

## Redmine - Defect #29848

### Query to determine role ids is incorrect

2018-10-25 21:31 - Frank Schwarz

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Database	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	3.4.6
<b>Resolution:</b>	Invalid		

#### Description

After migrating to stable branch 3.4, the issues page stopped listing entries except for admin users.

Some debugging revealed a query, that seems to be the culprit:

```
SELECT
  `members`.`user_id`,
  `members`.`role_id`,
  `members`.`project_id`
FROM `members`
  INNER JOIN `projects` ON `projects`.`id` = `members`.`project_id`
  INNER JOIN `member_roles` ON `member_roles`.`member_id` = `members`.`id`
WHERE
  (projects.status <> 9) AND
  (members.user_id = 7 OR (projects.is_public = 1 AND members.user_id = 36));
```

For some reasons unknown to me, the column `members`.`role\_id` is always 0. However in table `member\_roles`, the corresponding column is filled correctly. IMHO, the query should actually look like this one:

```
SELECT
  `members`.`user_id`,
  `member_roles`.`role_id`, -- **** instead of `members`.`role_id`
  `members`.`project_id`
FROM `members`
  INNER JOIN `projects` ON `projects`.`id` = `members`.`project_id`
  INNER JOIN `member_roles` ON `member_roles`.`member_id` = `members`.`id`
WHERE
  (projects.status <> 9) AND
  (members.user_id = 7 OR (projects.is_public = 1 AND members.user_id = 36));
```

The suggested patch is this:

git diff user.rb

```
diff --git a/app/models/user.rb b/app/models/user.rb
index 357856609..0f0d7d64b 100644
--- a/app/models/user.rb
+++ b/app/models/user.rb
@@ -614,7 +614,7 @@ class User < Principal
     members = Member.joins(:project, :member_roles)
       .where("#{Project.table_name}.status <> 9")
       .where("#{Member.table_name}.user_id = ? OR (#{Project.table_name}.is_public = ? AND #{Member.table_name}.user_id = ?)", self.id, true, group_id).
-     pluck(:user_id, :role_id, :project_id)
+     pluck(:user_id, "#{MemberRole.table_name}.role_id", :project_id)

     hash = {}
     members.each do |user_id, role_id, project_id|
```

#### History

### #1 - 2018-10-27 04:08 - Go MAEDA

I could not reproduce the problem.

Could you try without plugins if you are using any plugins?

### #2 - 2018-10-30 21:00 - Frank Schwarz

There were no pertinent plugins active at the time the error occurred. I migrated from 3.2-stable. Project membership was set up via groups (not via individual users).

I wonder why there are two `role_id` columns: one in "Members" and one in "MemberRoles". The former remains "0" for all project-member assignments.

Current environment:

#### Environment:

```
Redmine version      3.4.6.stable
Ruby version         2.5.1-p57 (2018-03-29) [x86_64-linux-gnu]
Rails version        4.2.10
Environment          production
Database adapter     Mysql2
```

#### SCM:

```
Subversion           1.9.7
Git                  2.17.1
Filesystem
```

#### Redmine plugins:

```
redmine_auth_source_passwd 1.0.0.dev --> another AuthSource
redmine_html_attachments   0.0.1.dev --> patches Attachment to allow HTML-Attachments
redmine_ics_export         3.0.1.dev --> https://github.com/buschmais/redmics
redmine_overdue_notification_task 0.3.0 --> extends Mailer + Rake task
```

### #3 - 2020-02-23 10:15 - Frank Schwarz

- Status changed from *New* to *Resolved*

For some unknown reasons the members table (still) had a `role_id` column. Dropping that column manually, removed the join-table confusion.

### #4 - 2020-02-23 13:21 - Marius BĂLTEANU

- Status changed from *Resolved* to *Closed*

- Resolution set to *Invalid*

Frank Schwarz wrote:

For some unknown reasons the members table (still) had a `role_id` column. Dropping that column manually, removed the join-table confusion.

Thanks for your feedback.