

## Redmine - Defect #19544

### Malformed SQL query with SQLServer when grouping issues

2015-04-01 17:26 - Olivier Houdas

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jean-Philippe Lang	<b>% Done:</b>	0%
<b>Category:</b>	Issues	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	3.0.2	<b>Affected version:</b>	3.0.1
<b>Resolution:</b>	Fixed		
<b>Description</b>			
Environment: CentOS, Redmine 3.0.1, using Rails 4.2 and MS SQL database 2012 (through activerecord-sqlserver-adapter-4.2.4)			
On a project, filter issues and group them by category. When trying that on our test deployment of Redmine 3.0.1, I got an Error 500, with the following message: TinyTds::Error: A column has been specified more than once in the order by list. Columns in the order by list must be unique. SELECT [issues].[id] AS t0_r0 (...) ORDER BY issue_categories.name ASC, (CASE WHEN versions.effective_date IS NULL THEN 1 ELSE 0 END), versions.effective_date, versions.name, versions.id, enumerations.position DESC, issue_categories.name OFFSET 0 ROWS FETCH NEXT 100 ROWS ONLY			
It seems that group_by_sort_order from app/models/query.rb adds the first issue_categories.name ASC, but issue_categories.name is also contained in the order_by options when merged in app/models/issue_query.rb:			
<pre>def issues(options={})   order_option = [group_by_sort_order, options[:order]].flatten.reject(&amp;:blank?)</pre>			
which results in duplicated columns. It did not cause any error when using Rails 3.2, and Redmine 2.6.1.			
Can you confirm you reproduce? I fixed it by adding			
<pre>dup_index = order_option.rindex{ s  s[/#{group_by_sort_order.split(" ")[0]}/]}   if dup_index &gt; 0     order_option.delete_at(dup_index)   end</pre>			
in the issues() function after merging order options, but I'm not sure this is the right way to do it (maybe it should not be present in options[:order] in the first place?).			

#### Associated revisions

##### Revision 14165 - 2015-04-12 11:44 - Jean-Philippe Lang

SQL error with SQLServer when grouping+sorting issues (#19544).

##### Revision 14166 - 2015-04-12 14:34 - Jean-Philippe Lang

Fixed test failures (#19544).

##### Revision 14207 - 2015-04-25 09:37 - Jean-Philippe Lang

Merged r14165 and r14166 (#19544).

#### History

##### #1 - 2015-04-12 11:45 - Jean-Philippe Lang

- Subject changed from Malformed SQL query with MS SQL and Rails 4.2 when grouping issues to Malformed SQL query with SQLServer when grouping issues

- Status changed from New to Resolved

- Assignee set to Jean-Philippe Lang

- Target version set to 3.0.2

- Resolution set to Fixed

That should be fixed by [r14165](#).

#### #2 - 2015-04-13 10:17 - Olivier Houdas

I removed my changes and applied [r14165](#), and the filtering now works fine.

#### #3 - 2015-04-25 09:37 - Jean-Philippe Lang

- Status changed from Resolved to Closed

#### #4 - 2016-04-26 20:38 - Silvio Fernandes

- File `issue_query.rb` added

Hi guys,

I had the same problem of this issue, but my was with the fixed version column, in version 3.2.1 with SQL Server 2012, and this review 14207 did not solve my problem.

making a debug in the code, I found that the part where it should make a flatten between two Array, the variable `group_by_sort_order` comes as String, so I did a treatment to make this String into Array, and solved my problem.

code modified in `/app/models/issue_query.rb`

```
# Returns the issues
# Valid options are :order, :offset, :limit, :include, :conditions
def issues(options={})
  if group_by_sort_order.is_a?(String)
    # convert String in Array
    order_option = [group_by_sort_order.split(','), options[:order]].flatten.reject(&:blank?)
  else
    order_option = [group_by_sort_order, options[:order]].flatten.reject(&:blank?)
  end
end
```

...

#### #5 - 2016-04-27 06:14 - Toshi MARUYAMA

Silvio Fernandes wrote:

Hi guys,  
...

Please create new issue.

### Files

---

<code>issue_query.rb</code>	23.5 KB	2016-04-26	Silvio Fernandes
-----------------------------	---------	------------	------------------