Redmine - Defect #32546

Issue relations filter lacks "is not"

2019-12-02 10:53 - Alexander Achenbach

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Issues filter	**Estimated time:**	0.00 hour
Target version:	4.1.0		
Resolution:	Fixed	**Affected version:**	4.0.5

Description

It seems the "is not" operator was omitted in the :relation definition of query operators.
The actual implementation seems to have it, though.
The attached patch adds the operator to the list (applied to Redmine 4.0.5).

Related issues:

Related to Redmine - Feature #3265: Filter on issue relations	**Closed**	**2009-04-29**

Associated revisions

Revision 19363 - 2019-12-12 01:56 - Go MAEDA

Issue relations filter lacks "is not" (#32546).

Patch by Alexander Achenbach.

Revision 19364 - 2019-12-12 02:00 - Go MAEDA

Test for r19363 (#32546).

Patch by Go MAEDA.

Revision 19369 - 2019-12-13 01:35 - Go MAEDA

Merged r19363 from trunk to 4.1-stable (#32546).

Revision 19370 - 2019-12-13 01:36 - Go MAEDA

Merged r19364 from trunk to 4.1-stable (#32546).

History

#1 - 2019-12-03 15:07 - Go MAEDA

- Target version set to Candidate for next major release

Alexander Achenbach wrote:

> The actual implementation seems to have it, though.

The implementation for "is not" operator is at source:tags/4.0.5/app/models/issue_query.rb#L568. I think "!" operator was not included by mistake and this should be fixed.

```
      when "=", "!" 
        op = (operator == "=" ? 'IN' : 'NOT IN')
        "#{Issue.table_name}.id #{op} (SELECT DISTINCT #{IssueRelation.table_name}.#{join_column} FROM #{
IssueRelation.table_name} WHERE #{IssueRelation.table_name}.relation_type = '#{self.class.connection.
quote_string(relation_type)}' AND #{IssueRelation.table_name}.#{target_join_column} = #{value.first.to_i})" 
```

#2 - 2019-12-03 15:10 - Go MAEDA

- Related to Feature #3265: Filter on issue relations added

#3 - 2019-12-03 15:10 - Go MAEDA

- Target version changed from Candidate for next major release to Candidate for next minor release

#4 - 2019-12-09 04:10 - Go MAEDA

Go MAEDA wrote:

> The implementation for "is not" operator is at source:tags/4.0.5/app/models/issue_query.rb#L568. I think "!" operator was not included by mistake and this should be fixed.

I started to wonder whether the "is not" filter is practical.

The reason is that the "is not" filter returns too many results. In many cases, it returns almost all issues in the project.

Suppose that there are 1000 issues in the project, and issue 2 and issue 3 have a "related to" relationship with each other. if you apply the filter "Related to" "is not" "3", it returns 999 issues other than issue 2.

I think the result is not so useful.

#5 - 2019-12-09 06:01 - Go MAEDA

Go MAEDA wrote:

> I started to wonder whether the "is not" filter is practical.
>
> The reason is that the "is not" filter returns too many results. In many cases, it returns almost all issues in the project.

Sorry, I have changed my mind. Please forget what I wrote in #32546#note-4.

Although I wrote "is not" filter is not so useful, it is wrong. It is useful when it is used with the combination with other filters. For example:

- "Subject" "contains" "awesome feature"
- "Is duplicate of" "is not" "10"

Suppose that issue 10 is an issue to implement an awesome feature. There are lot of duplicate issues and you want to create "Is duplicte of" relation ship for all those issues. You can find issues without "Is duplicate of" relationship by the above filter.

#6 - 2019-12-09 07:52 - Go MAEDA

Here is a test for the patch.

```
diff --git a/test/unit/query_test.rb b/test/unit/query_test.rb
index df6c61551..56ec82d14 100644
--- a/test/unit/query_test.rb
+++ b/test/unit/query_test.rb
@@ -1153,6 +1153,10 @@ class QueryTest < ActiveSupport::TestCase
     query = IssueQuery.new(:name => '_')
     query.filters = {"relates" => {:operator => '=', :values => ['2']}}
     assert_equal [1], find_issues_with_query(query).map(&:id).sort
+
+    query = IssueQuery.new(:name => '_')
+    query.filters = {"relates" => {:operator => '!', :values => ['1']}}
+    assert_equal Issue.where.not(:id => [2, 3]).order(:id).ids, find_issues_with_query(query).map(&:id).sort
   end

   def test_filter_on_relations_with_any_issues_in_a_project
```

#7 - 2019-12-09 08:42 - Alexander Achenbach

Go MAEDA wrote:

> It is useful when it is used with the combination with other filters. [...]

Admittedly we have not had a need for this in years of using Redmine, but projects have become somewhat more complex, and now our use case is a pending re-factorization of related issues, so we have to look for issues "related to A, but not related to B" prior to updating their relations.

Thanks for the test.

#8 - 2019-12-10 08:52 - Go MAEDA

- Target version changed from Candidate for next minor release to 4.2.0

Setting the target version to 4.2.0.

#9 - 2019-12-12 03:30 - Go MAEDA

- Status changed from New to Resolved

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the patch. Thank you for reporting and fixing the issue.

#10 - 2019-12-12 03:34 - Go MAEDA

- Target version changed from 4.2.0 to 4.1.0

I propose to merge this small fix into 4.1-stable.

#11 - 2019-12-13 01:36 - Go MAEDA

- Status changed from Resolved to Closed

Files

fix-relation-query-ops.patch	403 Bytes	2019-12-02	Alexander Achenbach