

Redmine - Feature #15861

Issues, Filter 'number of parents'

2014-01-13 20:41 - Jérôme BATAILLE

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:			
Resolution:			
Description			
The aim is to be able to filter on a sub-tree, in projects with a deep tree.			

History

#1 - 2014-01-13 20:52 - Jérôme BATAILLE

Only 4 things have to be done :

```
# 1) Add this method in the Issue query model
def sql_for_parent_count_field(field, operator, value)
  # db_table empty
  db_field = "( #{Issue.table_name}.#{Issue.left_column_name} - (SELECT #{Issue.left_column_name} FROM #{
Issue.table_name} AS issues_root WHERE issues_root.id = #{Issue.table_name}
.root_id AND issues_root.parent_id IS NULL) )"
  sql_for_field(field, operator, value, '', db_field)
end

# 2) Enhance the sql_for_field method (C.f.: ENHANCEMENT HERE)
#
# Helper method to generate the WHERE sql for a +field+, +operator+ and a +value+
def sql_for_field(field, operator, value, db_table, db_field, is_custom_filter=false)
  sql = ''
  case operator
  when "="
    if value.any?
      case type_for(field)
      when :date, :date_past
        sql = date_clause(db_table, db_field, (Date.parse(value.first) rescue nil), (Date.parse(
value.first) rescue nil))
      when :integer
        # ENHANCEMENT HERE : manage case where db_table param is not provided
        db_table_and_field = (db_table.present? ? "#{db_table}.#{db_field}" : "#{db_field}")
        if is_custom_filter
          sql = "(#{db_table_and_field} <> '' AND CAST(CASE #{db_table_and_field}
WHEN '' THEN '0' ELSE #{db_table_and_field} END AS decimal(30,3)) = #{value.first.to_i})"
        else
          sql = "#{db_table_and_field} = #{value.first.to_i}"
        end
      end
      when :float
        if is_custom_filter
          sql = "(#{db_table}.#{db_field} <> '' AND CAST(CASE #{db_table}.#{db_field}
WHEN '' THEN '0' ELSE #{db_table}.#{db_field} END AS decimal(30,3)) BETWEEN #{value.first.to_f - 1e-5} AND #{
value.first.to_f + 1e-5})"
        else
          sql = "#{db_table}.#{db_field} BETWEEN #{value.first.to_f - 1e-5} AND #{value.first.to_f +
1e-5}"
        end
      end
      else
        sql = "#{db_table}.#{db_field} IN (" + value.collect{|val| "'#{connection.quote_string(val)}'"
}.join(",") + ")"
      end
      else
        # IN an empty set
        sql = "1=0"
      end
    end
  when "!="
    if value.any?
```

```

        sql = "#{db_table}.#{db_field} IS NULL OR #{db_table}.#{db_field} NOT IN (" + value.collect{|
val| "'#{connection.quote_string(val)}'").join(",") + ")"
    else
        # NOT IN an empty set
        sql = "1=1"
    end
when "!="
    # ENHANCEMENT HERE : manage case where db_table param is not provided
    db_table_and_field = (db_table.present? ? "#{db_table}.#{db_field}" : "#{db_field}")
    sql = "#{db_table_and_field} IS NULL"
    sql << " OR #{db_table_and_field} = ''" if is_custom_filter
when "*"
    db_table_and_field = (db_table.present? ? "#{db_table}.#{db_field}" : "#{db_field}")
    sql = "#{db_table_and_field} IS NOT NULL"
    sql << " AND #{db_table_and_field} <> ''" if is_custom_filter
when ">="
    if [:date, :date_past].include?(type_for(field))
        sql = date_clause(db_table, db_field, (Date.parse(value.first) rescue nil), nil)
    else
        # ENHANCEMENT HERE : manage case where db_table param is not provided
        db_table_and_field = (db_table.present? ? "#{db_table}.#{db_field}" : "#{db_field}")
        if is_custom_filter
            sql = "(#{db_table_and_field} <> '' AND CAST(CASE #{db_table_and_field}
WHEN '' THEN '0' ELSE #{db_table}.#{db_field} END AS decimal(30,3)) >= #{value.first.to_f})"
        else
            sql = "#{db_table_and_field} >= #{value.first.to_f}"
        end
    end
end
when "<="
    if [:date, :date_past].include?(type_for(field))
        sql = date_clause(db_table, db_field, nil, (Date.parse(value.first) rescue nil))
    else
        # ENHANCEMENT HERE : manage case where db_table param is not provided
        db_table_and_field = (db_table.present? ? "#{db_table}.#{db_field}" : "#{db_field}")
        if is_custom_filter
            sql = "(#{db_table_and_field} <> '' AND CAST(CASE #{db_table_and_field}
WHEN '' THEN '0' ELSE #{db_table_and_field} END AS decimal(30,3)) <= #{value.first.to_f})"
        else
            sql = "#{db_table_and_field} <= #{value.first.to_f}"
        end
    end
end
when "><"
    if [:date, :date_past].include?(type_for(field))
        sql = date_clause(db_table, db_field, (Date.parse(value[0]) rescue nil), (Date.parse(value[1])
rescue nil))
    else
        # ENHANCEMENT HERE : manage case where db_table param is not provided
        db_table_and_field = (db_table.present? ? "#{db_table}.#{db_field}" : "#{db_field}")
        if is_custom_filter
            sql = "(#{db_table_and_field} <> '' AND CAST(CASE #{db_table_and_field}
WHEN '' THEN '0' ELSE #{db_table_and_field} END AS decimal(30,3)) BETWEEN #{value[0].to_f} AND #{value[1].
to_f})"
        else
            sql = "#{db_table_and_field} BETWEEN #{value[0].to_f} AND #{value[1].to_f}"
        end
    end
end
when "o"
    sql = "#{queried_table_name}.status_id IN (SELECT id FROM #{IssueStatus.table_name}
WHERE is_closed=#{connection.quoted_false})" if field == "status_id"
when "c"
    sql = "#{queried_table_name}.status_id IN (SELECT id FROM #{IssueStatus.table_name}
WHERE is_closed=#{connection.quoted_true})" if field == "status_id"
when "><t-"
    # between today - n days and today
    sql = relative_date_clause(db_table, db_field, - value.first.to_i, 0)
when ">t-"
    # >= today - n days
    sql = relative_date_clause(db_table, db_field, - value.first.to_i, nil)
when "<t-"
    # <= today - n days
    sql = relative_date_clause(db_table, db_field, nil, - value.first.to_i)
when "t-"
    # = n days in past
    sql = relative_date_clause(db_table, db_field, - value.first.to_i, - value.first.to_i)
when "><t+"

```

```

        # between today and today + n days
        sql = relative_date_clause(db_table, db_field, 0, value.first.to_i)
    when ">t+"
        # >= today + n days
        sql = relative_date_clause(db_table, db_field, value.first.to_i, nil)
    when "<t+"
        # <= today + n days
        sql = relative_date_clause(db_table, db_field, nil, value.first.to_i)
    when "t+"
        # = today + n days
        sql = relative_date_clause(db_table, db_field, value.first.to_i, value.first.to_i)
    when "t"
        # = today
        sql = relative_date_clause(db_table, db_field, 0, 0)
    when "ld"
        # = yesterday
        sql = relative_date_clause(db_table, db_field, -1, -1)
    when "w"
        # = this week
        first_day_of_week = 1(:general_first_day_of_week).to_i
        day_of_week = Date.today.cwday
        days_ago = (day_of_week >= first_day_of_week ? day_of_week - first_day_of_week : day_of_week + 7
- first_day_of_week)
        sql = relative_date_clause(db_table, db_field, - days_ago, - days_ago + 6)
    when "lw"
        # = last week
        first_day_of_week = 1(:general_first_day_of_week).to_i
        day_of_week = Date.today.cwday
        days_ago = (day_of_week >= first_day_of_week ? day_of_week - first_day_of_week : day_of_week + 7
- first_day_of_week)
        sql = relative_date_clause(db_table, db_field, - days_ago - 7, - days_ago - 1)
    when "l2w"
        # = last 2 weeks
        first_day_of_week = 1(:general_first_day_of_week).to_i
        day_of_week = Date.today.cwday
        days_ago = (day_of_week >= first_day_of_week ? day_of_week - first_day_of_week : day_of_week + 7
- first_day_of_week)
        sql = relative_date_clause(db_table, db_field, - days_ago - 14, - days_ago - 1)
    when "m"
        # = this month
        date = Date.today
        sql = date_clause(db_table, db_field, date.beginning_of_month, date.end_of_month)
    when "lm"
        # = last month
        date = Date.today.prev_month
        sql = date_clause(db_table, db_field, date.beginning_of_month, date.end_of_month)
    when "y"
        # = this year
        date = Date.today
        sql = date_clause(db_table, db_field, date.beginning_of_year, date.end_of_year)
    when "~"
        sql = "LOWER(#{db_table}.#{db_field}) LIKE '%#{connection.quote_string(value.first.to_s.downcase
)}}%"
    when "!~"
        sql = "LOWER(#{db_table}.#{db_field}) NOT LIKE '%#{connection.quote_string(value.first.to_s.
downcase)}}%"
    else
        raise "Unknown query operator #{operator}"
    end

    return sql
end

```

```

# 3) Add a new filter in initialize_available_filters
add_available_filter 'parent_count', :type => :integer

```

```

# 4) Add a new label in the localization files :
field_parent_count: Parent count

```

## #2 - 2014-01-13 21:03 - Jérôme BATAILLE

- File *parent\_count\_filter\_all.png* added

- File *parent\_count\_filter\_level\_3.png* added

Example (Filter on root task is specific to our Redmine version) :

Issues

▼ Filters

☒ Root task

is

#28998 6 levels tree

▼

⊞

Add filter

☒ Parent count

>=

0

▼

► Options

✓ Apply

🔄 Clear

💾 Save

Total Budget time: 100 hours   Total time: 100 hours   Total Remaining time: 0 hour

✓	# ▼ 2	Project	Tracker	Parent task ▼	Status	Subject	Assignee	Estimated time	Budget time	Spent time	Remaining time	% Done
<input type="checkbox"/>	28998	Test CDC B & A	Task		New	6 levels tree			100	100	0	<div></div>
<input type="checkbox"/>	28999	Test CDC B & A	Task	Task #28998: 6 levels tree	New	► level. 2			90	90	0	<div></div>
<input type="checkbox"/>	29000	Test CDC B & A	Task	Task #28999: level. 2	New	► level. 3			70	70	0	<div></div>
<input type="checkbox"/>	29001	Test CDC B & A	Task	Task #29000: level. 3	New	► level. 4			40	40	0	<div></div>
<input type="checkbox"/>	29002	Test CDC B & A	Task	Task #29000: level. 3	New	► level 4. bis			0	0	0	<div></div>
<input type="checkbox"/>	29003	Test CDC B & A	Task	Task #29002: level 4. bis	New	► level 5			0	0	0	<div></div>
<input type="checkbox"/>	29004	Test CDC B & A	Task	Task #29003: level 5	New	► level 6			0	0	0	<div></div>
Total									100.00	100.00	0.00	

Issues

▼ Filters

☒ Root task

is

#28998 6 levels tree

▼

⊞

Add filter

☒ Parent count

>=

3

▼

► Options

✓ Apply

🔄 Clear

💾 Save

Total Budget time: 0 hour   Total time: 0 hour   Total Remaining time: 0 hour

✓	# ▼ 2	Project	Tracker	Parent task ▼	Status	Subject	Assignee	Estimated time	Budget time	Spent time	Remaining time	% Done
<input type="checkbox"/>	29001	Test CDC B & A	Task	Task #29000: level. 3	New	level. 4			40	40	0	<div></div>
<input type="checkbox"/>	29002	Test CDC B & A	Task	Task #29000: level. 3	New	level 4. bis			0	0	0	<div></div>
<input type="checkbox"/>	29003	Test CDC B & A	Task	Task #29002: level 4. bis	New	► level 5			0	0	0	<div></div>
<input type="checkbox"/>	29004	Test CDC B & A	Task	Task #29003: level 5	New	► level 6			0	0	0	<div></div>
Total									0.00	0.00	0.00	

(1-4/4)

#3 - 2014-02-07 06:26 - Toshi MARUYAMA

- Category set to Issues

#4 - 2014-06-19 14:18 - Jérôme BATAILLE

Hi, any chance this patch could be reviewed ?

#5 - 2016-04-07 01:25 - Jérôme BATAILLE

TODO: propose a cleaner patch

Files

parent_count_filter_all.png	56.1 KB	2014-01-13	Jérôme BATAILLE
parent_count_filter_level_3.png	40 KB	2014-01-13	Jérôme BATAILLE