Traps in inserting related objects using external id

December 16, 2014 bartoszborowiec 1 comment

Salesforce has very useful feature. You can insert related objects using external id. In this case salesforce automatically fill id of related object. Unfortunately there is a trap in this approach. Please consider following unit test:

```
1   @isTest
2   private class ExternalIdTest {
3
4       @isTest
5       static void createLookupUsingExternalIdPassTest() {
6           Master__c master = new Master__c(
7               name = 'M',
8               eid__c = 10 //external id
9           );
10
11          Servant__c servant = new Servant__c(
12              name = 'S',
13              eid__c = 111,
14              Master__r = new Master__c(eid__c = 10)
15          );
16
17          insert master;
18          insert servant;
19
20          Servant__c servantFromDb = [
21              select Id, name, master__r.name
22              from Servant__c
23              where id = :servant.Id
24          ];
25
26          System.assertEquals( 'S', servantFromDb.name);
27          System.assertEquals( 'M', servantFromDb.master__r.na
28      }
29
30      @isTest
31      static void createLookupUsingExternalIdFail1Test() {
32          Master__c master = new Master__c(
33              name = 'M',
34              eid__c = 10 // external
35          );
36
37          Servant__c servant = new Servant__c(
38              name = 'S',
39              eid__c = 111,
40              Master__r = master
41          );
42
43          insert master;
44          insert servant;
45
46          Servant__c servantFromDb = [
47              select Id, name, master__r.name
48              from Servant__c
49              where id = :servant.Id
50          ];
51
52          System.assertEquals( 'S', servantFromDb.name);
53          System.assertEquals( null , servantFromDb.master__r.
54      }
55
56
57  }
```

createLookupUsingExternalIdPassTest – is very straightforward. We create a new object (new Master__c(eId = 10)), which has only external id filled. In this case salesforce automatically creates lookup between master and servant object.

createLookupUsingExternalIdFail1Test – this example is more difficult. We did not created a new object but assign an reference that points to existing object. In this case. Object is inserted, but lookup is not created (please refer to line *System.assertEquals(null , servantFromDb.master__r.name); .* It is Quite confusing, isn't it?