Redmine - Defect #6468

wrong update query in Issue model

2010-09-23 07:44 - Nelzin Alexander

Status:	Closed	**Start date:**	2010-09-23
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Resolution:	Cant reproduce	**Affected version:**	1.0.1

Description

Redmine 1.0.1.devel (OracleEnhanced)

There was a problem during creating or updating issues. It happened in Issue instance method "update_nested_set_attributes". The update query was formed invalid. Like this:

```
update issues set root_id = , lgt = 21 ...
                             * blank field
```

line 651 in Issue.rb:

```
      Issue.update_all("root_id = #{root_id}, lft = #{lft}, rgt = #{rgt}", ["id = ?", id])
```

line 672-673 in Issue.rb:

```
        Issue.update_all("root_id = #{root_id}, lft = lft + #{offset}, rgt = rgt + #{offset}",
                         ["root_id = ? AND lft >= ? AND rgt <= ? ", old_root_id, lft, rgt])
```

in Redmine 1.0.1.stable.1536 (OracleEnhanced)
The query is generated in the same wrong way.

Here is my diff to make update query be well formed

```
svn diff --old=app/models/issue.rb@1537 --new=app/models/issue.rb@1548
Index: app/models/issue.rb
===================================================================
--- app/models/issue.rb    (revision 1537)
+++ app/models/issue.rb    (revision 1548)
@@ -628,7 +628,7 @@
       # issue was just created
       self.root_id = (@parent_issue.nil? ? id : @parent_issue.root_id)
       set_default_left_and_right
-      Issue.update_all("root_id = #{root_id}, lft = #{lft}, rgt = #{rgt}", ["id = ?", id])
+      Issue.update_all(["root_id = ?, lft = ?, rgt = ?", root_id, lft, rgt], ["id = ?", id])
       if @parent_issue
         move_to_child_of(@parent_issue)
       end
@@ -649,8 +649,8 @@
         self.root_id = (@parent_issue.nil? ? id : @parent_issue.root_id )
         target_maxright = nested_set_scope.maximum(right_column_name) || 0
         offset = target_maxright + 1 - lft
-        Issue.update_all("root_id = #{root_id}, lft = lft + #{offset}, rgt = rgt + #{offset}",
-                         ["root_id = ? AND lft >= ? AND rgt <= ? ", old_root_id, lft, rgt])
+        Issue.update_all("root_id = ?, lft = lft + #{offset}, rgt = rgt + #{offset}",
+                         ["root_id = ? AND lft >= ? AND rgt <= ? ", root_id, old_root_id, lft, r
gt])
         self[left_column_name] = lft + offset
         self[right_column_name] = rgt + offset
```

```
        if @parent_issue
```

History

#1 - 2010-09-23 07:46 - Nelzin Alexander

```
~/Sites/redmine-1.0 $ ruby script/about
About your application's environment
Ruby version              1.8.7 (universal-darwin10.0)
RubyGems version          1.3.5
Rack version              1.0
Rails version             2.3.5
Active Record version     2.3.5
Active Resource version   2.3.5
Action Mailer version     2.3.5
Active Support version    2.3.5
Application root          /Users/alexander_nelzin/Sites/redmine-1.0
Environment               development
Database adapter          oracle_enhanced
Database schema version   20100819172912
```

#2 - 2010-09-23 08:41 - Jean-Baptiste Barth

This one looks OK. If somebody can review it on Postgres, Mysql, Sqlite, we could integrate it.

#3 - 2010-09-23 08:42 - Jean-Baptiste Barth

- Status changed from Resolved to New

One more thing, don't use the "Resolved" state, we use it to track patches already committed in trunk that are waiting for merge in stable branch.
Thanks

#4 - 2010-09-23 09:50 - Nelzin Alexander

I have also tested this patch with PostgreSQL 8.4.4 - works fine.

#5 - 2010-09-23 09:54 - Nelzin Alexander

Redmine with Postgres info:
gem postgres-pr (0.6.3)
PostgreSQL server 8.4.4

```
~/Sites/redmine-1.0-postgres $ ruby script/about
About your application's environment
Ruby version              1.8.7 (universal-darwin10.0)
RubyGems version          1.3.5
Rack version              1.0
Rails version             2.3.5
Active Record version     2.3.5
Active Resource version   2.3.5
Action Mailer version     2.3.5
Active Support version    2.3.5
Application root          /Users/alexander_nelzin/Sites/redmine-1.0-postgres
Environment               development
Database adapter          postgresql
Database schema version   20100819172912
```

#6 - 2010-09-23 13:48 - Felix Schäfer

Jean-Baptiste Barth wrote:

> This one looks OK. If somebody can review it on Postgres, Mysql, Sqlite, we could integrate it.

Veto. I don't see what this change does other than changing how the variable is interpolated, which should have no effect whatsoever on the result.

The error the OP (original poster) reports seems to come from a missing id or root_id somewhere, which "should not happen".

Alexander: could you please look at your DB and confirm every record in the issues table has a root_id? Does this error happen on new sub-issues only, or on new issues not being sub-issues only, or on all issues?

#7 - 2010-09-23 15:16 - Jean-Baptiste Barth

I don't know if it will effectively solve Nelzin problem. But I thought it would be a good practice to systematically sanitize our conditions.

#8 - 2010-10-11 12:26 - Nelzin Alexander

I caught this error again during updating sub-issue.
Updated to latest stable Redmine 1.0.2.stable.4247 (OracleEnhanced)

```
ActiveRecord::StatementInvalid in IssuesController#update
OCIError: ORA-00936: missing expression: UPDATE "ISSUES" SET root_id = , lft = 23803, rgt = 23804 WHERE (id =
9762)

RAILS_ROOT: /Users/alexander_nelzin/Sites/redmine-1.0-dev
Application Trace | Framework Trace | Full Trace
/System/Library/Frameworks/Ruby.framework/Versions/1.8/usr/lib/ruby/gems/1.8/gems/activerecord-2.3.5/lib/activ
e_record/connection_adapters/abstract_adapter.rb:219:in `log'
/Library/Ruby/Gems/1.8/gems/activerecord-oracle_enhanced-adapter-1.3.0/lib/active_record/connection_adapters/o
racle_enhanced_adapter.rb:1726:in `log'
/Library/Ruby/Gems/1.8/gems/activerecord-oracle_enhanced-adapter-1.3.0/lib/active_record/connection_adapters/o
racle_enhanced_adapter.rb:605:in `execute'
/Users/alexander_nelzin/Sites/redmine-1.0-dev/app/models/issue.rb:631:in `update_nested_set_attributes'
/Users/alexander_nelzin/Sites/redmine-1.0-dev/app/models/issue.rb:510:in `save_issue_with_child_records'
/Users/alexander_nelzin/Sites/redmine-1.0-dev/app/models/issue.rb:492:in `save_issue_with_child_records'
/Users/alexander_nelzin/Sites/redmine-1.0-dev/app/controllers/issues_controller.rb:170:in `update'
```

The sql query is not valid.
This error occurs after migrating from redmine 0.9.4 (with Issues hierarchy plugin) to 1.x.x

Jean-Baptiste Barth, I've checked my db and no issue has root_id, they are nulls.

> The error the OP (original poster) reports seems to come from a missing id or root_id somewhere, which "should not happen".

Sanitizing sql conditions is a good practice, and helps to avoid such errors.

Btw do you have any script to convert data used by Issues hierarchy plugin to be valid in current version?

#9 - 2010-10-11 12:39 - Felix Schäfer

Nelzin Alexander wrote:

> This error occurs after migrating from redmine 0.9.4 (with Issues hierarchy plugin) to 1.x.x

That is probably the root of your problem, sanitizing the queries may alleviate it, but not solve it (especially: no one can guarantee other changes won't break stuff again later). There is an (unofficial) migration floating around on redmine.org, can't remember which issue ID it was though. Maybe search for "subtask plugin" or something similar.

> Jean-Baptiste Barth, I've checked my db and no issue has root_id, they are nulls.

Which confirms the above assumption.

> > The error the OP (original poster) reports seems to come from a missing id or root_id somewhere, which "should not happen".
>
> Sanitizing sql conditions is a good practice, and helps to avoid such errors.

I didn't say I didn't want to sanitize the SQL parameters, just that it would probably not correct the cause of your problem but only the symptom. :-) As I said, nil root_ids *should* not happen, having had the subtask plugin installed and not removed/migrated properly is one of the known causes for that.

> Btw do you have any script to convert data used by Issues hierarchy plugin to be valid in current version?

No official, because the plugin is third-party and thus not directly supported, but as I said, there is a user-supplied migration somewhere that takes care of that.

#10 - 2024-02-05 15:58 - Go MAEDA

- Status changed from New to Closed

- Resolution set to Cant reproduce

I am closing this issue because awesome_nested_set gem, the library formerly used to implement nested sets, is no longer used in Redmine. The gem was replaced with a custom implementation in Redmine 3.0.0 (#18860).