# VegBank - Bug #2497

## Denorm utility incorrect in order of statements (based on tables, not sequence)

07/20/2006 03:22 PM - Michael Lee

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **Start date:** | 07/20/2006 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Chad Berkley | **% Done:** | 0% |
| **Category:** | misc | **Estimated time:** | 0.00 hour |
| **Target version:** | UnMilestoned | **Spent time:** | 0.00 hour |
| **Bugzilla-Id:** | 2497 | | |

**Description**

I think the denorm utility updates after an XML load from table-to-table. This is incorrect. It should go in the order of the numbered dnrm- statements in SQLStore, NOT BY TABLE.

MLee should check that this is indeed happening and possibly suggest a solution in the java utility.

### History

#### #1 - 07/20/2006 08:44 PM - Michael Lee

I have confirmed that this is indeed happening-- the ordering gets messed up because each table is run in the order of tables (what order of tables, I don't know). Further, it seems that they are sorted in alphabetic order, not in order of the numeric keys in SQLStore.

So, the keys in SQLStore are like so:
dnrm-[table]-[sequence #]-[human readable description]

We should run all of the denormalizations, but in the order of [sequence #] not grouping by tables. The offending code is here:
vegbank/src/java/org/vegbank/common/utility/DenormUtility.java
The method init() is called and it "sorts" by table then sequence ID. This is called by LoadTreeToDatabase method runDenorms(), which itself loops through tables, but instead should call a new method in the DenormUtility which runs through all denorm statements in order, rather than requesting each table.

One SQLStore statement is wrong, I'm removing that in SQLStore now.

The implications of this is that plots are not properly denormalized and thus are embargoed (at least partly) by default.

#### #2 - 08/17/2006 01:31 PM - Chad Berkley

the denormutility now builds a vector of all of the keys in the sqlStore.properties file and sorts them by the sort value. When a subprocess needs to run denorms, it can now just "queue" a tablename. after all of the tables it wants to denormed are queued, the execute() method is called and denormUtility first orders all of the queued tables in the order they are in in the ordered vector, then runs the denorms. This appears to be working and it makes denormalization much more effective (since now it's running the sql statements in the order in which they were designed to be run).

#### #3 - 03/27/2013 02:20 PM - Redmine Admin

Original Bugzilla ID was 2497