > *lastname varchar(15)*  
> > )  
> >  
> >  
> > *CREATE TABLE family*  
> > (  
> > *familyid int,*  
> > *familyname varchar(15),*  
> > *include bit*  
> > )  
> >  
> >  
> > *CREATE TABLE familymembership*  
> > (  
> > *personid int,*  
> > *familyid int*  
> > )  
> >  
> > *GO*  
> >  
> > *INSERT INTO person (personid, firstname, lastname) VALUES (1, 'Jim',  
> > 'Bo')*  
> > *INSERT INTO person (personid, firstname, lastname) VALUES (2, 'Bob',  
> > 'Smith')*  
> > *INSERT INTO person (personid, firstname, lastname) VALUES (3, 'Joe',  
> > 'Schmo')*  
> >  
> > *INSERT INTO familymembership VALUES (1, 1)*  
> > *INSERT INTO familymembership VALUES (2, 2)*  
> > *INSERT INTO familymembership VALUES (1, 3)*  
> >  
> > *INSERT INTO family (familyid, familyname, include) VALUES (1, 'Jim's  
> > Family', 1)*

microsoft.public.sqlserver.programming: Re: GROUP BY and performance

```
> > INSERT INTO family (familyid, familyname, include) VALUES (2, 'Bob''s
> > Family', 0)
> > INSERT INTO family (familyid, familyname, include) VALUES (3, 'Another
> > Family', 1)
> >
> >
>
>
```