

Redmine - Defect #31287

Ordering wiki pages should not be case sensitive

2019-05-01 22:02 - hyper loop

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Wiki	Estimated time:	0.00 hour
Target version:	4.2.0	Affected version:	
Resolution:	Fixed		
Description			
<p>There is a tricky problems for the ordering of WikiPages under Postgresql, the title is designed to be case insensitive.</p> <p>https://github.com/redmine/redmine/blob/24ccf9d981d907ec7c8fa3901d0a79bc7f641825/app/models/wiki_page.rb#L49</p> <p>But when it comes to order the wiki pages, it's ordered in a case sensitive way, which results in the three WikiPages titled with ["Aac", "Aace_", "ABc", "Acc"] is ordered as ["ABc", "Aac", "Aace_", "Acc"]`, which is not correct since `Aac`, "Aace_" should precede `ABc`.</p> <p>A quick fix would be change the</p> <pre>order(:title)</pre> <p>to</p> <pre>order('lower(title)')</pre> <p>, patch is attached.</p> <p>The order works well under mysql since mysql will order regardless of the case. But I think it should also work properly under Postgresql.</p>			

Associated revisions

Revision 19831 - 2020-06-20 08:50 - Go MAEDA

Ordering wiki pages should not be case sensitive (#31287).

Patch by Yuichi HARADA.

History

#1 - 2019-05-01 22:03 - hyper loop

- File patch.diff added

#2 - 2019-05-01 22:14 - hyper loop

- File patch2.diff added

Similar for

```
has_many :pages, lambda {order('title')}, :class_name => 'WikiPage', :dependent => :destroy
```

<https://github.com/redmine/redmine/blob/24ccf9d981d907ec7c8fa3901d0a79bc7f641825/app/models/wiki.rb#L23>, which should be

```
has_many :pages, lambda {order('lower(title)')}, :class_name => 'WikiPage', :dependent => :destroy
```

patch is attached too.

#3 - 2019-05-12 01:43 - Go MAEDA

- Target version set to Candidate for next major release

#4 - 2019-06-14 02:24 - Yuichi HARADA

A warning occurred when I created a test and applied [patch2.diff](#) to the trunk.

```
diff --git a/test/unit/wiki_test.rb b/test/unit/wiki_test.rb
index b4ecd29bb..9184d4efc 100644
--- a/test/unit/wiki_test.rb
+++ b/test/unit/wiki_test.rb
@@ -52,6 +52,16 @@ class WikiTest < ActiveSupport::TestCase
   assert_equal page, wiki.find_page('ANOTHER page')
 end

+ def test_ordering_pages_should_not_be_case_sensitive
+   wiki = Wiki.find(1)
+   wiki.pages.destroy_all
+   %w(Acc ABc Aace_ Aac).each do |title|
+     wiki.pages.create(:title => title)
+   end
+   wiki.reload
+   assert_equal %w(Aac Aace_ ABc Acc), wiki.pages.pluck(:title)
+ end
+
+ def test_find_page_with_cyrillic_characters
+   wiki = Wiki.find(1)
+   page = WikiPage.find(10)
```

```
$ bundle exec rake test TEST=test/unit/wiki_test.rb:56
Run options: --seed 43297
```

Running:

```
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (called from block in
  <class:Wiki> at ./app/models/wiki.rb:23)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (called from block in
  <class:Wiki> at ./app/models/wiki.rb:23)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (
  called from test_ordering_pages_should_not_be_case_sensitive at ./test/unit/wiki_test.rb:57)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (
  called from initialize at ./app/models/wiki_page.rb:74)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (
  called from initialize at ./app/models/wiki_page.rb:74)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (
  called from initialize at ./app/models/wiki_page.rb:74)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (
  called from initialize at ./app/models/wiki_page.rb:74)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (called from block in
  <class:Wiki> at ./app/models/wiki.rb:23)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
  called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
  Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
  ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (called from block in
```

```
<class:Wiki> at ./app/models/wiki.rb:23)
DEPRECATION WARNING: Dangerous query method (method whose arguments are used as raw SQL)
called with non-attribute argument(s): "LOWER(title)". Non-attribute arguments will be disallowed in
Rails 6.0. This method should not be called with user-provided values, such as request parameters or model att
ributes. Known-safe values can be passed by wrapping them in Arel.sql(). (
called from test_ordering_pages_should_not_be_case_sensitive at ./test/unit/wiki_test.rb:62)
.

Finished in 0.137138s, 7.2919 runs/s, 7.2919 assertions/s.
1 runs, 1 assertions, 0 failures, 0 errors, 0 skips
$
```

I made a patch based on [patch.diff](#) and [patch2.diff](#) .

#5 - 2019-06-27 02:52 - Go MAEDA

- Category set to Wiki
- Target version changed from Candidate for next major release to 4.1.0

Setting the target version to 4.1.0.

#6 - 2019-09-08 17:53 - Anonymous

+1, but will commit happen? Why are we waiting?

#7 - 2019-09-28 00:33 - Go MAEDA

- Target version changed from 4.1.0 to 4.2.0

#8 - 2020-06-18 01:41 - Go MAEDA

- File *sqlite.png* added
- Subject changed from Case sensitivity for title of WikiPage under postgresql to Ordering wiki pages should not be case sensitive

Not only PostgreSQL but SQLite is also affected.

Index by title

- [ABC](#)
- [ANOTHER](#)
- [Aaa](#)
- [Another page](#)

#9 - 2020-06-20 08:50 - Go MAEDA

- Status changed from New to Closed
- Assignee set to Go MAEDA
- Resolution set to Fixed

Committed the patch. Thank you.

Files

patch.rb	566 Bytes	2019-05-01	hyper loop
patch.diff	566 Bytes	2019-05-01	hyper loop
patch2.diff	520 Bytes	2019-05-01	hyper loop
fixed-31287.patch	4.15 KB	2019-06-14	Yuichi HARADA
sqlite.png	18 KB	2020-06-17	Go MAEDA