

Redmine - Defect #32610

A inner join can be removed

2019-12-11 18:24 - jwjw yy

Status: New	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	Affected version:
Resolution:	
Description	
There is a query	
<pre>SELECT DISTINCT(issues.id) FROM issues INNER JOIN projects ON projects.id = issues.project_id INNER JOIN issue_statuses ON issue_statuses.id = issues.status_id LEFT OUTER JOIN users ON users.id = issues.assigned_to_id LEFT OUTER JOIN trackers ON trackers.id = issues.tracker_id LEFT OUTER JOIN enumerations ON enumerations.id = issues.priority_id AND enumerations.type IN ('IssuePriority') LEFT OUTER JOIN versions ON versions.id = issues.priority_id WHERE (projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=0)) AND projects.id = 3821) ORDER BY issues.id DESC LIMIT 501;</pre>	
which actually can be simplified due to the foreign key and presence constraint	
-- remove LEFT OUTER JOIN (because of id being PK, already done by postgres)	
-- remove INNER JOIN (because of FK; no need to add issues.project_id IS NOT NULL / issues.status_id IS NOT NULL because of constraint)	
-- remove DISTINCT (FK)	
After removing, the query becomes:	
<pre>SELECT issues.id FROM issues INNER JOIN projects ON projects.id = issues.project_id LEFT OUTER JOIN enumerations ON enumerations.id = issues.priority_id AND enumerations.type IN ('IssuePriority') WHERE (projects.status <> 9 AND projects.id IN (SELECT em.project_id FROM enabled_modules em WHERE em.name='issue_tracking')) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=0)) AND projects.id = 3821) ORDER BY issues.id DESC LIMIT 501;</pre>	
It can make the code easier to read and faster	