Redmine - Patch #29161

Avoid SQL errors when adding a project custom field as a time report criteria

2018-06-29 17:14 - Holger Just

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Go MAEDA		**% Done:**	0%
Category:	Custom fields		**Estimated time:**	0.00 hour
Target version:	4.0.0			

Description

Custom fields can have different visibilities, depending on the custom field type and the current user:

- Custom fields are always visible to admins
- They are not visible for anonymous unless explicitly checked
- They are visible if the visible flag is checked (or the user in in a checked group for issue custom fields)

Now, to confirm whether a custom field is visible in a certain project, we use CustomField#visibility_by_project_condition. Given the following conditions:

- We are a non-admin user
- We defined a ProjectCustomField of type List which is not visible

Now, when we show a time entry report and select the ProjectCustomField as an aggregation criteria, we get an error because of invalid SQL:

```
ActiveRecord::StatementInvalid: Mysql2::Error: Unknown column 'projects.project_id' in 'IN/ALL/ANY
 subquery':
SELECT Sum(`time_entries`.`hours`) AS sum_hours,
       Coalesce(cf_9.value, '')    AS coalesce_cf_9_value,
       `time_entries`.`tyear`      AS time_entries_tyear,
       `time_entries`.`tmonth`     AS time_entries_tmonth,
       `time_entries`.`tweek`      AS time_entries_tweek,
       `time_entries`.`spent_on`   AS time_entries_spent_on
FROM   `time_entries`
       INNER JOIN `projects`
               ON `projects`.`id` = `time_entries`.`project_id`
       LEFT OUTER JOIN `enumerations`
                    ON `enumerations`.`id` = `time_entries`.`activity_id`
                       AND `enumerations`.`type` IN ( 'TimeEntryActivity' )
       LEFT OUTER JOIN `issues`
                    ON `issues`.`id` = `time_entries`.`issue_id`
       LEFT OUTER JOIN custom_values cf_9
                    ON cf_9.customized_type = 'Project'
                       AND cf_9.customized_id = projects.id
                       AND cf_9.custom_field_id = 9
                       AND ( projects.project_id IN
                             (SELECT DISTINCT m.project_id
                              FROM    members m
                                      INNER JOIN member_roles mr
                                              ON mr.member_id =
                                                 m.id
                                      INNER JOIN custom_fields_roles cfr
                                              ON cfr.role_id = mr.role_id
                                                 WHERE   m.user_id = 3
                                                         AND
                             cfr.custom_field_id = 9
                                                 ) )
                       AND cf_9.value <> ''
                       AND cf_9.id = (SELECT Max(cf_9_2.id)
                                      FROM    custom_values cf_9_2
                                      WHERE   cf_9_2.customized_type =
                                              cf_9.customized_type
```

```
                                          AND cf_9_2.customized_id =
                                              cf_9.customized_id
                                          AND cf_9_2.custom_field_id =
                                              cf_9.custom_field_id)
WHERE  (( ( projects.status <> 9
           AND projects.id IN ( 1, 2, 3, 5,
                                12, 14, 15, 19, 25 ) )
         AND ( projects.id IN ( 1, 2, 4, 5 ) ) ))
      AND (( time_entries.spent_on IS NOT NULL ))
GROUP  BY Coalesce(cf_9.value, ''),
         `time_entries`.`tyear`,
         `time_entries`.`tmonth`,
         `time_entries`.`tweek`,
         `time_entries`.`spent_on`
```

The reason for that is the SQL generated by the CustomField#visibility_by_project_condition method called for Redmine::FieldFormat::Base#join_for_order_statement Here, we are using the final else case and attempt to use the project_id field on the projects table.

The correct column here would however simply be id instead of project_id. The attached patch fixes this behavior and avoids the SQL error. Unfortunately, I'm not really sure how to properly test this. This whole area seems to be not well tested currently...

Related issues:	
Related to Redmine - Patch #29162: Only allow visible custom fields as aggreg...	**Closed**

Associated revisions

Revision 17517 - 2018-09-24 07:31 - Go MAEDA

Use the id field as project_key in ProjectCustomField#visibility_by_project_condition by default (#29161).

Patch by Holger Just.

History

#1 - 2018-06-29 17:31 - Holger Just

- Related to Patch #29162: Only allow visible custom fields as aggregation criteria in time reports added

#2 - 2018-07-12 07:53 - Go MAEDA

- Target version set to Candidate for next minor release

#3 - 2018-09-20 19:26 - Holger Just

- File 0001-Use-the-id-field-as-project_key-in-ProjectCustomFiel.patch added

i just updated the patch to include a test for this behavior.

While the symptom described here doesn't occur that way after #29162 was applied, the underlying issue still stands. This patch thus fixes the behavior of the ProjectCustomField#visibility_by_project_condition method. Without the applied model fix, the test fails with the described SQL error.

#4 - 2018-09-20 19:26 - Holger Just

- File deleted (0001-Use-the-id-field-as-project_key-in-ProjectCustomFiel.patch)

#5 - 2018-09-24 07:32 - Go MAEDA

- Status changed from New to Closed

- Assignee set to Go MAEDA

- Target version changed from Candidate for next minor release to 4.0.0

Committed. Thank you for reporting and fixing this issue.

Files

0001-Use-the-id-field-as-project_key-in-ProjectCustomFiel.patch	1.9 KB	2018-09-20	Holger Just