

Redmine - Patch #36416

Cleanup more dependent objects on project delete

2022-01-11 20:11 - Holger Just

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Marius BĂLTEANU	% Done:	0%
Category:	Database	Estimated time:	0.00 hour
Target version:	5.0.0		
Description On project deletion, there are currently some rows in the database which are not correctly destroyed and thus become orphaned. With this patch, we will destroy overridden time entry activities (and all of their children) of a project on project destroy. We now also destroy dependent objects of stored queries for a project on destroy. This will cleanup rows from the habtm join table queries_roles which were also orphaned before. The attached patch is against current trunk but should also apply to older versions. When we apply this, we may also want to add a migration to cleanup previously orphaned rows in the database.			
Related issues:			
Related to Redmine - Patch #36844: Cleanup orphaned query and role ids from h...		Closed	
Related to Redmine - Defect #38286: "Cannot delete enumeration" error may occ...		Closed	

Associated revisions

Revision 21437 - 2022-02-24 20:44 - Marius BĂLTEANU

Destroy overridden time entry activities and rows from the habtm join table queries_roles on project deletion (#36416).

Patch by Holger Just.

Revision 21438 - 2022-02-24 20:55 - Marius BĂLTEANU

Add migration to delete orphaned time entry activities (#36416).

Patch by Holger Just.

Revision 21443 - 2022-02-27 15:06 - Marius BĂLTEANU

Fix rubocop offense (#36416).

Revision 21444 - 2022-02-28 22:16 - Marius BĂLTEANU

Add habtm relation between roles and queries_roles (#36416).

History

#1 - 2022-01-12 00:27 - Go MAEDA

- Target version set to Candidate for next major release

#2 - 2022-01-25 18:53 - Holger Just

- File 0001-Cleanup-more-dependent-objects-on-project-delete.patch added

Attached is an updated patch which included a migration to delete potentially orphaned database rows.

#3 - 2022-01-25 18:53 - Holger Just

- File deleted (0001-Cleanup-more-dependent-objects-on-project-delete.patch)

#4 - 2022-01-30 11:41 - Marius BĂLTEANU

- Target version changed from Candidate for next major release to 5.0.0

#5 - 2022-02-24 21:03 - Marius BĂLTEANU

- Assignee set to Marius BĂLTEANU

Holger, thanks for the patches. I've committed most of the changes ([r21437](#) and [r21438](#)), but the migration for OrphanedQueriesRoles throws the below error on PostgreSQL after I changed from DELETE #{queries_roles} FROM #{queries_roles}... to DELETE FROM #{queries_roles}....

Caused by:

```
PG::SyntaxError: ERROR: syntax error at or near "LEFT"
LINE 1: DELETE FROM queries_roles LEFT OUTER JOIN queries ON queries...
```

#6 - 2022-02-24 21:19 - Marius BĂLTEANU

What do you think about the following query migration?

```
queries_roles = "#{Query.table_name_prefix}queries_roles#{Query.table_name_suffix}"
queries = Query.table_name
ActiveRecord::Base.connection.execute "DELETE FROM #{queries_roles} WHERE query_id IN (SELECT #{queries_roles}
.query_id FROM #{queries_roles} LEFT OUTER JOIN #{queries} ON #{queries_roles}.query_id = #{queries}.id WHERE
#{queries}.id IS NULL)"
```

it will generate the following query:

```
DELETE FROM queries_roles WHERE query_id IN (
  SELECT queries_roles.query_id FROM queries_roles
  LEFT OUTER JOIN queries ON queries_roles.query_id = queries.id WHERE queries.id IS NULL
)
```

#7 - 2022-02-24 21:31 - Marius BĂLTEANU

Also, it seems like the habtm join table queries_roles is not clean up when a role is destroyed and we should fix this case as well.

#8 - 2022-02-25 18:30 - Holger Just

Marius BALTEANU wrote:

What do you think about the following query migration?

[...]

This fails on MySQL (or at least older MySQL versions):

You can't specify target table 'queries_roles' for update in FROM clause

The version I have proposed works there.

Also, it seems like the habtm join table queries_roles is not clean up when a role is destroyed and we should fix this case as well.

That is true. It might be enough to add a simple habtm relation to the Role model here. If I'm not mistaken, then Rails will automatically cleanup rows from the join table on destroy.

#9 - 2022-02-28 22:58 - Marius BĂLTEANU

Holger Just wrote:

That is true. It might be enough to add a simple habtm relation to the Role model here. If I'm not mistaken, then Rails will automatically cleanup rows from the join table on destroy.

Yes, it works as you said, I've added this change in [r21444](#).

Regarding the migrations, I tested the following queries on both MySQL and PostgreSQL and it works as expected:

Orphaned by query:

```
DELETE FROM queries_roles WHERE query_id NOT IN (SELECT DISTINCT(id) FROM queries);
```

Orphaned by role:

```
DELETE FROM queries_roles WHERE role_id NOT IN (SELECT DISTINCT(id) FROM roles);
```

Migration:

```
class DeleteOrphanedQueriesRoles < ActiveRecord::Migration[6.1]
  def self.up
    queries_roles = "#{Query.table_name_prefix}queries_roles#{Query.table_name_suffix}"
    queries = Query.table_name
    roles = Role.table_name

    ActiveRecord::Base.connection.execute "DELETE FROM #{queries_roles}
    WHERE query_id NOT IN (SELECT DISTINCT(id) FROM #{queries})"
    ActiveRecord::Base.connection.execute "DELETE FROM #{queries_roles}
    WHERE role_id NOT IN (SELECT DISTINCT(id) FROM #{roles})"
  end

  def self.down
    # no-op
  end
end
```

Do you see any issue with this approach?

#10 - 2022-03-18 19:02 - Marius BĂLTEANU

Holger, did you have a chance to check my last comment?

#11 - 2022-03-24 14:47 - Holger Just

Sorry for the late reply. Your approach is a nice alternative which should probably work fine. I don't see any obvious issues there.

#12 - 2022-03-27 23:20 - Marius BĂLTEANU

- Related to Patch #36844: Cleanup orphaned query and role ids from habtm join table queries_roles added

#13 - 2022-03-27 23:21 - Marius BĂLTEANU

Holger Just wrote:

Sorry for the late reply. Your approach is a nice alternative which should probably work fine. I don't see any obvious issues there.

Thanks Holger! I've created [#36844](#) to track the migration because [5.0.0](#) will be released as soon as possible.

#14 - 2022-03-27 23:22 - Marius BĂLTEANU

- Status changed from New to Closed

#15 - 2023-03-01 15:13 - Go MAEDA

- Related to Defect #38286: "Cannot delete enumeration" error may occur when attempting to delete a project with time entries added

Files

0001-Cleanup-more-dependent-objects-on-project-delete.patch	4.02 KB	2022-01-25	Holger Just
---	---------	------------	-------------