

EDIT - task #6753

Edaphobase data issues before import

06/24/2017 07:26 PM - Andreas Müller

Status:	Rejected	Start date:	
Priority:	Highest	Due date:	
Assignee:	Andreas Müller	% Done:	40%
Category:	data	Estimated time:	0:00 hour
Target version:	Reviewed Next Major Release		
Severity:	normal		
Description			
There are a couple of data that need to be checked before a final edaphobase import to avoid inconsistent or missing data.			

History

#1 - 06/24/2017 07:29 PM - Andreas Müller

Synonymrelationen mit fehlendem b_taxon (oder a_taxon). Laut Stephan ist das Datenmüll und kann vernachlässigt werden. Es macht aber Sinn ihn am Anfang der Anfragen aufzuräumen:

```
SELECT *
FROM tax_synonym
WHERE b_taxon_fk_taxon_id IS NULL OR a_taxon_fk_taxon_id IS NULL
```

derzeit 0 (15,2,3) DS

#2 - 06/24/2017 07:32 PM - Andreas Müller

Duplikate in den Synonymrelationen

```
SELECT remark, synonym_role_fk, document_fk, a_taxon_fk_taxon_id, b_taxon_fk_taxon_id, deleted, synonym_role,
count(*) as n
FROM tax_synonym sr
GROUP BY remark, synonym_role_fk, document_fk, a_taxon_fk_taxon_id, b_taxon_fk_taxon_id, deleted, synonym_role
Having count(*) > 1
ORDER BY sr.a_taxon_fk_taxon_id
```

evtl. auch nur mit den Kernfeldern

```
SELECT synonym_role_fk, a_taxon_fk_taxon_id, b_taxon_fk_taxon_id, synonym_role, count(*) as n
FROM tax_synonym sr
GROUP BY synonym_role_fk, a_taxon_fk_taxon_id, b_taxon_fk_taxon_id, synonym_role
Having count(*) > 1
ORDER BY sr.a_taxon_fk_taxon_id
```

derzeit 0 (1) DS.

#3 - 06/24/2017 07:35 PM - Andreas Müller

Synonymrelationen zwischen 2 invaliden Taxa, bei denen das b-Taxon nicht wieder Teil einer weiteren Synonym Relation ist und bei denen das a-Taxon nicht Teil einer weiteren Synonymbeziehung zu einem dann validen Taxon ist

```
SELECT sr.synonym_role, s.parent_taxon_fk synParent, t.parent_taxon_fk accParent,
s.tax_rank_fk synRank, t.tax_rank_fk accRank, p.tax_rank_fk parRank, s.name synName, t.name syn2Name, p.name
parName, s.taxon_id synId, t.taxon_id syn2Id, t2.taxon_id validId,
p.valid parValid, p2.valid par2Valid, p3.valid par3Valid,
sr2.synonym_role, t2.valid t2Valid, p.display_string par, p2.display_string par2, s.display_string syn, t.
display_string syn2, t2.display_string acc
FROM tax_taxon s
INNER JOIN tax_synonym sr ON s.taxon_id = sr.a_taxon_fk_taxon_id
INNER JOIN tax_taxon t ON t.taxon_id = sr.b_taxon_fk_taxon_id
LEFT JOIN tax_taxon p ON s.parent_taxon_fk = p.taxon_id
LEFT JOIN tax_taxon p2 ON t.parent_taxon_fk = p2.taxon_id
LEFT OUTER JOIN tax_synonym sr2 ON t.taxon_id = sr2.a_taxon_fk_taxon_id
LEFT JOIN tax_taxon t2 ON t2.taxon_id = sr2.b_taxon_fk_taxon_id
LEFT JOIN tax_taxon p3 ON t2.parent_taxon_fk = p3.taxon_id
WHERE s.valid = false AND t.valid = false AND sr.synonym_role = 11614 AND t2.taxon_id IS NULL
AND s.taxon_id NOT IN (
```

```
SELECT srsr.a_taxon_fk_taxon_id
FROM tax_synonym srsr INNER JOIN tax_taxon tt ON tt.taxon_id = srsr.b_taxon_fk_taxon_id
WHERE tt.valid = true
```

)

derzeit 0(6, 4) DS

#4 - 06/24/2017 07:38 PM - Andreas Müller

Synonymrelationen zwischen invaliden Taxa, bei denen das a-Taxon keine weitere Beziehung zu einem validen Taxon hat, aber das b-Taxon eine Beziehung zu einem validen Taxon hat.

```
SELECT sr.synonym_role, s.parent_taxon_fk synParent, t.parent_taxon_fk accParent,
s.tax_rank_fk synRank, t.tax_rank_fk accRank, p.tax_rank_fk parRank, s.name synName, t.name syn2Name, p.name
parName, s.taxon_id synId, t.taxon_id syn2Id, t2.taxon_id validId,
p.valid parValid, p2.valid par2Valid, p3.valid par3Valid,
sr2.synonym_role, t2.valid t2Valid, p.display_string par, p2.display_string par2, s.display_string syn, t.
display_string syn2, t2.display_string acc
FROM tax_taxon s
INNER JOIN tax_synonym sr ON s.taxon_id = sr.a_taxon_fk_taxon_id
INNER JOIN tax_taxon t ON t.taxon_id = sr.b_taxon_fk_taxon_id
LEFT JOIN tax_taxon p ON s.parent_taxon_fk = p.taxon_id
LEFT JOIN tax_taxon p2 ON t.parent_taxon_fk = p2.taxon_id
LEFT OUTER JOIN tax_synonym sr2 ON t.taxon_id = sr2.a_taxon_fk_taxon_id
LEFT JOIN tax_taxon t2 ON t2.taxon_id = sr2.b_taxon_fk_taxon_id
LEFT JOIN tax_taxon p3 ON t2.parent_taxon_fk = p3.taxon_id
WHERE s.valid = false AND t.valid = false AND sr.synonym_role = 11614 AND t2.taxon_id IS NOT NULL
AND s.taxon_id NOT IN (
SELECT srsr.a_taxon_fk_taxon_id
FROM tax_synonym srsr INNER JOIN tax_taxon tt ON tt.taxon_id = srsr.b_taxon_fk_taxon_id
WHERE tt.valid = true
```

)

ORDER BY t2.name

derzeit 1(2,0) DS

#5 - 06/24/2017 07:41 PM - Andreas Müller

Synonymbeziehungen mit einem validen Taxon auf der a-Seite und einer "is synonym of" Beziehung

```
SELECT s.valid, t.valid, synonym_role, s.display_string syn, t.display_string syn2
FROM tax_taxon s
INNER JOIN tax_synonym sr ON s.taxon_id = sr.a_taxon_fk_taxon_id
INNER JOIN tax_taxon t ON t.taxon_id = sr.b_taxon_fk_taxon_id
WHERE s.valid = true AND sr.synonym_role = 11614
ORDER BY s.valid
```

derzeit 1(4,0) DS

#6 - 06/24/2017 07:42 PM - Andreas Müller

Valide Kindtaxa mit invaliden parents. (passen nicht in den Baum der validen Taxa)

a) Parent ist Synonym in Synonymbeziehung:

```
SELECT c.taxon_id, c.rank_fk, c.name, c.display_string, p.display_string, c.path_to_root, c.valid,
p.taxon_id, p.rank_fk, p.name, p.path_to_root, p.valid, sr.synonym_role
FROM tax_taxon c
INNER JOIN tax_taxon p ON c.parent_taxon_fk = p.taxon_id
INNER JOIN tax_synonym sr ON p.taxon_id = sr.a_taxon_fk_taxon_id
WHERE p.valid = false AND c.valid = true
```

derzeit 13(3,4,9) DS

b) Parent ist akzeptiertes Taxon in Synonymbeziehung:

wie oben aber mit p.taxon_id = sr.b_taxon_fk_taxon_id

0 DS

c) parent gar nicht in Synonymbeziehung

```
SELECT c.taxon_id, c.rank_fk, c.name, c.display_string, p.display_string, c.path_to_root, c.valid,
p.taxon_id, p.rank_fk, p.name, p.path_to_root, p.valid
FROM tax_taxon c
```

```

INNER JOIN tax_taxon p ON c.parent_taxon_fk = p.taxon_id
WHERE p.valid = false AND c.valid = true
      AND p.taxon_id NOT IN (
          SELECT a_taxon_fk_taxon_id FROM tax_synonym WHERE a_taxon_fk_taxon_id IS NOT NULL
      )
      AND p.taxon_id NOT IN (
          SELECT b_taxon_fk_taxon_id FROM tax_synonym WHERE b_taxon_fk_taxon_id IS NOT
NULL )

```

0 DS

#7 - 06/24/2017 10:16 PM - Andreas Müller

Invalide Kinder (> Species) und mit validem Parent, aber nicht in einer Synonymrelation:

```

SELECT c.taxon_id childId, c.tax_rank_fk childRank, c.name childName, c.display_string childFull, p.
display_string parFull,
      c.path_to_root childPath, c.valid childValid,
      p.taxon_id parId, p.tax_rank_fk parentRank, p.name parName, p.path_to_root parPath, p.valid parValid
FROM tax_taxon c
      INNER JOIN tax_taxon p ON c.parent_taxon_fk = p.taxon_id
WHERE p.valid = true AND c.valid = false
      AND c.tax_rank_fk <= 11631
      AND c.taxon_id NOT IN (
          SELECT a_taxon_fk_taxon_id FROM tax_synonym WHERE a_taxon_fk_taxon_id IS NOT NULL )
      AND c.taxon_id NOT IN (
          SELECT b_taxon_fk_taxon_id FROM tax_synonym WHERE b_taxon_fk_taxon_id IS NOT NULL )

```

derzeit 0(11,1) DS

#8 - 06/24/2017 10:19 PM - Andreas Müller

Invalide Kinder mit validem Parent, die nicht in einer Synonymbeziehung sind und kein eigenes Kind haben (unabhängig vom Rang):

```

SELECT c.taxon_id childId, c.tax_rank_fk childRank, c.name childName, c.display_string childFull, c.
path_to_root childPath, c.valid childValid,
      p.taxon_id parId, p.tax_rank_fk parRank, p.name parName, p.display_string parFull, p.path_to_root parPath, p
.valid parValid
FROM tax_taxon c
      INNER JOIN tax_taxon p ON c.parent_taxon_fk = p.taxon_id
WHERE p.valid = true AND c.valid = false
      AND c.taxon_id NOT IN (
          SELECT a_taxon_fk_taxon_id FROM tax_synonym WHERE a_taxon_fk_taxon_id IS NOT NULL )
      AND c.taxon_id NOT IN (
          SELECT b_taxon_fk_taxon_id FROM tax_synonym WHERE b_taxon_fk_taxon_id IS NOT NULL )
      AND c.taxon_id NOT IN (SELECT parent_taxon_fk FROM tax_taxon cc WHERE parent_taxon_fk IS NOT NULL)
ORDER BY c.tax_rank_fk, c.name

```

derzeit 0(986,6) DS

#9 - 06/24/2017 10:41 PM - Andreas Müller

- Status changed from New to In Progress

- Priority changed from New to Highest

Invalide Kinder mit validen Parents, mit invaliden Kindeskindern, bei denen weder Kinder noch Kindeskindern in Synonymrelation sind.

```

SELECT c.taxon_id childId, c.tax_rank_fk childRank, c.name childName, c.display_string childFull, c.
path_to_root childPath, c.valid childValid,
      p.taxon_id parId, p.tax_rank_fk parRank, p.name parName, p.display_string parFull, p.path_to_root parPath, p
.valid parValid
FROM tax_taxon c
      INNER JOIN tax_taxon p ON c.parent_taxon_fk = p.taxon_id
WHERE p.valid = true AND c.valid = false
      AND c.taxon_id NOT IN (
          SELECT a_taxon_fk_taxon_id FROM tax_synonym WHERE a_taxon_fk_taxon_id IS NOT NULL )
      AND c.taxon_id NOT IN (
          SELECT b_taxon_fk_taxon_id FROM tax_synonym WHERE b_taxon_fk_taxon_id IS NOT NULL )
      AND c.taxon_id IN (SELECT parent_taxon_fk FROM tax_taxon cc WHERE cc.valid = false
          AND cc.taxon_id NOT IN (
              SELECT a_taxon_fk_taxon_id FROM tax_synonym WHERE a_taxon_fk_taxon_id IS NOT NULL )
          AND cc.taxon_id NOT IN (
              SELECT b_taxon_fk_taxon_id FROM tax_synonym WHERE b_taxon_fk_taxon_id IS NOT NULL )
          AND cc.taxon_id NOT IN (SELECT parent_taxon_fk FROM tax_taxon ccc WHERE parent_taxon_fk IS NOT NULL)

```

```
)  
ORDER BY c.tax_rank_fk, c.name
```

derzeit 0 (34) DS

vermutlich auch abgedeckt durch nächsten Query

#10 - 06/24/2017 10:45 PM - Andreas Müller

Kind und Parent invalid, Kind nicht in Synonymrelation

```
SELECT c.taxon_id, c.tax_rank_fk childRank, c.name,c.display_string, p.display_string, c.path_to_root, c.  
valid,  
      p.taxon_id, p.tax_rank_fk parentRank, p.name, p.path_to_root, p.valid  
FROM tax_taxon c  
  INNER JOIN tax_taxon p ON c.parent_taxon_fk = p.taxon_id  
WHERE p.valid = false AND c.valid = false  
  AND c.taxon_id NOT IN (  
    SELECT a_taxon_fk_taxon_id FROM tax_synonym WHERE a_taxon_fk_taxon_id IS NOT NULL )  
  AND c.taxon_id NOT IN (  
    SELECT b_taxon_fk_taxon_id FROM tax_synonym WHERE b_taxon_fk_taxon_id IS NOT NULL )  
/* AND c.taxon_id NOT IN (SELECT parent_taxon_fk FROM tax_taxon cc WHERE parent_taxon_fk IS NOT NULL) */  
ORDER BY c.tax_rank_fk
```

derzeit 0(370,2) DS

#11 - 06/24/2017 11:00 PM - Andreas Müller

Invalide Taxa mit mehreren akzeptierten Taxa:

```
SELECT s.taxon_id synId, t.taxon_id accId, s.display_string syn, t.display_string syn2  
FROM tax_taxon s  
INNER JOIN tax_synonym sr ON s.taxon_id = sr.a_taxon_fk_taxon_id  
INNER JOIN tax_taxon t ON t.taxon_id = sr.b_taxon_fk_taxon_id  
WHERE s.valid = false AND t.valid = true AND sr.synonym_role = 11614 AND EXISTS (  
  SELECT s2.taxon_id synId  
  FROM tax_taxon s2  
  INNER JOIN tax_synonym sr2 ON s2.taxon_id = sr2.a_taxon_fk_taxon_id  
  INNER JOIN tax_taxon t2 ON t2.taxon_id = sr2.b_taxon_fk_taxon_id  
  WHERE s2.valid = false AND t2.valid = true AND sr2.synonym_role = 11614  
    AND s.taxon_id = s2.taxon_id AND t.taxon_id <> t2.taxon_id  
)  
ORDER BY s.display_string, s.taxon_id
```

3(0)

#12 - 06/24/2017 11:04 PM - Andreas Müller

display_string Duplikate

```
SELECT display_string, count(*) as n  
FROM tax_taxon  
GROUP BY display_string  
HAVING count(*) > 1  
ORDER BY n DESC, display_string
```

derzeit 16 (was 8 + 2 incertis sedis, 2, 4)

#13 - 06/24/2017 11:06 PM - Andreas Müller

wie oben, aber ein Taxon mit fehlendem Autor und/oder Jahreszahl, bisheriger Query war zu inperformant => machen wir im CDM, siehe #6904

#14 - 07/06/2017 10:52 PM - Andreas Müller

- Target version changed from Release 4.8 to Release 4.9

#15 - 07/21/2017 10:13 AM - Andreas Müller

- Target version changed from Release 4.9 to Release 4.10

#16 - 07/25/2017 04:09 PM - Andreas Müller

- Tracker changed from task to bug

#17 - 07/25/2017 04:09 PM - Andreas Müller

- Tracker changed from bug to task

#18 - 08/12/2017 02:13 PM - Andreas Müller

Taxa mit Autor aber ohne Jahreszahl

```
SELECT *
FROM tax_taxon
WHERE tax_year is NULL AND tax_author_name IS NOT NULL
```

596(623) DS

und andersherum:

```
SELECT *
FROM tax_taxon
WHERE tax_year is Not NULL AND tax_author_name IS NULL
```

0 (2) DS

#19 - 08/12/2017 02:47 PM - Andreas Müller

Incorrect usage of name_addition. Das Feld wird häufig für Autoren verwendet. Oder für infraspezifische Epithete: "[var.] alpinus", Weitere eher singuläre Fälle sind: "temp", "nec Börner, 1902", "unavailable name", "aciculatus", "cf. sensu Weigmann 2006", "ad partem (Nearctis)", "i.L.", "non Wahlgren, 1906", "cf."

```
SELECT name_addition, count(*) as n
FROM tax_taxon
WHERE name_addition is NOT NULL AND name_addition <> ''
GROUP BY name_addition
ORDER BY n DESC
```

evtl. gültig: nomen nudum, sensu lato, nomen dubium, sensu stricto, sensu {Autor}{(,){Jahr}, "Fossil incerta sedis", "(nomen oblitum)", "incertae sedis", "auct. nec Reuter, 1891"

373

#20 - 08/12/2017 11:23 PM - Andreas Müller

Autoren mit " in "

```
SELECT *
FROM tax_taxon
WHERE tax_author_name like '% in %'
```

157/71 distinkte DS

#21 - 09/24/2017 12:54 AM - Andreas Müller

- Target version changed from Release 4.10 to Release 4.11

#22 - 11/08/2017 11:27 AM - Andreas Müller

- Target version changed from Release 4.11 to Release 4.12

#23 - 12/05/2017 02:08 PM - Andreas Müller

- Target version changed from Release 4.12 to Release 4.13

#24 - 01/31/2018 12:49 PM - Andreas Müller

- Target version changed from Release 4.13 to Release 4.14

#25 - 02/15/2018 03:38 PM - Andreas Müller

- Target version changed from Release 4.14 to Release 5.0

#26 - 03/06/2018 12:01 PM - Andreas Müller

Taxa mit Rolle A in "included in" und "is synonym of" Beziehung. Beide valid=f ist evtl. nicht kritisch.

```
SELECT sr.tax_synonym_id , s.valid a_valid, t.valid b_valid, synonym_role, s.display_string a, t.
display_string b, s.taxon_id taxon_a_id, t.taxon_id taxon_b_id
FROM tax_taxon s
INNER JOIN tax_synonym sr ON s.taxon_id = sr.a_taxon_fk_taxon_id
```

```

INNER JOIN tax_taxon t ON t.taxon_id = sr.b_taxon_fk_taxon_id
WHERE sr.a_taxon_fk_taxon_id IN (SELECT sr.a_taxon_fk_taxon_id
FROM tax_taxon s
INNER JOIN tax_synonym sr ON s.taxon_id = sr.a_taxon_fk_taxon_id
INNER JOIN tax_taxon t ON t.taxon_id = sr.b_taxon_fk_taxon_id
WHERE sr.synonym_role = 11613 AND EXISTS (
SELECT s2.taxon_id synId
FROM tax_taxon s2
INNER JOIN tax_synonym sr2 ON s2.taxon_id = sr2.a_taxon_fk_taxon_id
INNER JOIN tax_taxon t2 ON t2.taxon_id = sr2.b_taxon_fk_taxon_id
WHERE sr2.synonym_role = 11614
AND s.taxon_id = s2.taxon_id AND t.taxon_id <> t2.taxon_id
)
)
ORDER BY s.taxon_id, synonym_role

```

2x10 DS

siehe auch [#6137#note-34](#)

#27 - 03/06/2018 12:02 PM - Andreas Müller

- % Done changed from 10 to 40

#28 - 03/07/2018 01:47 PM - Andreas Müller

included_in relationships with an invalid B taxon

```

SELECT s.valid, t.valid, synonym_role, s.display_string A, t.display_string B,
s.taxon_id synId, t.taxon_id accId
FROM tax_taxon s
INNER JOIN tax_synonym sr ON s.taxon_id = sr.a_taxon_fk_taxon_id
INNER JOIN tax_taxon t ON t.taxon_id = sr.b_taxon_fk_taxon_id
WHERE t.valid = false AND sr.synonym_role = 11613
ORDER BY s.display_string, s.taxon_id;

```

12 DS

#29 - 04/13/2018 02:02 PM - Andreas Müller

Synonym relationships with deleted taxa

```

SELECT
sr.tax_synonym_id, sr.deleted, a.taxon_id, a.valid, a.deleted, b.taxon_id, b.valid , b.deleted
FROM
public.tax_synonym sr INNER JOIN
public.tax_taxon a ON sr.a_taxon_fk_taxon_id = a.taxon_id
INNER JOIN public.tax_taxon b ON sr.b_taxon_fk_taxon_id = b.taxon_id
WHERE
(a.deleted = true OR b.deleted = true) AND sr.deleted = false

```

13 DS

#30 - 05/14/2018 11:34 PM - Andreas Müller

- Target version changed from Release 5.0 to Release 5.1

#31 - 06/27/2018 02:31 PM - Andreas Müller

- Target version changed from Release 5.1 to Release 5.2

#32 - 08/17/2018 04:04 PM - Andreas Müller

- Target version changed from Release 5.2 to Release 5.3

#33 - 09/06/2018 10:06 AM - Andreas Müller

- Target version changed from Release 5.3 to Release 5.5

#34 - 02/14/2019 10:38 PM - Andreas Müller

- Target version changed from Release 5.5 to Release 5.6

#35 - 02/19/2019 05:12 PM - Andreas Müller

- Target version changed from Release 5.6 to Reviewed Next Major Release

#36 - 01/17/2024 12:44 PM - Andreas Müller

- Tags set to *edapho*

- Status changed from *In Progress* to *Rejected*

Edaphobase is not developed further anymore. So we won't fix.