Redmine - Patch #26122

Implementation of visible conditions with inner join instead of subselect

2017-06-07 22:51 - Pavel Rosický

Status:	Resolved	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Performance	**Estimated time:**	0.00 hour
Target version:	Candidate for next major release		

Description

The change from #21608 should be reverted because it's a speed regression:

1/ current version

```
EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em. NAME = '
issue_tracking')
```

2/ previous version (actually faster even without an index on enabled modules)

```
projects.id IN (SELECT project_id FROM enabled_modules em WHERE em.project_id = projects.id AND em
. NAME = 'issue_tracking'
```

3/ fastest version (in some cases better indexes were used)

```
INNER JOIN `enabled_modules` ON `enabled_modules`.`project_id` = `projects`.`id` WHERE `enabled_mo
dules`.`name` = 'issue_tracking'
```

or

```
INNER JOIN `enabled_modules` ON `enabled_modules`.`project_id` = `projects`.`id` AND `enabled_modu
les`.`name` = 'issue_tracking'
```

Patches:

- revert of #21608, because the previous version was faster (about 30%)
- new index "enabled_modules_name", but it's not very helpful unless you have many projects (3000+)
- implementation of visible conditions with inner join instead of subselect, it passes all tests, but it should be refactored. I want to know what do you think about it first

```
Rails version              4.2.8
Ruby version               2.1.9-p490 (x64-mingw32)
RubyGems version           2.6.12
Rack version               1.6.8
Middleware                 Rack::Sendfile, Rack::ContentLength, ActionDispatch::Static, Rack::Lock,
 #<ActiveSupport::Cache::Strategy::LocalCache::Middleware:0x00000007745610>, Rack::Runtime, Rack::
MethodOverride, ActionDispatch::RequestId, Rails::Rack::Logger, ActionDispatch::ShowExceptions, Ac
tionDispatch::DebugExceptions, ActionDispatch::RemoteIp, ActionDispatch::Reloader, ActionDispatch:
:Callbacks, ActiveRecord::ConnectionAdapters::ConnectionManagement, ActiveRecord::QueryCache, Acti
onDispatch::Cookies, ActionDispatch::Session::CookieStore, ActionDispatch::Flash, ActionDispatch::
ParamsParser, ActionDispatch::XmlParamsParser, Rack::Head, Rack::ConditionalGet, Rack::ETag, Reque
stStore::Middleware, OpenIdAuthentication
Environment                development
Database adapter           mysql2
Database schema version    20170607051650
```

History

#1 - 2017-06-08 13:29 - Pavel Rosický

- Status changed from New to Resolved

#2 - 2017-06-08 13:39 - Pavel Rosický

I did some additional benchmarking and I was wrong. Exists really performs better.

so the only relevant patch is joins_enabled_modules.patch

Issue.visible is faster on my environment with this patch about 10% compared to exists with all caching disabled. I don't think it's worth to use it on other places.

#3 - 2017-06-09 08:57 - Mischa The Evil

- Subject changed from revert #21608 to Implementation of visible conditions with inner join instead of subselect

- Description updated

- Status changed from Resolved to New

Edited issue properties based on the feedback and slightly improved the markup of the description.

#4 - 2017-06-29 15:51 - Toshi MARUYAMA

- Target version set to 3.4.0

#5 - 2017-06-29 16:02 - Toshi MARUYAMA

- Target version changed from 3.4.0 to 4.1.0

#6 - 2017-06-30 13:44 - Fernando Hartmann

As a speed regression, I suggest to put this on 3.4.1

#7 - 2017-06-30 14:10 - Pavel Rosický

Well, it havily depends on what filters and other conditions are used, it isn't easy to measure. For my long time observations using INNER JOINS is slightly faster (especially on big tables like issues) and mysql has better working plans with it.
On the other side, joins can't be passed directly into WHERE statement which is what allowed_to_condition should do - it increases code complexity. I want some opinions if you think it's worth to use it or not.
Fernando Sanche - as discused before, it's not a speed regression just an improvement, so 3.5.0 is fine.

#8 - 2019-09-27 17:34 - Marius BĂLTEANU

- Target version changed from 4.1.0 to Candidate for next major release

Pavel Rosický wrote:

> Well, it havily depends on what filters and other conditions are used, it isn't easy to measure. For my long time observations using INNER JOINS is slightly faster (especially on big tables like issues) and mysql has better working plans with it.
> On the other side, joins can't be passed directly into WHERE statement which is what allowed_to_condition should do - it increases code complexity. I want some opinions if you think it's worth to use it or not.
> Fernando Sanche - as discused before, it's not a speed regression just an improvement, so 3.5.0 is fine.

I'm removing this from 4.1.0 because it is not a simple change and without performance tests that can be run before and after on both mysql and postgres, it is hard to say if the code complexity added by the patch it's worth it or not. Please correct me if I am wrong.

#9 - 2019-09-27 18:22 - Pavel Rosický

- Status changed from New to Resolved

no, I agree. Thanks for the feedback, this can be closed.

Files

20170607051650_add_index_on_enabled_modules_name.rb.patch	416 Bytes	2017-06-07	Pavel Rosický
revert_exists.patch	959 Bytes	2017-06-07	Pavel Rosický
joins_enabled_modules.patch	11 KB	2017-06-07	Pavel Rosický