Redmine - Feature #32311

Filter Issues by multiple issue_id doesn't support space character or semicolon/dash as the separator

2019-10-21 09:20 - Taine Woo

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Issues filter	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

Issues -> Filter -> Issue
statement "is", input the filter value as multiple ids, such as "1,2,3".

It only supports comma "," as the separator of issue ids.
It doesn't support other common separators, such as space, semicolon, dash.

And actually if we copied the ids from the id column from excel, it would be multiple lines combined with also semicolon.

History

#1 - 2019-10-21 09:35 - Taine Woo

If the value of the issue_id is "1,2,3" in the filter, the query executes successfully.
If the value of the issue_id is "1 2 3" or "1 2 3" or "1, 2, 3", the query fails with message "•Issue is invalid"

#2 - 2019-10-21 15:25 - Taine Woo

app/models/query.rb

line 417, replace the "," as "\D" can use any non-numeric characters as separator.

#3 - 2019-10-21 15:47 - Taine Woo

/\A[+-]?\d+(\D[+-]?\d+)*\z/
digit separated by one non-digit

/\A[+-]?\d+(\D*\d+)*\z/
digit separated by any non-digit, easier for users.

#4 - 2019-10-22 22:43 - Marius BĂLTEANU

- Tracker changed from Defect to Feature

It is the expected behaviour and I'm not sure that we should support more delimiters.

#5 - 2019-11-01 15:52 - Taine Woo

Marius BALTEANU wrote:

> It is the expected behaviour and I'm not sure that we should support more delimiters.

Hi Marius,

Currently the regexp is quite strict, only support digits seperated by only comma, no other character is allowed, it's not a bug.
But according to my survey, most of the users will not want to input the Issue ID one by one manually.
Users will copy the IDs from tables such as excel, the data would be "1 2 3 ", the delimiter should be "tab", and if copied from csv, may be the delimiter would be space.

So if we loose the restrict, users may have better experience.

#6 - 2019-12-25 09:35 - Taine Woo

Here is my patch, case check added for integer.

```
  def validate_query_filters
    filters.each_key do |field|
      if values_for(field)
        case type_for(field)
        when :integer
          case operator_for(field)
          when "="

#add_filter_error(field, :invalid) if values_for(field).detect {|v| v.present? && !v.match(/\A[+-]?\d+(\D[+-]?
\d+)*\z/) }
            add_filter_error(field, :invalid) if values_for(field).detect {|v| v.present? && !v.match(
/\A[+-]?\d+(\D*\d+)*\z/) }
          else
            add_filter_error(field, :invalid) if values_for(field).detect {|v| v.present? && !v.match(
/\A[+-]?\d+(,[+-]?\d+)*\z/) }
          end
        when :float
          add_filter_error(field, :invalid) if values_for(field).detect {|v| v.present? && !v.match(
/\A[+-]?\d+(\.\d*)?\z/) }
        when :date, :date_past
          case operator_for(field)
          when "=", ">=", "<=", "><"
            add_filter_error(field, :invalid) if values_for(field).detect {|v|
              v.present? && (!v.match(/\A\d{4}-\d{2}-\d{2}(T\d{2}((:)?\d{2}){0,2}(Z|\d{2}:?\d{2})?)?\z/) ||
parse_date(v).nil?)
            }
          when ">t-", "<t-", "t-", ">t+", "<t+", "t+", "><t+", "><t-"
            add_filter_error(field, :invalid) if values_for(field).detect {|v| v.present? && !v.match(/^\d+$/)
 }
          end
        end
      end

      add_filter_error(field, :blank) unless
          # filter requires one or more values
          (values_for(field) and !values_for(field).first.blank?) or
          # filter doesn't require any value
          ["o", "c", "!*", "*", "t", "ld", "w", "lw", "l2w", "m", "lm", "y", "*o", "!o"].include? operator_for
(field)
    end if filters
  end
```