redmineorg-copy202205 - Vote #80871

Better performance for Time entries without issue and activity filters

2022/05/09 19:05 - Admin Redmine

ステータス:	New	**開始日:**	2022/05/09
優先度:	通常	**期日:**	
担当者:		**進捗率:**	0%
カテゴリ:	Performance_53	**予定工数:**	0.00時間
対象バージョン:	Candidate for next major release_32	**作業時間:**	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/33431	**status_id:**	1
category_id:	53	**tracker_id:**	3
version_id:	32	**plus1:**	0
issue_org_id:	33431	**affected_version:**	
author_id:	29628	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	
comments:	6		

説明

At the moment all time entry queries join issue and activities. If there is a large number of time entries and issues, this is a big performance issue.

Some of our customers have more than 500.000 time entries and even more issues. A query without a time filter requires minutes. With this attached patch this improves it to some seconds.

Issue and activity joins are only add, if they are required.

It would be great, to get this in Redmine.

journals

Looks good to me.

My patch has problems with column sort order.

But I found the real problem now. It is a missing condition to project_id for Enumeration. I am working on a new patch to fix this problem. Please do not commit the current patch.

Alexander Meindl wrote:

> My patch has problems with column sort order.
>
> But I found the real problem now. It is a missing condition to project_id for Enumeration. I am working on a new patch to fix this problem. Please do not commit the current patch.

Nice feature, but we should have tests for this change.

Here is my new fixed version.

Adding

`where("#{Enumeration.table_name}.project_id = #{TimeEntry.table_name}.project_id")`

fixed the problem. But I moved activity join from base_scope to results_scope, because as I understood it, activity join is not required for totals and count.

I am not sure, how I can test this change with an additional test. Existing test should work, of course.

And again, now all tests are running.

With my test data, /time_entries without filters

- without patch: ActiveRecord: 345038.1ms
- with patch ActiveRecord: 6038.4ms

This is not perfect, but way better than before. At the moment I have no idea how it can be more improved. The problem is table layout of enumerations table.

履歴

#1 - 2022/05/10 17:00 - Admin Redmine

- カテゴリ を Performance_53 にセット

- 対象バージョン を Candidate for next major release_32 にセット