BIODIVERSITY
BUILDING
BLOCKS FOR
POLICY

M5 Software has been tested by partners and feedback is collected

14/12/2024

Author(s): Peter Desmet, Matthew Blissett, Tim Robertson

Funded by
the European Union

This project receives funding from the European Union’s Horizon Europe Research and Innovation Programme (ID No 101059592). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the EU nor the EC can be held responsible for them.

Table of contents

Summary 3
List of abbreviations 3
1. Development of the occurrence cube software 4
2. Setup and demonstration of a test environment 4
3. Capturing feedback 4
4. Acknowledgements 5
5. References 6

Summary

GBIF is developing occurrence cube software (D2.2) that implements the specification for occurrence cubes and their production (D2.1). This milestone reports what steps have been taken so that partners can test the software and report issues and improvements.

A cloud-hosted test environment has been set up by GBIF on Microsoft Azure. It provides access to the occurrence cube software that is in development as well as monthly snapshots of GBIF occurrence data. Testers can access this environment through the web-based platform Databricks and create cubes in their workspace using Jupyter notebooks. There is no need for local installation. GBIF organized an online training session on November 24 and provided user accounts to all interested partners. Users can provide feedback, report errors or request features through GitHub issues in a public repository.

In alignment with specification, SQL has been selected as the query language to generate cubes. By providing the occurrence cube software *user defined functions*, it allows users to filter data, group it into dimensions and calculate measures.

List of abbreviations

EU	European Union
EEA	European Environment Agency
GBIF	Global Biodiversity Information Facility
HDFS	Hadoop Distributed File System
SQL	Structured Query Language
UDF	User Defined Function

1. Development of the occurrence cube software

GBIF is developing occurrence cube software (D2.2) that implements the specification for occurrence cubes and their production (D2.1). In alignment with the specification, SQL has been selected as the query language to generate cubes. The software provides user defined functions (UDF), which users can load into their SQL to simplify some of the complex aggregation queries (Figure 1). One such function “EeaCellCodeUdf()” is already available. It allows random assignment of an occurrence within its coordinate uncertainty to a cell of the EEA Reference Grid [1]. It replicates functionality developed for the TrIAS project [2], which is currently in use for cube generation [3]. Those cubes can now be generated with the new software.

2. Setup and demonstration of a test environment

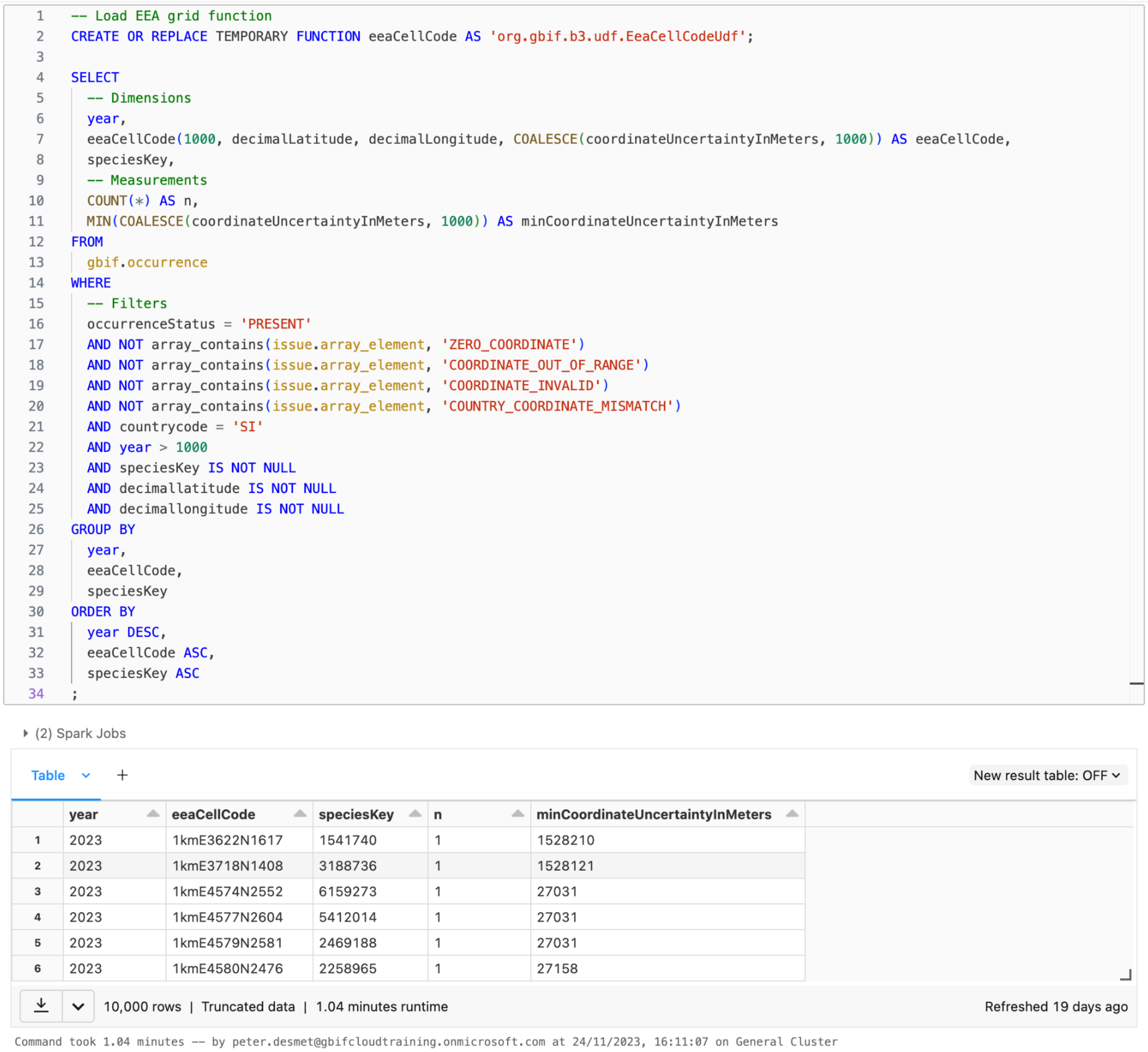
Following an initial validation test on the GBIF infrastructure (Hadoop HDFS, Hive, MapReduce and Spark), a cloud-hosted test environment was set up by GBIF on Microsoft Azure. It provides access to the occurrence cube software that is in development as well as monthly snapshots of GBIF occurrence data [4], as a queryable table.

Testers can access this environment through the web-based platform Databricks and create cubes in their workspace using Jupyter notebooks. There is no need for local installation. Databricks was selected as it closely resembles the GBIF infrastructure and therefore a tried-and-tested method for reporting on GBIF data. Databricks also provides SQL based querying of data tables which is well suited as a notation format for realising the species occurrence cube specification. The test environment was set up using credits awarded to GBIF by Microsoft and GBIF is monitoring spending.

GBIF organized an online training session to demonstrate the test environment on November 24. It was organized by Matt Blissett (GBIF) and Tim Robertson (GBIF), and attended by Peter Desmet (EV INBO), Ward Langeraert (EV INBO), Maarten Trekels (MeiseBG), Lissa Breugelmans (MeiseBG), Matilde Martini (UNIBO), Michele Di Musciano (UNIBO). Pieter Huybrechts (EV INBO), Damiano Oldoni (EV INBO) and Sandra MacFadyen (SUN) watched the recording. All participants were provided a user account and they completed a basic workflow to produce a species occurrence cube using an EEA grid (Figure 1).

3. Capturing feedback

Initial feedback was collected and answered during the training session. An agreement has been made that further feedback, bug reports and feature requests will be captured and managed in the GitHub repository for the occurrence cube software [5]. This allows open discussion, collaborative development, and traceability between commits and the issue they fix.



```sql
-- Load EEA grid function
CREATE OR REPLACE TEMPORARY FUNCTION eeaCellCode AS 'org.gbif.b3.udf.EeaCellCodeUdf';

SELECT
  -- Dimensions
  year,
  eeaCellCode(1000, decimalLatitude, decimalLongitude, COALESCE(coordinateUncertaintyInMeters, 1000)) AS eeaCellCode,
  speciesKey,
  -- Measurements
  COUNT(*) AS n,
  MIN(COALESCE(coordinateUncertaintyInMeters, 1000)) AS minCoordinateUncertaintyInMeters
FROM
  gbif.occurrence
WHERE
  -- Filters
  occurrenceStatus = 'PRESENT'
  AND NOT array_contains(issue.array_element, 'ZERO_COORDINATE')
  AND NOT array_contains(issue.array_element, 'COORDINATE_OUT_OF_RANGE')
  AND NOT array_contains(issue.array_element, 'COORDINATE_INVALID')
  AND NOT array_contains(issue.array_element, 'COUNTRY_COORDINATE_MISMATCH')
  AND countrycode = 'SI'
  AND year > 1000
  AND speciesKey IS NOT NULL
  AND decimallatitude IS NOT NULL
  AND decimallongitude IS NOT NULL
GROUP BY
  year,
  eeaCellCode,
  speciesKey
ORDER BY
  year DESC,
  eeaCellCode ASC,
  speciesKey ASC
;
```

(2) Spark Jobs

	year	eeaCellCode	speciesKey	n	minCoordinateUncertaintyInMeters
1	2023	1kmE3622N1617	1541740	1	1528210
2	2023	1kmE3718N1408	3188736	1	1528121
3	2023	1kmE4574N2552	6159273	1	27031
4	2023	1kmE4577N2604	5412014	1	27031
5	2023	1kmE4579N2581	2469188	1	27031
6	2023	1kmE4580N2476	2258965	1	27158

10,000 rows | Truncated data | 1.04 minutes runtime

Refreshed 19 days ago

```
Command took 1.04 minutes -- by peter.desmet@gbifcloudtraining.onmicrosoft.com at 24/11/2023, 16:11:07 on General Cluster
```

Figure 1: Screenshot of the test environment, where users can use SQL (code above) to create species occurrence cubes (table below).

4. Acknowledgements

We thank Microsoft for providing credits for the Azure cloud computing platform under the Planetary Computer program [6], allowing GBIF to evaluate and run project-wide training on Databricks.

5. References

1. European Environment Agency (2013) EEA reference grid. Accessed via https://www.eea.europa.eu/data-and-maps/data/eea-reference-grids-2 on 2023-12-13.
2. Vanderhoeven S, Adriaens T, Desmet P, Strubbe D, Backeljau T, Barbier Y, Brosens D, Cigar J, Coupremanne M, De Troch R, Eggermont H, Heughebaert A, Hostens K, Huybrechts P, Jacquemart A, Lens L, Monty A, Paquet J, Prévot C, Robertson T, Termonia P, Van De Kerchove R, Van Hoey G, Van Schaeybroeck B, Vercayie D, Verleye T, Welby S, Groom Q (2017). Tracking Invasive Alien Species (TrIAS): Building a data-driven framework to inform policy. Research Ideas and Outcomes 3: e13414. https://doi.org/10.3897/rio.3.e13414
3. Oldoni D, Groom Q, Adriaens T, Davis AJS, Reyserhove L, Strubbe D, Vanderhoeven S, Desmet P (2023). Occurrence cubes for non-native taxa in Belgium and Europe (Version 20231106) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.10058400
4. https://www.gbif.org/occurrence-snapshots
5. https://github.com/gbif/occurrence-cube
6. https://planetarycomputer.microsoft.com