redmineorg-copy202205 - Vote #81906

Chained custom field filter doesn't work for User fields

2022/05/09 19:23 - Admin Redmine

:	Confirmed	:		2022/05/09
:		:		
:		:		0%
:	Filters_59	:		0.00
:	Candidate for next minor release_33	:		0.00
Redmineorg_URL:	https://www.redmine.org/issues/36940	**status_id:**		9
category_id:	59	**tracker_id:**		1
version_id:	33	**plus1:**		0
issue_org_id:	36940	**affected_version:**		
author_id:	291	**closed_on:**		
assigned_to_id:	0	**affected_version_id:**		173
comments:	2			

A chained custom field filter doesn't work for fields with format <u>User</u>.

Steps to reproduce:

Create a *user* custom field *A* (the format doesn't matter) and check <u>Used as a filter</u>

Create an *issue* custom field *B* with format *User* and check <u>Used as a filter</u>

On the <u>My account</u> page set the custom value *A* for your own account to an arbitrary value

Create an issue and select <> for the custom field *B*

Filter the issue list by the filter *B's A* and use the value you set on the <u>My account</u> page

Expected result:

- The issue is found

Actual result:

- The issue is not found

The reason is that the SQL that filters the issues contains a @WHERE customized_type='User'@. However, the correct clause would be @WHERE customized_type='Principal'@.

Here is a patch to fix the issue:

```
--- a/app/models/query.rb
+++ b/app/models/query.rb
@@ -1166,7 +1166,7 @@ class Query < ActiveRecord::Base
         end

         filter = available_filters[field]
```

```
-        target_class  =  filter[:through].format.target_class
+        target_class  =  filter[:through].format.target_class.base_class

      "#{queried_table_name}.id  #{not_in}  IN  ("  +
         "SELECT  customized_id  FROM  #{CustomValue.table_name}"  +
```

Best regards,
Thomas

journals

#1 - 2022/05/10 16:57 - Admin Redmine

-            *Filters_59*

-                  *Candidate for next minor release_33*