

Redmine - Defect #15017

Search results show only the last page when using MSSQL

2013-10-01 13:22 - Olivier Houdas

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Search engine	Estimated time:	0.00 hour
Target version:		Affected version:	2.3.3
Resolution:			
Description			
<p>We have setup Redmine 2.3.3 on CentOS and MS SQL 2008, running on Rails 2.0.-p247 (but also tested with Rails 1.9.3). I have tested without any plugin. I created the DB from scratch with redmine MIGRATE commands. I created a new project, with one version and one category.</p> <p>Then,</p> <ol style="list-style-type: none">1. I added 15 bugs, all containing "mot clé" in the description.2. I entered "mot" in the search area (top right of redmine screen) <p>Results: I got a page listing 10 bugs from "Bug 11" down to "Bug 02".</p> <p>Expected results: start with "Bug 15" on top.</p> <p>If I look in debug logs, I can see the following requests:</p> <pre>[1m[35mSQL (58.5ms)[0m EXEC sp_executesql N'SELECT TOP (11) [issues].id FROM [issues] LEFT OUTER JOIN [projects] ON [projects].[id] = [issues].[project_id] LEFT OUTER JOIN [journals] ON [journals].[journalized_id] = [issues].[id] AND (journals.private_notes = 0 OR (projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND [journals].[journalized_type] = N''Issue'' WHERE (projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND (issues.project_id IN (1)) AND ((LOWER(subject) LIKE N''%mot%'') OR (LOWER(issues.description) LIKE N''%mot%'') OR (LOWER(journals.notes) LIKE N''%mot%'')) GROUP BY [issues].id ORDER BY MAX(issues.id)'</pre> <pre>[1m[36mSQL (20.5ms)[0m [1mEXEC sp_executesql N'SELECT [issues].[id] AS t0_r0, [issues].[tracker_id] AS t0_r1, [issues].[project_id] AS t0_r2, [issues].[subject] AS t0_r3, [issues].[description] AS t0_r4, [issues].[due_date] AS t0_r5, [issues].[category_id] AS t0_r6, [issues].[status_id] AS t0_r7, [issues].[assigned_to_id] AS t0_r8, [issues].[priority_id] AS t0_r9, [issues].[fixed_version_id] AS t0_r10, [issues].[author_id] AS t0_r11, [issues].[lock_version] AS t0_r12, [issues].[created_on] AS t0_r13, [issues].[updated_on] AS t0_r14, [issues].[start_date] AS t0_r15, [issues].[done_ratio] AS t0_r16, [issues].[estimated_hours] AS t0_r17, [issues].[parent_id] AS t0_r18, [issues].[root_id] AS t0_r19, [issues].[lft] AS t0_r20, [issues].[rgt] AS t0_r21, [issues].[is_private] AS t0_r22, [issues].[closed_on] AS t0_r23, [projects].[id] AS t1_r0, [projects].[name] AS t1_r1, [projects].[description] AS t1_r2, [projects].[homepage] AS t1_r3, [projects].[is_public] AS t1_r4, [projects].[parent_id] AS t1_r5, [projects].[created_on] AS t1_r6, [projects].[updated_on] AS t1_r7, [projects].[identifier] AS t1_r8, [projects].[status] AS t1_r9, [projects].[lft] AS t1_r10, [projects].[rgt] AS t1_r11, [projects].[inherit_members] AS t1_r12, [journals].[id] AS t2_r0, [journals].[journalized_id] AS t2_r1, [journals].[journalized_type] AS t2_r2, [journals].[user_id] AS t2_r3, [journals].[notes] AS t2_r4, [journals].[created_on] AS t2_r5, [journals].[private_notes] AS t2_r6 FROM [issues] LEFT OUTER JOIN [projects] ON [projects].[id] = [issues].[project_id] LEFT OUTER JOIN [journals] ON [journals].[journalized_id] = [issues].[id] AND (journals.private_notes = 0 OR (projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND [journals].[journalized_type] = N''Issue'' WHERE [issues].[id] IN (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11) AND (projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND (issues.project_id IN (1)) AND ((LOWER(subject) LIKE N''%mot%'') OR (LOWER(issues.description) LIKE N''%mot%'') OR (LOWER(journals.notes) LIKE N''%mot%'')) ORDER BY issues.id DESC'</pre>			

Whereas when I do it on a MYSQL database, I get the correct results, and the query to search for issues sorts by "issues.id DESC", correctly as expected (shown here on keyword "gps" with a database filled with several issues) :

```
[1m[36mSQL (20839.9ms)[0m  [1mSELECT DISTINCT `issues`.id FROM `issues` LEFT OUTER JOIN `projects`
ON `projects`.id = `issues`.project_id LEFT OUTER JOIN `journals` ON `journals`.journalized_
id = `issues`.id AND (journals.private_notes = 0 OR (projects.status <> 9 AND projects.id IN (S
ELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND `journals`.jour
nalized_type = 'Issue' WHERE (projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM
enabled_modules em WHERE em.name='issue_tracking')) AND (((LOWER(subject) LIKE '%gps%') OR (LOWER(
issues.description) LIKE '%gps%') OR (LOWER(journals.notes) LIKE '%gps%') OR issues.id IN (SELECT
customized_id FROM custom_values WHERE customized_type='Issue' AND customized_id=issues.id AND LOW
ER(value) LIKE '%gps%' AND custom_values.custom_field_id IN (2,10,12))))
ORDER BY issues.id DESC LIMIT 11
```

```
[1m[35mSQL (127.9ms)[0m  SELECT `issues`.id AS t0_r0, `issues`.tracker_id AS t0_r1, `issues`.
project_id AS t0_r2, `issues`.subject AS t0_r3, `issues`.description AS t0_r4, `issues`.due_
date AS t0_r5, `issues`.category_id AS t0_r6, `issues`.status_id AS t0_r7, `issues`.assigned_
to_id AS t0_r8, `issues`.priority_id AS t0_r9, `issues`.fixed_version_id AS t0_r10, `issues`
.author_id AS t0_r11, `issues`.lock_version AS t0_r12, `issues`.created_on AS t0_r13, `issue
s`.updated_on AS t0_r14, `issues`.start_date AS t0_r15, `issues`.done_ratio AS t0_r16, `issu
es`.estimated_hours AS t0_r17, `issues`.parent_id AS t0_r18, `issues`.root_id AS t0_r19, `is
sues`.lft AS t0_r20, `issues`.rgt AS t0_r21, `issues`.is_private AS t0_r22, `issues`.closed_
on AS t0_r23, `projects`.id AS t1_r0, `projects`.name AS t1_r1, `projects`.description AS
t1_r2, `projects`.homepage AS t1_r3, `projects`.is_public AS t1_r4, `projects`.parent_id AS
t1_r5, `projects`.created_on AS t1_r6, `projects`.updated_on AS t1_r7, `projects`.identifier
AS t1_r8, `projects`.status AS t1_r9, `projects`.lft AS t1_r10, `projects`.rgt AS t1_r11, `
projects`.inherit_members AS t1_r12, `journals`.id AS t2_r0, `journals`.journalized_id AS t2
_r1, `journals`.journalized_type AS t2_r2, `journals`.user_id AS t2_r3, `journals`.notes AS
t2_r4, `journals`.created_on AS t2_r5, `journals`.private_notes AS t2_r6 FROM `issues` LEFT OU
TER JOIN `projects` ON `projects`.id = `issues`.project_id LEFT OUTER JOIN `journals` ON `jour
nals`.journalized_id = `issues`.id AND (journals.private_notes = 0 OR (projects.status <> 9 AN
D projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) A
ND `journals`.journalized_type = 'Issue' WHERE `issues`.id IN (19084, 18950, 18709, 18691, 184
66, 18358, 18304, 18255, 18141, 18140, 18086) AND (projects.status <> 9 AND projects.id IN (SELECT
em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND (((LOWER(subject) LIKE
'%gps%') OR (LOWER(issues.description) LIKE '%gps%') OR (LOWER(journals.notes) LIKE '%gps%') OR i
ssues.id IN (SELECT customized_id FROM custom_values WHERE customized_type='Issue' AND customized_
id=issues.id AND LOWER(value) LIKE '%gps%' AND custom_values.custom_field_id IN (2,10,12)))) ORDER
BY issues.id DESC
```

History

#1 - 2013-10-09 14:54 - Olivier Houdas

The bug seems to happen in Ruby ActiveRecord SQL adapter 3.2.12:
in

```
activerecord-sqlserver-adapter-3.2.12/lib/arel/visitors/sqlserver.rb
```

in

```
def visit_Arel_Nodes_SelectStatementWithoutOffset(o, windowed=false)
```

lines 149 to 152:

```
orders = orders.map do |x|
  expr = Arel.sql projection_without_expression(x.expr)
  x.descending? ? Arel::Nodes::Max.new([expr]) : Arel::Nodes::Min.new([expr])
end
```

#2 - 2014-02-04 16:13 - Olivier Houdas

- Status changed from New to Resolved

For those interested, I tried a fix in the Ruby On Rails SQL server adapter:
<https://github.com/rails-sqlserver/activerecord-sqlserver-adapter/issues/286>

In activerecord-sqlserver-adapter-3.2.12/lib/arel/visitors/sqlserver.rb, I changed the line 151:

```
x.descending? ? Arel::Nodes::Max.new([expr]) : Arel::Nodes::Min.new([expr])
```

to

```
if x.descending?  
  Arel::Nodes::Max.new([expr])  
  groups += [expr]  
  Arel::Nodes::Descending.new([expr])  
else  
  Arel::Nodes::Min.new([expr])  
end
```

This fix has been working without showing visible bugs over the last 3 months. We have 20 active users, over 20000 issues/tasks in our database, and we use Gantt and Roadmaps as well as of course Filters and the global search.

#3 - 2015-10-21 16:23 - Olivier Houdas

Note that this is fixed with Redmine 3.x, which uses Rails 4.2, which includes an ActiveRecord SQL adapter which changed that part of the code, and does not have the bug anymore.

This issue should be closed.

#4 - 2015-10-21 19:20 - Jean-Philippe Lang

- *Status changed from Resolved to Closed*

Thanks for reporting that.