BIEN 3 - Bug #948

fix duplicated rows in view_full_occurrence_individual

08/25/2014 04:56 PM - Aaron Marcuse-Kubitza

Status:	Resolved	**Start date:**	08/25/2014
Priority:	High	**Due date:**	
Assignee:	Aaron Marcuse-Kubitza	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

issue

from Brody:

> SELECT * FROM view_full_occurrence_individual WHERE datasource =
> 'VegBank' AND plot_name = 'CO081202XY04'
> ;
>
> returns two lines with "Lupinus argenteus". Those lines appear to be complete duplicates, except for the last field (taxonobservation_id) which is 1230 for one and 1231 for the next. Both lines show cover_percent of 2.
>
> Looking at VegBank (http://vegbank.org/vegbank/views/observation_comprehensive.jsp?view=comprehensive&entity=observation&wparam=42558&strata2Show=1&params=42558&placeholder=), I see a single entry of Lupinus argenteus, with a cover of 2%.
>
> There's a similar situation for Quercus gambelii in this particular plot.

status

the problem is in the TNRS table, which at one point was allowing the same input name to be scrubbed multiple times (and therefore to have multiple best-match entries). the duplicate scrubbing no longer happens (make scrub does not rescrub any names), so it is just a matter of removing the existing duplicates.

fix

~~check staging tables~~

the duplication is not in the source data or the staging tables, as VegBank/CVS themselves have only one entry for this:

- VegBank: using aggregate_organism_observation_id = 937165 from those two rows:

```
SELECT * FROM "VegBank".taxonimportance WHERE taxonimportance_id = 937165; -- 1 row
SELECT * FROM "VegBank".taxoninterpretation WHERE taxonobservation_id = 697934; -- 1 row
```

- CVS: using aggregate_organism_observation_id = 396239 from two rows that have this problem:

```
SELECT * FROM "CVS"."taxonImportance" WHERE "taxonImportance_ID" = 396239; -- 1 row
SELECT * FROM "CVS"."taxonObservation_" WHERE "taxonOccurrenceID__overall_plot" = 162259;
-- 1 row
```

~~check normalized DB~~

go through all of the tables used by view_full_occurrence_individual_view (below), and figure out if any of them have duplication. those that have duplication will cause the corresponding LEFT JOIN to add duplicated rows.

using the example CVS row above:

FROM source
JOIN location USING (source_id)
~~LEFT JOIN locationevent USING (location_id)~~*query:*

```
SELECT * FROM locationevent WHERE locationevent_id = 912711; -- 1 row
```

LEFT JOIN place USING (place_id)
LEFT JOIN location parent_location ON parent_location.location_id = location.parent_id
LEFT JOIN coordinates USING (coordinates_id)
LEFT JOIN geoscrub.geoscrub_output ON ARRAY[geoscrub_output."decimalLatitude"] = ARRAY[coordinates.latitude_deg] AND ARRAY[geoscrub_output."decimalLongitude"] = ARRAY[coordinates.longitude_deg] AND ARRAY[geoscrub_output.country] = ARRAY[place.country] AND ARRAY[geoscrub_output."stateProvince"] = ARRAY[place.stateprovince] AND ARRAY[geoscrub_output.county] = ARRAY[place.county]
LEFT JOIN "newWorld".iso_code_gadm ON iso_code_gadm."*GADM country" = COALESCE(geoscrub_output."acceptedCountry", place.country)
LEFT JOIN "newWorld"."newWorldCountries" ON "newWorldCountries"."*isoCode" = iso_code_gadm."*2-digit iso code"
LEFT JOIN geoscrub.county_centroids ON place.country = 'United States'::text AND county_centroids."stateProvince" = COALESCE(geoscrub_output."acceptedStateProvince", place.stateprovince) AND county_centroids.county = COALESCE(geoscrub_output."acceptedCounty", place.county)
LEFT JOIN locationevent parent_event ON parent_event.locationevent_id = locationevent.parent_id
LEFT JOIN project ON project.project_id = COALESCE(locationevent.project_id, parent_event.project_id)
LEFT JOIN stratum ON stratum.stratum_id = COALESCE(locationevent.stratum_id, parent_event.stratum_id)
LEFT JOIN method ON method.method_id = COALESCE(locationevent.method_id, parent_event.method_id);
~~LEFT JOIN taxonoccurrence ON taxonoccurrence.locationevent_id = "plot.**".locationevent_id~~*query:*

```
SELECT * FROM taxonoccurrence WHERE taxonoccurrence_id = 11298823; -- 1 row
```

LEFT JOIN party collector ON collector.party_id = taxonoccurrence.collector_id
~~LEFT JOIN aggregateoccurrence USING (taxonoccurrence_id)~~*query:*

```
SELECT * FROM aggregateoccurrence WHERE aggregateoccurrence_id = 11292621; -- 1 row
```

~~LEFT JOIN plantobservation USING (aggregateoccurrence_id)~~*query:*

```
SELECT * FROM plantobservation WHERE plantobservation_id = 11865135; -- 1 row
```

LEFT JOIN specimenreplicate USING (plantobservation_id)
LEFT JOIN sourcelist ON sourcelist.sourcelist_id = specimenreplicate.duplicate_institutions_sourcelist_id
~~LEFT JOIN taxondetermination ON taxondetermination.taxonoccurrence_id = taxonoccurrence.taxonoccurrence_id AND taxondetermination.iscurrent~~*query:*

```
SELECT * FROM taxondetermination WHERE taxonoccurrence_id = 11298823; -- 1 row
```

LEFT JOIN party identifiedby ON identifiedby.party_id = taxondetermination.party_id
~~LEFT JOIN taxonverbatim USING (taxonverbatim_id)~~*query:*

```
SELECT * FROM taxonverbatim WHERE taxonverbatim_id = 830666; -- 1 row
```

~~LEFT JOIN taxonlabel USING (taxonlabel_id)~~*query:*

```
SELECT * FROM taxonlabel WHERE taxonlabel_id = 1534892; -- 1 row
```

bug here:
~~LEFT JOIN "TNRS".taxon_scrub ON taxon_scrub."*Name_submitted" = taxonlabel.taxonomicname~~*query:*

```
SELECT * FROM "TNRS".taxon_scrub WHERE "*Name_submitted" =
'Grossulariaceae Ribes rotundifolium Michx.'; -- 2 rows
```

LEFT JOIN family_higher_plant_group ON family_higher_plant_group.family =

taxon_scrub."[scrubbed_]family~(-Accepted_-)__@TNRS__@vegpath.org"
LEFT JOIN cultivated_family_locations ON cultivated_family_locations.family =
taxon_scrub."[scrubbed_]family~(-Accepted_-)__@TNRS__@vegpath.org" AND cultivated_family_locations.country =
"plot.**".country;

using the example VegBank row above:

FROM source
JOIN location USING (source_id)
LEFT JOIN locationevent USING (location_id)
LEFT JOIN place USING (place_id)
LEFT JOIN location parent_location ON parent_location.location_id = location.parent_id
LEFT JOIN coordinates USING (coordinates_id)
LEFT JOIN geoscrub.geoscrub_output ON ARRAY[geoscrub_output."decimalLatitude"] = ARRAY[coordinates.latitude_deg] AND
ARRAY[geoscrub_output."decimalLongitude"] = ARRAY[coordinates.longitude_deg] AND ARRAY[geoscrub_output.country] =
ARRAY[place.country] AND ARRAY[geoscrub_output."stateProvince"] = ARRAY[place.stateprovince] AND
ARRAY[geoscrub_output.county] = ARRAY[place.county]
LEFT JOIN "newWorld".iso_code_gadm ON iso_code_gadm."*GADM country" = COALESCE(geoscrub_output."acceptedCountry",
place.country)
LEFT JOIN "newWorld"."newWorldCountries" ON "newWorldCountries"."*isoCode" = iso_code_gadm."*2-digit iso code"
LEFT JOIN geoscrub.county_centroids ON place.country = 'United States'::text AND county_centroids."stateProvince" =
COALESCE(geoscrub_output."acceptedStateProvince", place.stateprovince) AND county_centroids.county =
COALESCE(geoscrub_output."acceptedCounty", place.county)
LEFT JOIN locationevent parent_event ON parent_event.locationevent_id = locationevent.parent_id
LEFT JOIN project ON project.project_id = COALESCE(locationevent.project_id, parent_event.project_id)
LEFT JOIN stratum ON stratum.stratum_id = COALESCE(locationevent.stratum_id, parent_event.stratum_id)
LEFT JOIN method ON method.method_id = COALESCE(locationevent.method_id, parent_event.method_id);
LEFT JOIN taxonoccurrence ON taxonoccurrence.locationevent_id = "plot.**".locationevent_id
LEFT JOIN party collector ON collector.party_id = taxonoccurrence.collector_id
~~LEFT JOIN aggregateoccurrence USING (taxonoccurrence_id)~~*query:*

```
SELECT * FROM aggregateoccurrence WHERE (source_id, (COALESCE(sourceaccessioncode, '\N'::text))
COLLATE pg_catalog."default") = (38,'937165') AND sourceaccessioncode IS NOT NULL; -- 1 row
```

~~LEFT JOIN plantobservation USING (aggregateoccurrence_id)~~*query:*

```
SELECT * FROM plantobservation WHERE aggregateoccurrence_id = 3823285; -- 1 row
```

~~LEFT JOIN specimenreplicate USING (plantobservation_id)~~*query:*

```
SELECT * FROM specimenreplicate WHERE plantobservation_id = 4462528; -- 1 row
```

LEFT JOIN sourcelist ON sourcelist.sourcelist_id = specimenreplicate.duplicate_institutions_sourcelist_id
LEFT JOIN taxondetermination ON taxondetermination.taxonoccurrence_id = taxonoccurrence.taxonoccurrence_id AND
taxondetermination.iscurrent
LEFT JOIN party identifiedby ON identifiedby.party_id = taxondetermination.party_id
LEFT JOIN taxonverbatim USING (taxonverbatim_id)
LEFT JOIN taxonlabel USING (taxonlabel_id)
LEFT JOIN "TNRS".taxon_scrub ON taxon_scrub."*Name_submitted" = taxonlabel.taxonomicname
LEFT JOIN family_higher_plant_group ON family_higher_plant_group.family =
taxon_scrub."[scrubbed_]family~(-Accepted_-)__@TNRS__@vegpath.org"
LEFT JOIN cultivated_family_locations ON cultivated_family_locations.family =
taxon_scrub."[scrubbed_]family~(-Accepted_-)__@TNRS__@vegpath.org" AND cultivated_family_locations.country =
"plot.**".country;

~~remove existing duplicates~~

~~review duplicates:~~

434,880 names, all with exactly 2 batches spaced seconds apart

```
SET enable_seqscan = off;
SELECT "*Name_submitted", array_agg(DISTINCT batch)
FROM "TNRS".taxon_match
```

```
GROUP BY "*Name_submitted"
HAVING COUNT(DISTINCT batch) > 1
-- runtime: 6.5 min ("383793 ms")
```

~~remove duplicates:~~

```
SET enable_seqscan = off;
DELETE FROM "TNRS".taxon_match WHERE (batch, "*Name_submitted") IN (
SELECT max(batch), "*Name_submitted"
FROM "TNRS".taxon_match
GROUP BY "*Name_submitted"
HAVING COUNT(DISTINCT batch) > 1
);
-- runtime: 2.5 min ("834755 rows affected, 149336 ms") (note rows affected > # names because mult
iple matches/name)
```

History

#1 - 08/25/2014 05:05 PM - Aaron Marcuse-Kubitza

- Description updated

#2 - 08/29/2014 01:08 AM - Aaron Marcuse-Kubitza

- Description updated

#3 - 08/29/2014 01:08 AM - Aaron Marcuse-Kubitza

- Priority changed from Normal to High

#4 - 08/29/2014 01:35 PM - Aaron Marcuse-Kubitza

- Description updated

#5 - 08/29/2014 01:38 PM - Aaron Marcuse-Kubitza

- Description updated

#6 - 08/29/2014 01:39 PM - Aaron Marcuse-Kubitza

- Description updated

#7 - 08/29/2014 02:09 PM - Aaron Marcuse-Kubitza

- Description updated

#8 - 08/29/2014 02:20 PM - Aaron Marcuse-Kubitza

- Description updated

#9 - 08/29/2014 02:52 PM - Aaron Marcuse-Kubitza

- Description updated

#10 - 08/29/2014 02:53 PM - Aaron Marcuse-Kubitza

- Description updated

#11 - 08/29/2014 02:56 PM - Aaron Marcuse-Kubitza

- Description updated

#12 - 08/29/2014 02:58 PM - Aaron Marcuse-Kubitza

- Description updated

#13 - 08/29/2014 03:14 PM - Aaron Marcuse-Kubitza

- Description updated

#14 - 08/29/2014 03:21 PM - Aaron Marcuse-Kubitza

- Description updated

- % Done changed from 0 to 50

#15 - 08/29/2014 04:07 PM - Aaron Marcuse-Kubitza

- Description updated

- % Done changed from 50 to 80

#16 - 09/04/2014 08:38 AM - Aaron Marcuse-Kubitza

- Description updated

#17 - 09/04/2014 08:40 AM - Aaron Marcuse-Kubitza

- Description updated

#18 - 09/04/2014 08:43 AM - Aaron Marcuse-Kubitza

- Description updated

#19 - 09/04/2014 08:51 AM - Aaron Marcuse-Kubitza

- Description updated

#20 - 09/04/2014 08:55 AM - Aaron Marcuse-Kubitza

- Description updated

#21 - 09/04/2014 09:00 AM - Aaron Marcuse-Kubitza

- Description updated

- % Done changed from 80 to 100

#22 - 09/04/2014 10:04 AM - Aaron Marcuse-Kubitza

- Status changed from New to Resolved