

Redmine - Defect #38788

"has been" and "has never been" filter operators can be very slow

2023-06-28 06:45 - Go MAEDA

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Fixed		

Description

The two filter operators "has been" and "has never been" introduced in [#38527](#) can be very slow when there are a large number of rows in the journal_details table.

The attached patch improves the performance by changing the SQL. The patch moves an additional query to outside the EXISTS. In my environment with 150,000 rows of journal_details, a query that took 230 seconds was reduced to 200 milliseconds.

before:

```
EXISTS (  
  SELECT 1  
  FROM journals  
  INNER JOIN journal_details ON journals.id = journal_details.journal_id  
  WHERE (  
    (  
      journals.private_notes = FALSE  
      OR journals.user_id = ?  
      OR (projects.status <> 9 AND projects.status <> 10)  
    )  
    AND journals.journalized_type = 'Issue'  
    AND journals.journalized_id = issues.id  
    AND journal_details.property = 'attr'  
    AND journal_details.prop_key = 'status_id'  
    AND journal_details.old_value = '?'  
  )  
  OR issues.status_id = '?'  
)
```

```
(228880.7ms) SELECT COUNT(*) FROM "issues" INNER JOIN "projects" ON "projects"."id" = "issues"  
."project_id" INNER JOIN "issue_sta  
tuses" ON "issue_statuses"."id" = "issues"."status_id" WHERE (projects.status <> 9 AND projects.st  
atus <> 10 AND EXISTS (SELECT 1 AS  
one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name='issue_tracking')) AND (  
(NOT EXISTS (SELECT 1 FROM journals  
INNER JOIN journal_details ON journals.id = journal_details.journal_id WHERE ((journals.private_n  
otes = FALSE OR journals.user_id =  
3 OR (projects.status <> 9 AND projects.status <> 10)) AND journals.journalized_type = 'Issue' AND  
journals.journalized_id = issues.i  
d AND journal_details.property = 'attr' AND journal_details.prop_key = 'status_id' AND journal_det  
ails.old_value IN ('2')) OR issues.  
status_id IN ('2')))) AND projects.lft >= 3 AND projects.rgt <= 8)  
↳ app/models/issue_query.rb:359:in `issue_count'
```

after:

```
EXISTS (  
  SELECT 1  
  FROM journals  
  INNER JOIN journal_details ON journals.id = journal_details.journal_id  
  WHERE (  
    (  
      journals.private_notes = FALSE  
      OR journals.user_id = ?  
      OR (projects.status <> 9 AND projects.status <> 10)  
    )  
    AND journals.journalized_type = 'Issue'  
    AND journals.journalized_id = issues.id  
    AND journal_details.property = 'attr'  
    AND journal_details.prop_key = 'status_id'  
    AND journal_details.old_value = '?'  
  )  
  OR issues.status_id = '?'  
)
```

```
(
  journals.private_notes = FALSE
  OR journals.user_id = ?
  OR (projects.status <> 9 AND projects.status <> 10)
)
AND journals.journalized_type = 'Issue'
AND journals.journalized_id = issues.id
AND journal_details.property = 'attr'
AND journal_details.prop_key = 'status_id'
AND journal_details.old_value = '?'
)
)
OR issues.status_id = '?'

(201.6ms) SELECT COUNT(*) FROM "issues" INNER JOIN "projects" ON "projects"."id" = "issues"."p
roject_id" INNER JOIN "issue_status
es" ON "issue_statuses"."id" = "issues"."status_id" WHERE (projects.status <> 9 AND projects.statu
s <> 10 AND EXISTS (SELECT 1 AS one
FROM enabled_modules em WHERE em.project_id = projects.id AND em.name='issue_tracking')) AND ((NO
T (EXISTS (SELECT 1 FROM journals I
NNER JOIN journal_details ON journals.id = journal_details.journal_id WHERE ((journals.private_not
es = FALSE OR journals.user_id = 3
OR (projects.status <> 9 AND projects.status <> 10)) AND journals.journalized_type = 'Issue' AND j
ournals.journalized_id = issues.id
AND journal_details.property = 'attr' AND journal_details.prop_key = 'status_id' AND journal_detai
ls.old_value IN ('2')))) OR issues.s
tatus_id IN ('2')))) AND projects.lft >= 3 AND projects.rgt <= 8)
↳ app/models/issue_query.rb:359:in `issue_count'
```

Related issues:
Related to Redmine - Feature #38527: New issues filter operators "has been", ... Closed

Associated revisions

Revision 22271 - 2023-06-30 11:40 - Go MAEDA

"has been" and "has never been" filter operators can be very slow (#38527, #38788).

Patch by Go MAEDA.

History

#1 - 2023-06-28 06:46 - Go MAEDA

- Related to Feature #38527: New issues filter operators "has been", "has never been", and "changed from" added

#2 - 2023-06-30 11:40 - Go MAEDA

- Status changed from New to Closed
- Assignee set to Go MAEDA
- Resolution set to Fixed

Committed the fix as a part of [#38527](#).

#3 - 2023-06-30 11:41 - Go MAEDA

- Subject changed from "has been" or "has never been" filter operator can be very slow to "has been" and "has never been" filter operators can be very slow

Files

fix-slow-query-of-has-been-opr.patch	823 Bytes	2023-06-28	Go MAEDA
--------------------------------------	-----------	------------	----------