There is a tricky problems for the ordering of WikiPages under Postgresql, the title is designed to be case insensitive.

https://github.com/redmine/redmine/blob/24ccf9d981d907ec7c8fa3901d0a79bc7f641825/app/models/wiki_page.rb#L49

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".

A quick fix would be change the order(:title) to order('lower(title)') , patch is attached.

The order works well under mysql since mysql will order regardless of the case. But I think it should also work properly under Postgresql.

---

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
A warning occurred when I created a test and applied attachment:patch2.diff to the trunk.

```diff
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 @@
     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 ABc ACc).each do |title|
+      wiki.pages.create(title => title)
+    end
+    wiki.reload
+    assert_equal %w(Aac Aace ABc Acc AAc), 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
```

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 attributes. 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 attributes. 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 attributes. 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 attributes. 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 attributes. Known-safe values can be passed by wrapping them in Arel.sql(). (called from initialize 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 attributes. 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 attributes. 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 attributes. 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 attributes. 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 attributes. 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 attributes. 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 attributes. 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 attributes. 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 attachment:patch.diff and attachment: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.
Committed the patch. Thank you.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>patch.rb</td>
<td>566 Bytes</td>
<td>2019-05-01</td>
<td>hyper loop</td>
</tr>
<tr>
<td>patch.diff</td>
<td>566 Bytes</td>
<td>2019-05-01</td>
<td>hyper loop</td>
</tr>
<tr>
<td>patch2.diff</td>
<td>520 Bytes</td>
<td>2019-05-01</td>
<td>hyper loop</td>
</tr>
<tr>
<td>fixed-31287.patch</td>
<td>4.15 KB</td>
<td>2019-06-14</td>
<td>Yuichi HARADA</td>
</tr>
<tr>
<td>sqlite.png</td>
<td>18 KB</td>
<td>2020-06-17</td>
<td>Go MAEDA</td>
</tr>
</tbody>
</table>