Redmine - Feature #23581

Query class inheritance

2016-08-12 17:21 - Andrey Iskousnykh

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

Description

I have a class inherited from query to perform filters and query jobs of my plugin. It works but some functions could be improved for better inheritance capability and extended functionality. Here some thin places:

1. some filters are desired to be not used for database requests.
2. some filters are desired to be a list value - suppose to have no operators.
3. filters could have its own operators.

There first problem is solved by overriding validate_query_filters method and playing with query.filters hash and writing sql_for_XXXX_field that returns empty string.

Here's the thin place for custom operators

```
    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
```

["o", "c", "!*", "*", "t", "ld", "w", "lw", "l2w", "m", "lm", "y", "*o", "!o"].include? operator_for(field) is better to be replaced with something like this unary_operators.include? operator_for(field)

Even more validate_query_filters should support validation of each filter separately. This should give availability of overriding filter validation independently.