For my work there is need to have a plan for the next month and report for the last. I have implemented this in Redmine by using additional filter options on date fields - last month, next month, this month. Patch attached. Maybe we can integrate this functionality into the trunk?

Related issues:
- Related to Redmine - Feature # 4729: Add Date-Based Filters for Issues List
  - Closed 2010-02-03
- Related to Redmine - Feature # 6954: Filter from date to date
  - Closed 2010-11-22
- Related to Redmine - Patch # 18868: Add support for queries with 'next week' ...
  - Closed

Associated revisions
Revision 17811 - 2019-01-19 08:42 - Go MAEDA
- New date filter operators: tomorrow, next week, next month (#4502).
  - Patch by Mizuki ISHIKAWA.

Revision 17898 - 2019-02-24 16:00 - Go MAEDA
- Update locales (#4502).

Revision 17942 - 2019-03-13 16:50 - Go MAEDA
- Fix the test (#4502).
  - Patch by Mizuki ISHIKAWA.

History
#1 - 2017-09-04 13:24 - Александр Курутин
- 7 years have passed, but there is still no filter (next month).

#2 - 2018-05-17 20:40 - Sebastian Paluch
- +1

#3 - 2018-05-27 10:47 - Go MAEDA
- Category set to Issues filter
#4 - 2018-12-18 12:25 - Go MAEDA
- Related to Patch #18868: Add support for queries with "next week" filter added

#5 - 2018-12-18 12:31 - Go MAEDA

Redmine 4.0.0 has "yesterday", "last week", and "last month" filter but does not "tomorrow", "next week", and "next month" filter.

I think those filters should be useful for many users.

#6 - 2019-01-10 01:48 - Mizuki ISHIKAWA
- File add-next-range-filters.patch added

Go MAEDA wrote:

Redmine 4.0.0 has "yesterday", "last week", and "last month" filter but does not "tomorrow", "next week", and "next month" filter.

I think those filters should be useful for many users.

I wrote a patch to add "tomorrow", "next week", and "next month" filter.

It is useful to have this filter when checking the tasks to be executed next month, next week, tomorrow.

#7 - 2019-01-10 02:42 - Mizuki ISHIKAWA
- File add-next-range-filters-v2.patch added

I fixed patch because unnecessary tests were included.

#8 - 2019-01-12 03:39 - Marius BALTEANU
- Target version set to Candidate for next major release

LGTM

#9 - 2019-01-13 05:44 - Go MAEDA
- File operators@2x.png added
- Target version changed from Candidate for next major release to 4.1.0

Setting the target version to 4.1.0.

I propose reordering the operators as follows. While the current patch shows operators in "present -> past -> future" order, my change shows operators in "future -> present -> past" order. For example, operators will be ordered like "tomorrow, today, yesterday" and "next week, this week, last week, last 2 weeks" instead of "today, yesterday, tomorrow" and "this week, last week, last 2 weeks, next week". The proposing order is a tidy reverse-chronological order and should be intuitive.

```
:date => [ "w", ">=", "<=", "<", "=", "<+", "<t+", "<t+", "nd", "t", "ld", "nw", "w", "lw", "nm", "m", "lm", "y", ">t-", "<t-", "<t-", "t-", "$" ]
```
#10 - 2019-01-18 09:39 - Go MAEDA
- Subject changed from Query date filters by months (prev, this, next) to New date filter operators: tomorrow, next week, next month

#11 - 2019-01-19 08:43 - Go MAEDA
- Status changed from New to Closed
- Assignee set to Go MAEDA

Committed the patch. Thank you for improving Redmine.

I changed the order or operators and added Japanese translation.

#12 - 2019-01-19 09:33 - Go MAEDA
- Status changed from Closed to Reopened

QueryTest#test_operator_tomorrow fails depending on timezone.

I could reproduce the problem at 22:21 HST (Hawaii Standard Time -1000, 08:21 UTC). Maybe the problem occurs if the local date is different from the UTC date.

```bash
$ date
Fri Jan 18 22:21:34 HST 2019
$ ruby test/unit/query_test.rb
Run options: --seed 35057

# Running:
.................................................................................................................F

Failure:
QueryTest#test_operator_tomorrow [test/unit/query_test.rb:645]:
Expected false to be truthy.

bin/rails test/unit/query_test.rb:641

.................................................................................................................

Finished in 22.371776s, 9.3421 runs/s, 26.4619 assertions/s.
209 runs, 592 assertions, 1 failures, 0 errors, 0 skips
```

#13 - 2019-01-19 10:52 - Go MAEDA

The current date of my PC is 2019-01-18 HST (2019-01-19 UTC).

QueryTest#test_operator_tomorrow expects that there are some issues whose due date is tomorrow (2019-01-19). However, there are issues whose due date is the day after the tomorrow (2019-01-20).
```sql
sqlite> select id, due_date from issues;

<table>
<thead>
<tr>
<th>id</th>
<th>due_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2019-01-29</td>
</tr>
<tr>
<td>3</td>
<td>2019-01-14</td>
</tr>
<tr>
<td>5</td>
<td>2019-01-18</td>
</tr>
<tr>
<td>6</td>
<td>2019-01-20</td>
</tr>
<tr>
<td>9</td>
<td>2019-01-20</td>
</tr>
</tbody>
</table>

Probably the reason is that the timezone of fixtures is UTC. 1.days.from_now.to_date.to_s(:db) returns a date in UTC (2019-01-20), while the test code expects 2019-01-19 that is the date of tomorrow in the local time (the current date of the local time is still 2019-01-18 HST).

```ruby
issues_006:  
  created_on: <%= 1.minute.ago.to_s(:db) %>  
  project_id: 5  
  updated_on: <%= 1.minute.ago.to_s(:db) %>  
  priority_id: 4  
  subject: Issue of a private subproject  
  id: 6  
  fixed_version_id:  
  category_id:  
  description: This is an issue of a private subproject of cookbook  
  tracker_id: 1  
  assigned_to_id:  
  author_id: 2  
  status_id: 1  
  start_date: <%= Date.today.to_s(:db) %>  
  due_date: <%= 1.days.from_now.to_date.to_s(:db) %>  
  root_id: 6  
  lft: 1  
  rgt: 2
```

The date of fixtures are UTC, I think we have to run the test in UTC. The following workaround resolves the failing test.

Index: test/unit/query_test.rb

--- test/unit/query_test.rb (revision 17811)  
+++ test/unit/query_test.rb (working copy)  
@@ -639,11 +639,13 @@

end

def test_operator_tomorrow

2020-04-03
User.current = User.find_by_login('admin')
User.current.pref.update_attribute :time_zone, 'UTC'

query = IssueQuery.new({:project => Project.find(1), :name => '_'}
query.add_filter('due_date', 'nd', [''])
issues = find_issues_with_query(query)
assert !issues.empty?
- issues.each{|issue| assert_equal Date.today.tomorrow, issue.due_date}
+ issues.each{|issue| assert_equal User.current.today.tomorrow, issue.due_date}
end

def test_operator_date_periods

#14 - 2019-03-05 03:21 - Mizuki ISHIKAWA
- File fix_test.patch added

Go MAEDA wrote:

> QueryTest#test_operator_tomorrow fails depending on timezone.

> I could reproduce the problem at 22:21 HST (Hawaii Standard Time -1000, 08:21 UTC). Maybe the problem occurs if the local date is different from the UTC date.

Thank you for pointing that out.

The attached patch updates the #4502#note-13 test to fix the random failure.

#15 - 2019-03-14 14:33 - Go MAEDA

- Status changed from Reopened to Closed

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>query_months_filters.patch</td>
<td>3.8 KB</td>
<td>2009-12-29</td>
<td>Andrew Chaika</td>
</tr>
<tr>
<td>add-next-range-filters.patch</td>
<td>7.18 KB</td>
<td>2019-01-10</td>
<td>Mizuki ISHIKAWA</td>
</tr>
<tr>
<td>add-next-range-filters-v2.patch</td>
<td>6.85 KB</td>
<td>2019-01-10</td>
<td>Mizuki ISHIKAWA</td>
</tr>
<tr>
<td><a href="mailto:operators@2x.png">operators@2x.png</a></td>
<td>48.3 KB</td>
<td>2019-01-13</td>
<td>Go MAEDA</td>
</tr>
<tr>
<td>fix_test.patch</td>
<td>910 Bytes</td>
<td>2019-03-05</td>
<td>Mizuki ISHIKAWA</td>
</tr>
</tbody>
</table>