

Redmine - Defect #29581

Issues in paginated views may be lost because sorting criteria are not unique

2018-09-10 07:37 - Mizuki ISHIKAWA

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Issues list	Estimated time:	0.00 hour
Target version:	4.1.0	Affected version:	
Resolution:	Fixed		
Description			
<p>This is a problem that the issue that should be present is not displayed on the issues/index. The same problem may exist even at index other than issues/index.</p> <p>This problem occurs when using PostgreSQL. If you use only non-unique fields such as trackers or categories as the sorting criteria, there is a possibility that some issues will not be displayed on pagination views.</p> <p>This phenomenon is described in the document of PostgreSQL. (As a specification rather than a bug) https://www.postgresql.org/docs/8.3/static/queries-limit.html</p> <p>Questions about the same problem: https://stackoverflow.com/questions/13580826/postgresql-repeating-rows-from-limit-offset</p> <p>I attached file is the test I wrote to reproduce this problem. If you are using PostgreSQL, that test will fail. (Non-paginated issue ids and paginated issue ids should be the same.)</p> <p>Failure:</p> <pre>--- expected +++ actual @@ -1,13 @@ -[2, 12, 11, 8, 7, 5, 3, 1, 13] # Non-paginated issue ids +[11, 12, 12, 7, 7, 5, 3, 1, 13] # Paginated issue ids</pre>			
Related issues:			
Related to Redmine - Defect #32737: Duplicate sort keys for issue query cause...			Closed
Has duplicate Redmine - Defect #31924: Paging misses some entries			Closed

Associated revisions

Revision 18263 - 2019-06-20 08:13 - Jean-Philippe Lang

Issues in paginated views may be lost because sorting criteria are not unique (#29581).

Patch by Mizuki ISHIKAWA.

History

#1 - 2018-09-20 10:22 - Mizuki ISHIKAWA

- File fix-29581.patch added

I wrote a patch to solve this problem.

I fixed to add unique fields (ex: issues.id, time_entries.id) as sort criteria.

#2 - 2018-10-01 11:09 - Go MAEDA

- Target version set to 4.1.0

Setting target version to 4.1.0.

#3 - 2019-06-17 07:30 - vzvu 3k6k

```
order_option += ['issues.id ASC'] unless order_option.include?("issues.id DESC") || order_option.include?("issues.id ASC")
```

Is it ok to use `issues.id ASC` as a default implicit order?

IssueQuery#default_sort_criteria uses issues.id DESC.

(Pair-reviewed with [maimai77](#))

#4 - 2019-06-18 07:25 - Mizuki ISHIKAWA

- File fix-29581-v2.patch added

vzvu 3k6k wrote:

```
order_option += ['issues.id ASC'] unless order_option.include?("issues.id DESC") || order_option.include?("issues.id ASC")
```

Is it ok to use `issues.id ASC` as a default implicit order?

IssueQuery#default_sort_criteria uses issues.id DESC.

(Pair-reviewed with [maimai77](#))

Thank you for reviewing fix-29581.patch.

As you point out, it seems natural to use the same sort criteria as IssueQuery#default_sort_criteria.
I have attached the file to fixed patch.

#5 - 2019-06-18 14:06 - vzvu 3k6k

Thank you for your response! Your v2 patch looks good to me.

#6 - 2019-06-18 17:51 - Seiei Miyagi

```
'issues.id DESC'
```

Is it OK to write table name of the Issue model directly?

In app/models/issue_query.rb, It seems code like following is more preferable.

```
"#{Issue.table_name}.id DESC"
```

#7 - 2019-06-19 02:19 - Mizuki ISHIKAWA

- File fix-29581-v3.patch added

Seiei Miyagi wrote:

```
'issues.id DESC'
```

Is it OK to write table name of the Issue model directly?

In app/models/issue_query.rb, It seems code like following is more preferable.

[...]

Thank you for pointing it out.

I changed the way of writing table names.

#8 - 2019-06-20 08:13 - Jean-Philippe Lang

- Status changed from New to Closed

- Assignee set to Jean-Philippe Lang

- Resolution set to Fixed

Patch committed, thanks.

#9 - 2019-08-30 15:46 - Go MAEDA

- Has duplicate Defect #31924: Paging misses some entries added

#10 - 2020-04-02 08:46 - Go MAEDA

- Related to Defect #32737: Duplicate sort keys for issue query cause SQL error with SQL Server added

Files

add-test-that-fails-when-using-postgresql.patch	877 Bytes	2018-09-10	Mizuki ISHIKAWA
fix-29581.patch	3.36 KB	2018-09-20	Mizuki ISHIKAWA
fix-29581-v2.patch	3.55 KB	2019-06-18	Mizuki ISHIKAWA
fix-29581-v3.patch	3.67 KB	2019-06-19	Mizuki ISHIKAWA