Redmine - Defect #5726

Cannot sort issues on custom fields

2010-06-22 18:25 - Cyber Sprocket

Status:	Closed	**Start date:**	2010-06-22
Priority:	Normal	**Due date:**	
Assignee:	Muntek Singh	**% Done:**	100%
Category:	Custom fields	**Estimated time:**	0.00 hour
Target version:			
Resolution:	Fixed	**Affected version:**	0.9.4

Description

We added a new custom field "Rank":

- Integer
- Length 0-3
- no regular expression
- All projects
- non-filter

We can edit this, but we cannot sort by it on the issues output. Also any queries that do anything with this column (group, sort) break.

The issues report simply comes up blank.

Possible cause - pre-existing records have a null value in that column? Just guessing here.

This makes it difficult to coordinate with one of our primary clients that requires a Priority + Rank system to refine their "what we need done first" list.

Related issues:

Related to Redmine - Defect #5778: Defect in the filters (custom field) from ...	**Closed**	**2010-06-28**
Related to Redmine - Defect #2755: Failing test after r2257 with MySQL 4.1.20...	**Closed**	**2009-02-15**

History

#1 - 2010-06-22 23:01 - Yuki Kita

This seems to be a duplicate of #1139 / #3488.
You can do it if you check "Used as a filter" of the custom field and select it in "Administration/Settings/Issue tracking".

#2 - 2010-06-22 23:21 - Cyber Sprocket

This does not help. Tried that.

I still cannot sort by the rank field. Also tried putting a value in that field for all open tickets.

#3 - 2010-07-06 22:28 - Cyber Sprocket

Still happens in Redmine 0.9.5.stable.3824 (MySQL).

This is definitely related to sort in custom filters. As soon as I remove the rank (integer) from the sort the issue filter works fine.

#4 - 2010-07-06 23:18 - Yuki Kita

I post the patch to fix #5778, and I assume it also fixes this issue.
Cyber,
can you apply the patch (attatched in #5778) and report the result?

#5 - 2010-07-08 03:21 - Cyber Sprocket

Yuki - Is there a quick way to apply patches, or do I have to hand-edit the code?

#6 - 2010-07-08 04:56 - Yuki Kita

Cyber Sprocket wrote:

Yuki - Is there a quick way to apply patches, or do I have to hand-edit the code?

Here is how to apply the patch.

```
$ cd your-redmine-root
$ patch -p0 < custom_field_filter.patch
patching file app/models/query.rb
patching file app/views/queries/_filters.rhtml
```

#7 - 2010-07-08 06:53 - Cyber Sprocket

It doesn't crash now, but it does not return results, the table is now blank.

#8 - 2010-07-08 06:56 - Cyber Sprocket

BTW, this happens even without custom queries. I changed the settings to show the Rank field on the default issue output. It shows up as the last column with the proper data intact. As soon as I click on the column heading to sort, the page returns but shows up blank.

You can see the effect (with the new patch in place) by going to our public portion of the site:
http://redmine.cybersprocket.com/projects/wpcafepress/issues

Click on "Rank" on the far right & it will now return the page "wireframe" but no issues table results:
http://redmine.cybersprocket.com/projects/wpcafepress/issues?sort=cf_4%2Cid%3Adesc

#9 - 2010-07-08 07:26 - Cyber Sprocket

I found the problem:

custom_field.rb uses a cast to decimal. This is not supported in MySQL v4.1 which is what we are running on our server. MySQL v4.1 is stated as the minimum supported version in the install docs:
http://www.redmine.org/wiki/redmine/RedmineInstall

custom_field.rb

```
      when 'int', 'float'
        # Make the database cast values into numeric
        # Postgresql will raise an error if a value can not be casted!
        # CustomValue validations should ensure that it doesn't occur
        "(SELECT CAST(cv_sort.value AS decimal(60,3)) FROM #{CustomValue.table_name} cv_sort" +
          " WHERE cv_sort.customized_type='#{self.class.customized_class.name}'" +
          " AND cv_sort.customized_id=#{self.class.customized_class.table_name}.id" +
          " AND cv_sort.custom_field_id=#{id} AND cv_sort.value <> '' AND cv_sort.value IS NOT NULL LIMIT 1)"
```

#10 - 2010-07-08 07:39 - Cyber Sprocket

This fixes the problem for us... but you still have an issue of what to do with FLOAT on MySQL 4.1 since cast to decimal is not an option.

```
  # Returns a ORDER BY clause that can used to sort customized
  # objects by their value of the custom field.
  # Returns false, if the custom field can not be used for sorting.
  def order_statement
    case field_format
      when 'string', 'text', 'list', 'date', 'bool'
        # COALESCE is here to make sure that blank and NULL values are sorted equally
        "COALESCE((SELECT cv_sort.value FROM #{CustomValue.table_name} cv_sort" +
          " WHERE cv_sort.customized_type='#{self.class.customized_class.name}'" +
          " AND cv_sort.customized_id=#{self.class.customized_class.table_name}.id" +
          " AND cv_sort.custom_field_id=#{id} LIMIT 1), '')"
      when 'int'
        # CyberSprocket.com fix...
        # Make the database cast values into numeric
        # Postgresql will raise an error if a value can not be casted!
        # CustomValue validations should ensure that it doesn't occur
        "(SELECT CAST(cv_sort.value AS signed) FROM #{CustomValue.table_name} cv_sort" +
          " WHERE cv_sort.customized_type='#{self.class.customized_class.name}'" +
          " AND cv_sort.customized_id=#{self.class.customized_class.table_name}.id" +
          " AND cv_sort.custom_field_id=#{id} AND cv_sort.value <> '' AND cv_sort.value IS NOT NULL LIMIT 1)"
      when 'float'
        # Make the database cast values into numeric
        # Postgresql will raise an error if a value can not be casted!
        # CustomValue validations should ensure that it doesn't occur
        "(SELECT CAST(cv_sort.value AS decimal(60,3)) FROM #{CustomValue.table_name} cv_sort" +
          " WHERE cv_sort.customized_type='#{self.class.customized_class.name}'" +
```

```
          " AND cv_sort.customized_id=#{self.class.customized_class.table_name}.id" +
          " AND cv_sort.custom_field_id=#{id} AND cv_sort.value <> '' AND cv_sort.value IS NOT NULL LIMIT 1)"
      else
        nil
    end
  end
```

#11 - 2010-07-08 21:28 - Mischa The Evil

Cyber Sprocket wrote:

> [...]
>
> MySQL v4.1 is stated as the minimum supported version in the install docs:
> http://www.redmine.org/wiki/redmine/RedmineInstall

That info is *not entirely* true. See e.g. issue #2755. Since Redmine 0.9.0 MySQL 5 is required to fully be compatible with the Redmine Core.

#12 - 2010-07-08 22:46 - Cyber Sprocket

Ok - well I guess that needs to be updated on the RedmineInstall page & a more prominent notice should go out to people when they upgrade.

We started on 0.8X and just went through the normal upgrade process. Never did id complain nor warn about requiring MySQL 5.x. Maybe even put a "we recommend MySQL 5.X, you are on 4.X" on the admin/info screen.

Also, I think the patch I proposed should be considered. At the very least it gets INT custom fields working on MySQL 4 installs and I can't see how it would hurt anyone on MySQL 5.X.

#13 - 2010-07-09 05:41 - Yuki Kita

- Assignee set to Muntek Singh

> Ok - well I guess that needs to be updated on the RedmineInstall page & a more prominent notice should go out to people when they upgrade.

Muntek,
Can you update RedmineInstall page so that users understand current Redmine does not support MySQL4.1?

#14 - 2010-07-09 05:55 - Muntek Singh

- Status changed from New to Closed

- % Done changed from 0 to 100

- Resolution set to Fixed

Done

#15 - 2010-07-11 19:21 - Jean-Baptiste Barth

JPL said he didn't want to drop Mysql 4.1 support in #2755 1 year ago, I don't know if it's still the case. I personally think people should upgrade now, Mysql 4 is far too old and not supported at all since 12/2009.

#16 - 2010-07-11 19:28 - Cyber Sprocket

The patch provided will at least allow int support on custom fields on 4.X.

As for the MySQL upgrade to 5.X, that is great in theory however many sites including our own makes that a big undertaking. We have a half-dozen clients on the server with some using systems built on MySQL 4.x.

Side-by-side installs of MySQL can be problematic. Upgrading all of our clients will cost us time, which for us is money. Our clients won't pay for the upgrade. To them it is not broken (their apps work) so why fix it. The simple argument of "MySQL 4 is too old" sounds good on the surface, but out in the real world the issue is not so cut & dried. We too agree that people should be on supported software like MySQL 5 and PHP5, but alas that is not something we can force on our clients... and in our business our clients are our livelihood.