

## Redmine - Defect #27165

### Delete\_all\_members

2017-10-11 11:53 - Enrique Vera

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Projects	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Affected version:</b>	
<b>Resolution:</b>	Invalid		
<b>Description</b>			
<p>The method delete_all_members in the model Project is the next:</p> <pre>def delete_all_members   me, mr = Member.table_name, MemberRole.table_name   self.class.connection.delete("DELETE FROM #{mr} WHERE #{mr}.member_id IN (SELECT #{me} .id FROM #{me} WHERE #{me}.project_id = #{id})")   Member.delete_all(['project_id = ?', id]) end</pre> <p>I was doing some test capturing the destroy event on a member patch and I realized that the event on this patch was only ran in some members.</p> <p>The question is, why we have to do a class.connection to delete the member_roles and then run a member.delete_all if we have connected the models correctly.</p> <p>I proved with this:</p> <pre>def delete_all_members   me, mr = Member.table_name, MemberRole.table_name   Member.destroy_all(['project_id = ?', id]) end</pre> <p>and the results were ok. Is there something that I don't know of it's a defect?</p> <p>Thanks!</p>			

### History

#### #1 - 2017-10-11 13:45 - Holger Just

When using destroy\_all (or destroy) on a model, what happens is that Rails first fetches the record from the database, destroys it and runs any defined callbacks. If you have a lot of members, this can be **very** expensive.

With the way this is done in Redmine right now, it deletes the necessary records from the database directly with two DELETE SQL statements without first fetching the records them and creating ActiveRecord model instances for each of them.

#### #2 - 2017-10-29 11:25 - Toshi MARUYAMA

- Description updated

#### #3 - 2020-03-08 13:38 - Go MAEDA

- Status changed from New to Closed

- Resolution set to Invalid

Closing this issue because this is not a defect but a question.