# Redmine - Feature #17889

## Searches should be twice faster

2014-09-16 18:06 - Olivier Houdas

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

**Description**

When we run a search, we can see in the debug log that for each category to look into ( issues, forums, wiki...) there are 2 queries which are almost the same, and take the same time. One is for counting results, the other one is for getting the first 11 lines.
This is discussed in http://www.redmine.org/boards/1/topics/41475 for example.

In /lib/plugins/acts_as_searchable.rb, line 126 and 128, we have (I think) the origin of those 2 queries:

```
l 126: results_count = scope.count
l 128: scope_with_limit = scope.limit(options[:limit])
```

Wouldn't it be possible to include a column like

```
"total=COUNT(*) OVER()"
```

(MSSQL or Oracle syntax) to get that results_count in only one query? I tested the query, and it takes the same time with or without the total column, even if on a large table (7000ms of query execution).

I mean, instead of having 2 queries:

```
[1m[35m (7805.9ms)[0m  EXEC sp_executesql N'SELECT COUNT(DISTINCT [issues].[id]) FROM ... ))))))
```

and

```
[1m[36mSQL (7592.5ms)[0m  [1mEXEC sp_executesql N'SELECT TOP (11) [issues].id FROM ... ))))))  GRO
UP BY [issues].id, issues.id ORDER BY issues.id DESC'[0m
```

We would have only one query:

```
[1m[36mSQL (7592.5ms)[0m  [1mEXEC sp_executesql N'SELECT TOP (11) [issues].id, total=COUNT(*) OVER
() FROM ... ))))))  GROUP BY [issues].id, issues.id ORDER BY issues.id DESC'[0m
```

And that would make searches twice faster.

**Related issues:**

| | | |
|---|---|---|
| Related to Redmine - Feature #18631: Better search results pagination | **Closed** | |

### History

#### #1 - 2014-09-18 14:34 - Holger Just

OVER() is not supported in Sqlite3 and MySQL, so this is probably hard to implement in the general case. It might however be feasible as an optimization for MS SQL and PotgreSQL with different code paths respectively.

#### #2 - 2014-12-12 21:55 - Jean-Philippe Lang

*- Related to Feature #18631: Better search results pagination added*

#### #3 - 2014-12-23 11:09 - Tomasz Kowalczyk

In MySQL you can use FOUND_ROWS(): http://dev.mysql.com/doc/refman/5.7/en/information-functions.html#function_found-rows