Redmine - Defect #33912

Sorting of multiple columns including Date column does not work

2020-08-27 07:23 - Yuichi HARADA

Status:	Confirmed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Time tracking	**Estimated time:**	0.00 hour
Target version:	Candidate for next major release		
Resolution:		**Affected version:**	

Description

When specifying the sorting condition of multiple columns including Date column in the custom query, sorting of columns after Date column does not work.

- Defining a custom query
  customquery.png

- List of spent time with custom query
  spenttime-with-customquery.png

History

#1 - 2020-09-02 07:30 - Yuichi HARADA

- File fixed-33912.patch added

I think the following patch will fix the problem.

```
diff --git a/app/models/time_entry_query.rb b/app/models/time_entry_query.rb
index c8f8eeaeb..59a4f25ca 100644
--- a/app/models/time_entry_query.rb
+++ b/app/models/time_entry_query.rb
@@ -23,7 +23,7 @@ class TimeEntryQuery < Query

   self.available_columns = [
     QueryColumn.new(:project, :sortable => "#{Project.table_name}.name", :groupable => true),
-    QueryColumn.new(:spent_on, :sortable => ["#{TimeEntry.table_name}.spent_on", "#{TimeEntry.table_name}.cre
ated_on"], :default_order => 'desc', :groupable => true),
+    QueryColumn.new(:spent_on, :sortable => "#{TimeEntry.table_name}.spent_on", :default_order => 'desc', :gr
oupable => true),
     TimestampQueryColumn.new(:created_on, :sortable => "#{TimeEntry.table_name}.created_on", :default_order =
> 'desc', :groupable => true),
     QueryColumn.new(:tweek, :sortable => ["#{TimeEntry.table_name}.spent_on", "#{TimeEntry.table_name}.create
d_on"], :caption => :label_week),
     QueryColumn.new(:author, :sortable => lambda {User.fields_for_order_statement}),
@@ -152,7 +152,12 @@ class TimeEntryQuery < Query
   def results_scope(options={})
     order_option = [group_by_sort_order, (options[:order] || sort_clause)].flatten.reject(&:blank?)

-    order_option << "#{TimeEntry.table_name}.id ASC"
+    order_option <<
+      if order_option.reverse.find{|order| order.match?(/#{TimeEntry.table_name}\./)}&.match?(/\sDESC$/i)
+        "#{TimeEntry.table_name}.id DESC"
+      else
+        "#{TimeEntry.table_name}.id ASC"
+      end
     base_scope.
       order(order_option).
       joins(joins_for_order_statement(order_option.join(',')))
```

#2 - 2020-11-17 16:04 - Go MAEDA

- Status changed from New to Confirmed

#3 - 2022-07-04 19:40 - Felix Schäfer

We (Plan.io) can confirm this issue and concur with the first part of the proposed patch. In addition the QueryColumn should be changed to a

TimestampQueryColumn.

I am not sure about the second part of the patch though. Currently the IssueQuery defaults to id DESC if no other id sorting is defined regardless of other sortings. If this is changed in TimeEntryQuery it should also be consistently change in the IssueQuery.

Go Maeda, what would be needed to commit this patch? Do you think a test would be necessary?

#4 - 2022-07-06 17:21 - Go MAEDA

Felix Schäfer wrote:

> I am not sure about the second part of the patch though. Currently the IssueQuery defaults to id DESC if no other id sorting is defined regardless of other sortings. If this is changed in TimeEntryQuery it should also be consistently change in the IssueQuery.

Without the second part, the sorting order is reversed from the current and an existing test TimelogControllerTest#test_index_should_sort_by_spent_on_and_created_on will fail.

> Go Maeda, what would be needed to commit this patch? Do you think a test would be necessary?

I can commit the patch if how handle the second part of the patch is decided.

#5 - 2022-07-22 12:22 - Go MAEDA

- Target version set to Candidate for next major release

Files

customquery.png	115 KB	2020-08-27	Yuichi HARADA
spenttime-with-customquery.png	155 KB	2020-08-27	Yuichi HARADA
fixed-33912.patch	3.17 KB	2020-09-02	Yuichi HARADA