

TYPO3.Flow - Bug #51972

Joins for every deep property constraint make cartesian selection

2013-09-13 10:04 - Adrian Förder

Status:	New	Start date:	2013-09-13
Priority:	Should have	Due date:	
Assignee:	Adrian Förder	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Complexity:	
PHP Version:			
Has patch:	Yes		

Description

When it comes to property lookups as in

```
$query->logicalAnd(  
    $query->contains('categories.posts', $firstPost),  
    $query->equals('categories.approved', TRUE)  
)
```

there are LEFT JOINS for both constraints each, resulting in something like

```
SELECT e FROM Acme\Post e  
LEFT JOIN e.categories categories0  
LEFT JOIN e.categories categories1  
WHERE ?1 MEMBER OF categories0.posts  
AND categories1.approved = ?2
```

This, at the end, likely resulted into unpredicted matches because it would match those Posts where a category having approved = false is involved in the post in question, as long as this category belongs to an (arbitrary) Post having any category approved = true.

Consider the following testing code:

```
<?php  
public function complexQueryWithJoinsMakesCorrectConjunctions() {  
    $postEntityRepository = new PostRepository;  
  
    $notApprovedCategory = new Category();  
    $approvedCategory = new Category();  
    $approvedCategory->setApproved(TRUE);  
  
    $firstPost = new Post;  
    $firstPost->setTitle('First Post');  
    $firstPost->addCategory($notApprovedCategory);  
    $postEntityRepository->add($firstPost);  
  
    $secondPost = new Post;  
    $secondPost->setTitle('Second Post');  
    $secondPost->addCategory($notApprovedCategory);  
    $secondPost->addCategory($approvedCategory);  
    $postEntityRepository->add($secondPost);  
  
    $thirdPost = new Post;  
    $thirdPost->setTitle('Third Post');  
    $thirdPost->addCategory($approvedCategory);  
    $postEntityRepository->add($thirdPost);  
}
```

```
$this->persistenceManager->persistAll();

$query = new Query('TYPO3\FLOW\Tests\Functional\Persistence\Fixtures\Post');
$query->matching(
    $query->logicalAnd(
        $query->contains('categories.posts', $firstPost),
        $query->logicalNot($query->equals('Persistence_Object_Identifier', $this->
persistenceManager->getIdentifierByObject($firstPost))),
        $query->equals('categories.approved', TRUE)
    )
);
$this->assertEquals(0, $query->count());
}
```

The intention is to find all Posts which have the same **approved** category as the *First Post*. In the shown example, the *First Post* shares an **unapproved** category with the *Second Post*, and the *Second Post* shares an **approved** category with the *Third Post*.

The query gives a single result which is the *Second Post*, but this is not correct since the *Second Post* does only share an **unapproved** category with the *First Post* which was asked for. The *Second* only "coincidentally" also is involved in an **approved** category with the *Third Post*.

A fix is under review at <https://review.typo3.org/#/c/23751/>