

Admin Dokumentation: PostgreSQL Snippets

Recherche

Nachfolger in aktueller Kartierung finden

Suche nach Giscode eines alten Biotops in Kartierobjekte (Vorgänger oder Zusammengefasste Bögen) bzw. Verlustbögen.

```
SELECT * FROM mvbio.kartierobjekte
WHERE '0503-244B5099' IN (SELECT UNNEST(zusammengefasste_boegen) UNION
SELECT alt_giscod)
```

bzw. in Verlustbögen:

```
SELECT * FROM mvbio.verlustobjekte
WHERE bogen_id IN (SELECT id FROM archiv.erfassungsboegen WHERE
giscod='0503-244B4029')
```

Stand Nov. 2022, Historisierung soll geändert werden auf bogen_id statt Giscod, dann Anpassung nötig

Artenvorkommen

Suche nach Biotopen/Bögen, die eine bestimmte Art (Beispiel hier: Krebssschere *Stratiotes aloides*) enthalten. Mit Angabe der Häufigkeit (D/Z/V) und Geometriespalte (Well-Known-Text WKT, einfacher Import in QGIS möglich). Anpassung der Art in letzter Zeile, auch nur Gattung möglich (Artnamen werden mit ausgegeben).

Aktuelle Kartierung

Ohne Einschränkung der Bearbeitungsstufe!

```
SELECT
  gsl.valid_name AS art_gsl, ko.id AS id_mvbio, ko.label, ko.biotopname,
  ko.vegeinheit,
  ko.hc, COALESCE(ko.uc1, '-') || ' ' || COALESCE(ko.uc2, '-') AS ucs,
  ko.e_datum AS datum, ko.kartierer, pv.dzv AS dzv, kk.bezeichnung AS
  quelle,
  ST_AsEWKT(ko.geom) AS geom_wkt
FROM mvbio.kartierobjekte ko
JOIN mvbio.pflanzenvorkommen pv ON pv.kartierung_id = ko.id
LEFT JOIN mvbio.kampagnen kk ON kk.id = ko.kampagne_id
LEFT JOIN mvbio.pflanzenarten_gsl gsl ON gsl.valid_nr = pv.valid_nr
```

WHERE

```
gsl.valid_name LIKE 'Stratiotes%' AND gsl.synonym='0' AND kk.art IN ('BK', 'LRT', 'BLRTK')
```

Archivbestände

nur aktuelle Bögen

SELECT

```
gsl.valid_name AS art_gsl, eb.id AS id_archiv, eb.label, eb.biotopname,
eb.vegeinheit AS vegeinheit, eb.hc,
COALESCE(eb.uc1, '-') || ' ' || COALESCE(eb.uc2, '-') AS ucs,
eb.e_datum AS datum, eb.kartierer, pv.dzv AS dzv, kk.bezeichnung AS
Quelle,
ST_AsEWKT(eb.geom) geom_WKT
FROM archiv.erfassungsboegen eb
JOIN archiv.pflanzenvorkommen pv ON pv.kartierung_id = eb.id
LEFT JOIN archiv.kampagnen kk ON kk.id = eb.kampagne_id
LEFT JOIN mvbio.pflanzenarten_gsl gsl ON gsl.valid_nr = pv.valid_nr
WHERE
gsl.valid_name LIKE 'Stratiotes%' AND gsl.synonym='0' AND eb.aktuell
ORDER BY
art_gsl, CASE dzv WHEN 'D' THEN 1 WHEN 'Z' THEN 2 WHEN 'V' THEN 3 END
```

Qualitätskontrolle**Verlustbögen: Herkunft des untergegangenen Bogens**

Verlustmeldungen dürfen sich nur auf aktuelle Biotope (BK1/BK1315, ...) beziehen.

Abfrage, welche Verlustbögen sich nicht auf einen aktuellen Biotop beziehen:

SELECT

```
v.id, a.giscode, v.kartierer, v.bearbeitungsstufe,
kk.abk, a.aktuell
FROM mvbio.verlustobjekte v, archiv.erfassungsboegen a
JOIN archiv.kampagnen kk ON kk.id = a.kampagne_id
WHERE v.bogen_id = a.id AND v.kartiergebiet_id != 3459 AND (NOT a.aktuell
OR a.kampagne_id NOT IN (1,3,12,15))
ORDER BY a.kampagne_id
```

Vollständigkeit Differenzflächen (BK2021)

Abfrage listet alle Differenzflächen auf, die nicht in einem Kartier- oder Verlustobjekt genannt werden. + Link zum BK1-Bogen in MVBIO-Pro.

Die Spalten `biotop_inters` bzw. `verlust_inters` zeigen an, ob eine räumliche Überlagerung mit Kartierobjekten vorliegt.

```
WITH diff AS
  (SELECT df.giscode giscode_diff, df.los_nr,
    EXISTS (SELECT * FROM mvbio.kartierobjekte ko WHERE
      st_intersects(ko.geom,df.the_geom)) AS biotop_inters,
    EXISTS (SELECT * FROM mvbio.kartierobjekte ko WHERE
      ko.giscode=df.giscode OR df.giscode IN (SELECT UNNEST
      (ko.zusammengefasste_boegen))) AS biotop_giscode,
    EXISTS (SELECT * FROM mvbio.verlustobjekte vo WHERE
      st_intersects(vo.geom,df.the_geom)) AS verlust_inters,
    EXISTS (SELECT * FROM mvbio.verlustobjekte vo WHERE
      vo.giscode=df.giscode) AS verlust_giscode
    FROM mvbio.differenzflaechen df)
SELECT * ,
  CASE
    WHEN SUBSTRING(giscode_diff FROM 10 FOR 1)='4' THEN
      (('=HYPERLINK("https://mvbio.de/kvwmap/index.php?go=Layer-Suche_Suchen&selected_layer_id=140&value_label='::text || giscode_diff) ||
      '&operator_label=%20=";"::text) || giscode_diff::text) || "'::text
    ELSE
      (('=HYPERLINK("https://mvbio.de/kvwmap/index.php?go=Layer-Suche_Suchen&selected_layer_id=196&value_label='::text || giscode_diff) ||
      '&operator_label=%20=";"::text) || giscode_diff::text) || "'::text
    END AS giscode_link
FROM diff
WHERE NOT biotop_giscode AND NOT verlust_giscode AND los_nr=2 ORDER BY
biotop_inters, giscode_diff
```

Rückgabewünsche zurücksetzen

Setzt alle Bögen mit Rückgabewunsch in allen Kampagnen auf Stufe 2 zurück und ergänzt eine Info in den Prüfhinweisen.

```
UPDATE mvbio.kartierobjekte
SET
  pruefer_rueckweisung=TRUE, --deaktiviert die Prüfung
  "validate_kartierobjekte"
  pruefer_pruefhinweis=COALESCE(pruefer_pruefhinweis||',',
  ',')||'|'|CURRENT_DATE::text||'|': Rückgabewunsch',
  bearbeitungsstufe=2
WHERE rueckgabewunsch

--analog für Verlustbögen:
UPDATE mvbio.verlustobjekte
SET
  pruefer_rueckweisung=TRUE, --deaktiviert die Prüfung
  "validate_kartierobjekte"
```

```
pruefer_pruefhinweis=COALESCE(pruefer_pruefhinweis||',',
',')||'|'|CURRENT_DATE::text||']: Rückgabewunsch',
bearbeitungsstufe=2
WHERE rueckgabewunsch
```

Rückgabe von Bögen mit QS-Fehlern

Genutzt wird der View mvbio.qs_00_alle_boegen, der die IDs aller Bögen aus den QS-Views sammelt. Im Prüfkomentar werden die QS-Codes mit Datum angehängt. Nach Rücksetzung sind die QS-Views und mvbio.qs_00_alle_boegen wieder leer. Analog für Verlustbögen

```
UPDATE mvbio.kartierobjekte k
SET
LOCK = TRUE, --deaktiviert die Prüfung "validate_kartierobjekte"
pruefer_rueckweisung=TRUE,
pruefer_pruefhinweis=COALESCE(pruefer_pruefhinweis||',',
',')||'|'|CURRENT_DATE::text||']: QS '||COALESCE((SELECT fehlercodes FROM
mvbio.qs_00_alle_boegen qs WHERE k.id=qs.bogen_id),'alles ok'),
bearbeitungsstufe=2
WHERE id IN (SELECT bogen_id FROM mvbio.qs_00_alle_boegen);

--analog für Verlustbögen:
UPDATE mvbio.verlustobjekte vo
SET
LOCK = TRUE, --deaktiviert die Prüfung "validate_kartierobjekte"
pruefer_rueckweisung=TRUE,
pruefer_pruefhinweis=COALESCE(pruefer_pruefhinweis||',',
',')||'|'|CURRENT_DATE::text||']: QS '||COALESCE((SELECT fehlercodes FROM
mvbio.qs_00_alle_verluste vq WHERE vo.id=vq.verlust_id),'alles ok'),
bearbeitungsstufe=2
WHERE id IN (SELECT verlust_id FROM mvbio.qs_00_alle_verluste);
```

From:

<https://mvbio.de/nutzerdoku/> - **Nutzerdoku MVBio**

Permanent link:

<https://mvbio.de/nutzerdoku/doku.php?id=admindoku&rev=1718649155>

Last update: **2024/11/22 11:34**

