EDIT - bug #8892

Taxon search does not work for names without rank

03/13/2020 01:01 PM - Katja Luther

Status:	Closed		**Start date:**	
Priority:	Highest		**Due date:**	
Assignee:	Andreas Kohlbecker		**% Done:**	100%
Category:	cdmlib		**Estimated time:**	0:00 hour
Target version:	Release 5.13			
Severity:	normal		**Found in Version:**	

Description
Searching for taxa by titleCache fails for taxa having a name without a rank. The hql query with "SELECT ... , name.rank, .. " does not return results with name.rank is null. The solution is to return the whole name and get the rank afterwards.

Related issues:	
Related to EDIT - bug #8815: Clean up TaxonNodeDto and UuidAndTitleCache usag...	**Closed**
Related to EDIT - task #8834: use TaxonNodeDto instead of UuidAndTitleCache	**Closed**
Copied to EDIT - task #8907: replace UuidAndTitleCacheTaxonComparator by type...	**Resolved**

Associated revisions

Revision 44deb132 - 03/13/2020 01:06 PM - Katja Luther

fix #8892: fix problem of name without rank in taxon search

Revision 8dd22643 - 03/13/2020 02:02 PM - Katja Luther

ref #8892: add missing comparator

Revision 919b6fe0 - 03/20/2020 07:46 PM - Andreas Kohlbecker

ref #8892 loading ranks in hql with LEFT OUTER JOINs

Revision 04825efd - 03/20/2020 07:46 PM - Andreas Kohlbecker

ref #8892 implementing test for TaxonNodeDaoHibernateImpl.getUuidAndTitleCache()

Revision 65867bfc - 03/21/2020 01:27 AM - Andreas Müller

fix #8892 fix select statement for TaxonNodeDaoHibernateImpl.getUuidAndTitleCache()

History

#1 - 03/13/2020 01:02 PM - Katja Luther

- Subject changed from taxon search does not work for names without rank to Taxon search does not work for names without rank

#2 - 03/13/2020 01:10 PM - Katja Luther

- Status changed from New to Resolved

- % Done changed from 0 to 50

Applied in changeset cdmlib|44deb132b6744b0a92b419fef363a9f98c543af0.

#3 - 03/13/2020 01:14 PM - Katja Luther

- Assignee changed from Katja Luther to Andreas Müller

please review

#4 - 03/19/2020 02:33 PM - Andreas Müller

The current solution solves the problem but contradicts the idea of UuidAndTitleCache (DTO). This idea is to not initialize any CDM objects. Already loading the rank contradicts this idea but is not so critical because the rank object is reused by different results. The name is definetly not.

We should find a solution without loading the name for performance reasons as performance is very important for searches.

#5 - 03/20/2020 02:20 PM - Andreas Müller

- Assignee changed from Andreas Müller to Andreas Kohlbecker

#6 - 03/20/2020 03:24 PM - Andreas Kohlbecker

Working on this code a couple of questions are coming up, since a couple of things are quire unclear:

1)

As last resort compare strategy UuidAndTitleCacheTaxonComparator falls back to return ((UUID)o1[0]).compareTo((UUID)o2[0]); is this by purpose, why not compare the taxonNode.id here?

2) UuidAndTitleCacheTaxonComparator is used by different methods which pass Object array with different content:

- TaxonNodeDaoHibernateImpl.getTaxonNodeDto(Integer limit, String pattern, UUID classificationUuid) ==> tn.uuid, tn.id, t.titleCache, t.name, rank
- TaxonNodeDaoHibernateImpl.getTaxonNodeUuidAndTitleCacheOfAcceptedTaxaByClassification(..) ==> nodes.uuid, nodes.id, taxon.titleCache, taxon.name, rank
- TaxonNodeDaoHibernateImpl.getUuidAndTitleCache(Integer limit, String pattern, UUID classificationUuid) ==> 'tn.uuid, tn.id, t.titleCache, name, name_rank'
- TaxonDaoHibernateImpl.getUuidAndTitleCache ==> tb.uuid, tb.id, tb.titleCache, tb.name.rank **BUG:** rank is passed instead of name here!
- IdentifiableDaoBase.getUuidAndTitleCache ==> uuid, id, titleCache **SCARY:** It seems as if this comparator has initially been used to compare arbitrary IndentifieableEntities. for my feeling this comparator is too much overloaded

Since UuidAndTitleCacheTaxonComparator is used a lot by methods it is quite unsave that it takes object arrays as parameters. We need a comparator for more specific dto class instead.

Question: Aren't all these UuidAndTitleCachemethods to be removed anyway?

Suggested solution:

- I will create specific comparator lioke SortableTaxonNodeQueryResultComparator implements Comparator<SortableTaxonNodeQueryResult> whereas SortableTaxonNodeQueryResult is a sortable dto which will never leave the persistence layer.

#7 - 03/20/2020 03:55 PM - Andreas Kohlbecker

- Related to bug #8815: Clean up TaxonNodeDto and UuidAndTitleCache usage in TaxonNodeService/Dao added

#8 - 03/20/2020 03:55 PM - Andreas Kohlbecker

- Related to task #8834: use TaxonNodeDto instead of UuidAndTitleCache added

#9 - 03/20/2020 04:46 PM - Andreas Müller

I have not checked the above but probably you are right that we should clean it up in such a way.
But I strongly recommend not to do this before the release as this is core functionality which is used at many places and therefore needs intensive testing. Also the related tickets belong to 5.14 and should be handled there.

#10 - 03/20/2020 04:47 PM - Andreas Müller

The ticket here is only about handling null values correctly which can be done without huge changes.

#11 - 03/20/2020 07:50 PM - Andreas Kohlbecker

Andreas Müller wrote:

> The ticket here is only about handling null values correctly which can be done without huge changes.

Ok ok, I don't really agree, but will stick strictly to what is demanded in the ticket description:

1. I fixed the problem
2. Implemented a test for TaxonNodeDaoHibernateImpl.getUuidAndTitleCache(), which was not covered so far (BTW: Only few of these core functionality methods are covered by tests yet.)

#12 - 03/20/2020 10:16 PM - Andreas Kohlbecker

The new test method obviously fails due to the problem described above http://int.e-taxonomy.eu/jenkins/job/cdmlib-INTEGRATION/11120/ :

```
java.lang.ClassCastException: eu.etaxonomy.cdm.model.name.TaxonName cannot be cast to eu.etaxonomy.cdm.model.name.Rank
    at eu.etaxonomy.cdm.model.taxon.UuidAndTitleCacheTaxonComparator.compare(UuidAndTitleCacheTaxonComparator.
```

```
java:39)
    at eu.etaxonomy.cdm.model.taxon.UuidAndTitleCacheTaxonComparator.compare(UuidAndTitleCacheTaxonComparator.
java:1)
 .....
```

this happens at this piece of code

```
        //Rank
        Rank rankTax1 = Rank.UNKNOWN_RANK();
        if (o1[3] != null){
            rankTax1 = (Rank)o1[3]; // <<<< ClassCastException !!!!
        }
        Rank rankTax2 = Rank.UNKNOWN_RANK();
        if (o2[3] != null){
            rankTax2 = (Rank)o2[3];
        }
```

Since this is core functionality I suggest to solve this problem instead of ignoring it.

#13 - 03/21/2020 01:39 AM - Andreas Müller

I do not really understand this. The test failed because commit 919b6fe025ff1b9 did change the number of parameters in the SELECT statements compared to the orgiginal version (name was not a parameter in the SELECT statement before). I changed this back and everything seems to work as expected. At least the tests are running.

#14 - 03/21/2020 01:52 AM - Andreas Müller

By the way, I saw that there is a new test data file TaxonNodeDaoHibernateImplTest.findWithoutRank.xml which seems to be a (more or less) copy of TaxonNodeDaoHibernateImplTest.xml.

I did not check the according test so there might be a reason why we need this new file and can't use the existing one. If there is a reason can we reduce it to the really necessary records to test *findWithoutRank*? The reason why I ask is that test datasets which are too large take time to maintain (mostly during model changes) and also the according tests are more difficult to understand if the test data is complex (and therefore the focus of the test does not become clear when you try to read it later).
Therefore we should try to create as less test data as possible and try to use as many test data as possible.
(if a new test data file is needed at all I guess all the AUD data and also data like TAXONNODE_EXCLUDEDNOTE, LANGUAGESTRING and maybe even REFERENCE is not needed. Also some fields are not needed (e.g. CREATED, UPDATED)

#15 - 03/26/2020 11:31 AM - Andreas Kohlbecker

- Copied to task #8907: replace UuidAndTitleCacheTaxonComparator by type save implementation added

#16 - 03/26/2020 11:33 AM - Andreas Kohlbecker

- Status changed from Resolved to Closed

- % Done changed from 50 to 100

remaining tasks are copied to #8907, so we can close this ticket now