BIODIVERSITY BUILDING BLOCKS FOR POLICY

D2.3 Occurrence cube service

25/02/2025

Author(s): Matthew Blissett, Morten Høfft, John Waller, Andrew Rodrigues, Daniel Noesgaard, Tim Robertson, Peter Desmet

Funded by the European Union

Prepared under contract from the European Commission
Grant agreement No. 101059592
EU Horizon Europe Research and Innovation Action

Project acronym:	**B3**
Project full title:	**Biodiversity Building Blocks for policy**
Project duration:	01.03.2023 – 31.08.2026 (42 months)
Project coordinator:	Dr. Quentin Groom, Agentschap Plantentuin Meise (MeiseBG)
Call:	HORIZON-CL6-2021-GOVERNANCE-01
Deliverable title:	Public services
Deliverable n°:	D2.3
WP responsible:	WP2
Nature of the deliverable:	OTHER
Dissemination level:	Public
Licence of use:	Creative Commons Attribution 4.0 International
Lead partner:	GBIFS
Recommended citation:	Blissett, M., Høfft, M., Waller, J., Rodrigues, A., Noesgaard, D., Robertson, T. & Desmet, P. (2025). ***Occurrence cube service***. B3 project deliverable D2.3.
Due date of deliverable:	Month n°24
Actual submission date:	Month n°24

Deliverable status:

Version	Status	Date	Author(s)
0.9	Draft for review	14 February 2024	Matthew Blissett, Morten Høfft, John Waller, Andrew Rodrigues, Daniel Noesgaard, Tim Robertson (all GBIFS), Peter Desmet (EV INBO)
1.0	Final	24 February 2024	Duccio Rocchini (UNIBO), Sabrina Kumschick (SUN), Lissa Breugelmans, Maarten Trekels (MeiseBG), Peter Desmet (EV INBO)

Table of contents

Key takeaway messages

- GBIF has deployed a new service to create species occurrence cubes, which is the final deliverable for WP2. It will be announced in March 2025.
- The service meets the initial requirements outlined in deliverable D2.1 and we have responded to additional user feedback. We will continue to do so now that the service is operational.
- The service allows users to query and aggregate occurrence data with the Structured Query Language (SQL), offering great customization.
- The service is integrated in the GBIF Download API, creating cubes that can be cited with a DOI.
- The service can be used via the command line, R library, Python library or browser (GBIF.org website). The latter provides a graphical user interface to create cubes without SQL knowledge.
- Documentation, tutorials and training events are produced/planned to support users.
- Long-term availability, maintenance and support are ensured by embedding the service in GBIF infrastructure and core operations.
- We aim to reach technology readiness level (TRL) 8 by the end of the project.

Executive summary

GBIF has deployed a service to create species occurrence cubes. The service uses software (described in D2.2) that allows users to query and aggregate species occurrence data with the Structured Query Language (SQL). This well-known language can handle diverse and complex data operations, well beyond generating data cubes. The software also provides user defined functions (UDF) to shield users from having to create complex aggregation queries themselves.

The service is integrated in the existing GBIF Download API. It can be used by any registered user of GBIF.org. Produced cubes (or other SQL queries) are assigned a DOI that can be cited. Users can interact with the service programmatically using the command line interface or the updated libraries in R (rgbif) and Python (pygbif). The easiest way to use the service is directly through the GBIF.org website, where a new download option allows users to create a cube through a series of dropdown menus and checkboxes. No SQL knowledge is required for this, but advanced users can switch to a SQL editor to customize their query further.

Requirements for this software and service were gathered in previous deliverables (D2.1), but we have responded to additional user feedback and will continue to do so in the future. We have also produced documentation and tutorials and will organize a number of training activities. Long-term availability and maintenance are guaranteed by embedding the service into GBIF.org infrastructure and core operations, with the aim of reaching technology readiness level (TRL) 8 by the end of the project. The service is the final deliverable for WP2 and will be announced by GBIF in March 2025.

Non-technical summary

The Global Biodiversity Information Facility (GBIF) has launched a new service that helps users download summaries of species occurrence data. It is built into the GBIF Download API and can be accessed via their website, command line, or programming tools in R and Python. The website offers a simple form to create cubes, but advanced users can customize their queries using the Structured Query Language (SQL). Generating cubes are assigned a unique identifier that can be used in citations. GBIF will maintain the service long term as part of their core operations and incorporate user suggestions. Documentation, tutorials and training events were also produced/planned. The service will be officially announced in March 2025.

List of abbreviations

API	Application Programming Interface
DOI	Digital Object Identifier
EEA	European Environment Agency
EBV	Essential Biodiversity Variable
EU	European Union
GBIF	Global Biodiversity Information Facility
SQL	Structured Query Language
TRL	Technology Readiness Level
UDF	User Defined Function
UI	User Interface

1. Introduction

The occurrence cube software (D2.2, Blissett et al. 2024a) implements the specification for occurrence cubes and their production (D2.1, Desmet et al. 2023a). In alignment with the specification, Structured Query Language (SQL) was selected as the query language to generate cubes. Its popularity, scalability, and extensibility makes it an ideal choice for handling diverse and complex data operations, well beyond generating cubes. The software provides user defined functions (UDF) (see Figure 1) to shield users from having to create complex aggregation queries themselves (e.g. assigning coordinates to a reference grid). It has been released under an open source license (Blissett et al. 2024b).

In this document we describe how we implemented this software as a service, which is the final deliverable for WP2. We also address feedback we received on the previous deliverable reports of WP2 (D2.1 and D2.2).

```
Unset
SELECT
  -- A temporal dimension (here: year and month)
  PRINTF('%04d-%02d', "year", "month") AS yearMonth,
  -- A spatial dimension (here: EEA grid supported by a UDF)
  GBIF_EEARGCode(
    1000,
    decimalLatitude,
    decimalLongitude,
    COALESCE(coordinateUncertaintyInMeters, 1000)
  ) AS eeaCellCode,
  -- A taxonomic dimension (here: species)
  speciesKey,
  species,
  -- A measurement (here: number of occurrences)
  COUNT(*) AS occurrence_count
FROM
  occurrence
WHERE
  occurrenceStatus = 'PRESENT'
  AND countryCode = 'PL'
GROUP BY
  yearMonth,
  eeaCellCode,
  speciesKey,
  species
```

Figure 1: An example of SQL query to aggregate occurrence data into a cube.

2. Requirements

Capturing requirements and feedback

Initial requirements were captured from project partners in the kick-off meeting (13-14 March 2023), two online calls (24 and 27 April 2023) and a document open for comments. These were formalized in requirements (D2.1), which served as the blueprint for the software (D2.2) and service (D2.3, this document).

We received further feedback from project partners and the sister EU project Fairicube, in the form of comments on the requirement document, feedback from testers (M5, [ref]) and feature requests as GitHub issues. We responded to these by implementing the suggested feature, recommending alternative ways to obtain the desired result and/or providing detailed documentation.

Feedback received up to now represents opinion from an informed user group of experts, but we acknowledge that there will be additional needs and suggestions now that the service is publicly available. Our approach has therefore been to use the specification (D2.1) as the requirement for an initial release and be open and prepared for additional feedback. This is supported by:

- Our design decision to use SQL and UDF for querying data, which can easily be extended with additional features.
- Following standard practices of open source development and feature request tracking using GitHub.
- Embedding the service within the established GBIF.org infrastructure, ensuring long-term maintenance, iterative improvements and sustainability.

As an example, this approach has been used to implement the user suggestion to support spatial aggregation of data in the hexagonal ISEA3H grid.

Non-functional requirements

The specification (D2.1) outlines the technical requirements, but several non-functional requirements were considered through the process. These include:

- **Usability**: achieved by offering a simple web form solution that satisfies most common needs.
- **Extendibility**: achieved by using SQL as the notation format and for computation, allowing for easy expansion without changes to the API.
- **Performance**: achieved by deploying the service on a clustered computing solution (Spark) that can be resized to accommodate growth in data volume and user demand.
- **Sustainability**: achieved by 1) integrating the service in GBIF.org infrastructure and core operations, such as training and helpdesk and 2) ensuring the software is portable to other environments.

- **Efficiency in operating costs (computation)**: achieved by integrating the service in GBIF.org infrastructure, rather than requiring new infrastructure, or a pay-by-use service on a commercial cloud solution.
- **Alignment with other initiatives**: achieved through engagement with sister EU projects and following data formats emerging from data cube initiatives in other domains.

Requirement coverage

The requirements initially described in the specification (D2.1) have different levels of importance, i.e. MUST, SHOULD, and MAY following RFC 2119. The delivered software and service now cover 98% of the MUST requirements (79 of 81), 87% of SHOULD requirements (59 of 68) and 35% of MAY requirements (13 of 37). See Table 1 for an overview.

Table 1: Overview of the requirements in the specification (D2.1) and to what extent they are implemented.

Section	MUST	SHOULD	MAY
3 Cube specification			
3.1 Dimensions	3 of 3		
3.1.1 Taxonomic	7 of 7	9 of 9	0 of 1
3.1.2 Temporal	5 of 5	3 of 3	0 of 1
3.1.3 Spatial	12 of 12	6 of 10	0 of 1
3.1.4 Other	4 of 4	2 of 2	3 of 6
3.2 Measures		1 of 1	
3.2.1 Occurrence count	2 of 2		
3.2.2 Minimum coordinate uncertainty	1 of 1	1 of 1	
3.2.3 Minimum temporal uncertainty		1 of 1	1 of 3
3.2.4 Sampling bias	2 of 2	12 of 12	1 of 1
3.3 Format	3 of 4[1]	1 of 2	0 of 8
3.4 Metadata	10 of 10[2]	3 of 4	0 of 1
3.5 Findability and storage	2 of 3[2]	4 of 5	0 of 3
4.1 Cube production software			

[1] EBV Cube format and EBV Data Portal integration are considered MUST requirements, but we decided those are better implemented downstream. See section 6 (Next steps).
[2] GBIF considers the creator of a (cube) download as personal data. It is therefore not shown through the API, website or metadata.

4.1.1 Source data	3 of 3	2 of 2	
4.1.2 Parameters	3 of 3	1 of 1	1 of 1
4.1.3 Reference grids	1 of 1	2 of 2	1 of 1
4.1.4 Cube specification	3 of 3	2 of 2	
4.1.5 Cloud processing	3 of 3	1 of 1	
4.1.6 Best practices	3 of 3	2 of 2	
4.1.7 Open source	2 of 2	2 of 2	
4.2 Cube workflow service		**1 of 1**	
4.2.1 Search and filter	1 of 1		
4.2.2 Exclude unwanted occurrences			1 of 2[3]
4.2.3 Dimensions	4 of 4		1 of 3
4.2.4 Measures			1 of 2
4.2.5 Output format	1 of 1	0 of 1[4]	1 of 1
4.2.6 Destination	1 of 1	0 of 1[5]	1 of 1
4.2.7 Documentation	1 of 1		
4.2.8 API	1 of 1	1 of 1	
4.2.9 GBIF API integration	1 of 1	2 of 2	1 of 1

3. API

The service is implemented as an extension to the GBIF Occurrence Download API, which now accepts SQL download requests. Since the capabilities of SQL are broader than generating cubes, this service is called the **SQL download API**.

Interface

User can interface with the API in four ways:

- **GBIF.org website (browser):** see the section Graphical User Interface below.
- **Command line interface**: e.g. by using curl.
- **Python**: by using the extended download functionality in the pygbif library.
- **R**: by using the new occ_download_sql function in the rgbif package.

[3] Supported by the API, but not the website (will be added later in 2025).
[4] Implemented as the suggested alternative.

Request

To use the API, one must send a POST request to the endpoint *occurrence/download/request* with a specific format ("format" = "SQL_TSV_ZIP") and the SQL query ("sql" = "SELECT …"). The request requires authentication with a GBIF.org username and password.

Response

The API will respond with one of the following HTTP statuses:

- **201**: the query is accepted and a download key is returned.
- **400**: the SQL is invalid. There may be useful information on the error at the end of the output. What SQL syntax is allowed is documented here.
- **401**: the provided username or password is incorrect
- **403**: the provided username does not have permission to use this feature (was used for testing)

If the request is valid, the download service will asynchronously create the requested download. The status of the download can be obtained via the endpoint *occurrence/download/{key}*, as well as the used SQL query and a DOI that can be used in citations. Once completed, the endpoint will also return a URL to download the data.

Documentation

Documentation of the service is available at:

- **techdocs.gbif.org**: a website with all technical documentation on GBIF services. It now has pages on the SQL download API, the UDFs and how to use SQL to create cubes.
- **OpenAPI and Swagger**: technical documentation of all GBIF APIs, including the option to interact and test requests directly in the browser.

Technology readiness level

The service has a technology readiness level (TRL) of 7 (System prototype demonstration in operational environment). We expect it to reach 8 (System complete and qualified) by the end of the project.

4. Graphical user interface

To aid non-technical users in using the service, we developed a graphical user interface that is integrated in the GBIF.org website. It caters to users who 1) want to specifically create cubed data (hiding the other possibilities of SQL) and 2) want a graphical user interface for their SQL query (rather than interacting with the API directly).

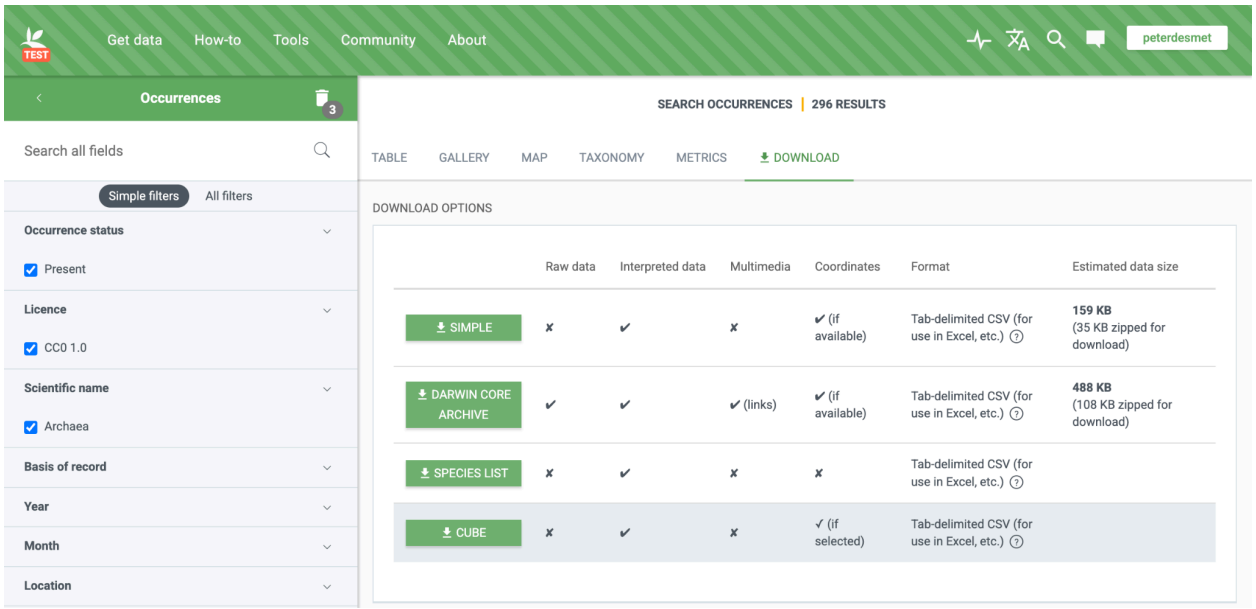

Figure 2: Screenshot of the GBIF.org download page, where users are presented with the option to format data as a cube.

To use this interface, users can search and filter occurrence data of their interest in the occurrence search page. Once they are happy with the result, they can download the data, which requires a (free) login. On the download page, they are now presented with the option to download data as a cube (see Figure 2), in addition to the already existing formats (“Simple”, “Darwin Core Archive” and “Species list”).

When selecting the “Cube” option, an intuitive interface is shown with dropdown menus and checkboxes (see Figure 3). This aids users in creating cubes in the dimensions, granularity and measures they want. This interface implements all the options and defaults as defined in the specification (D2.1) and covers the majority of use cases. For other use cases (or when users want to use SQL for other purposes than to create cubes), they have the option to switch to an online SQL editor to customize their query (see Figure 4).

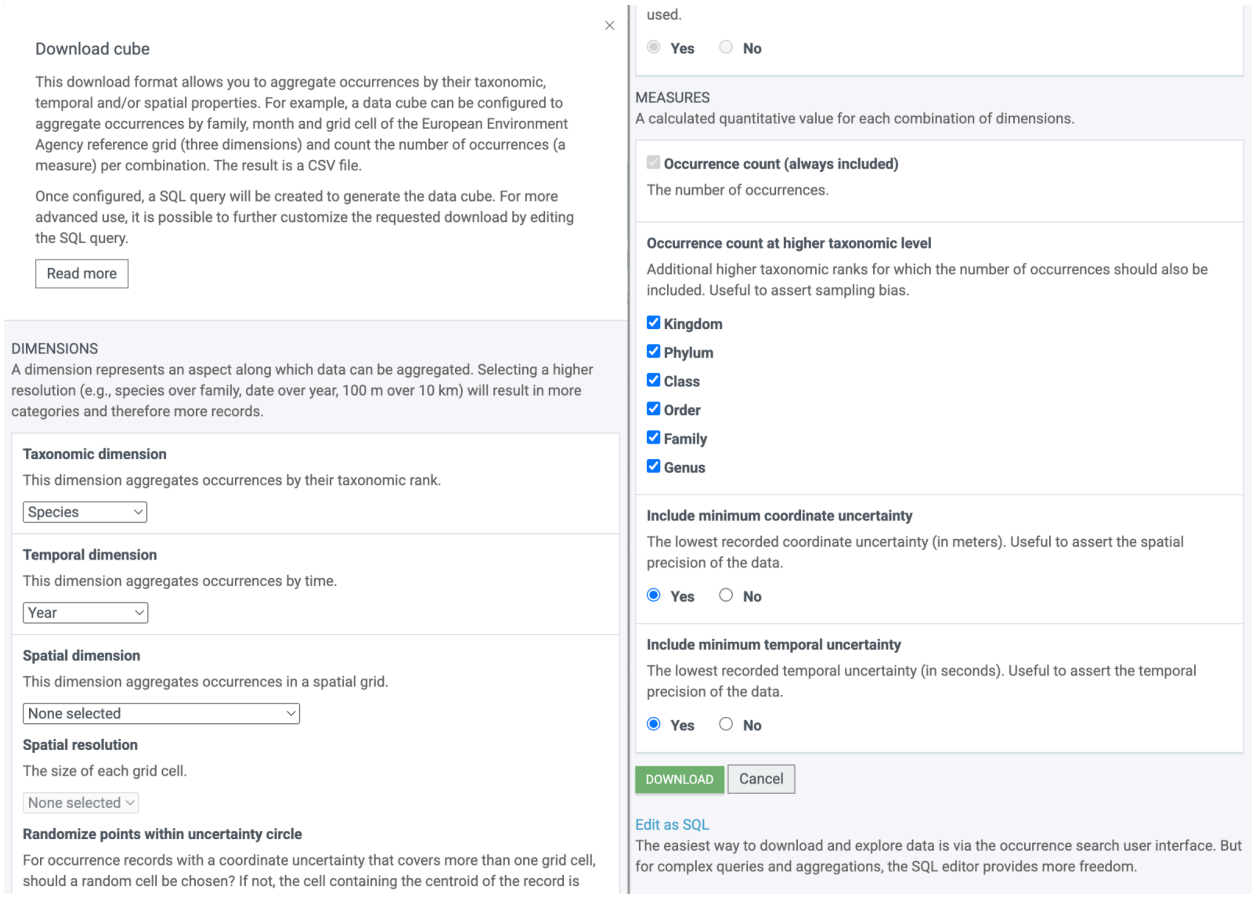

Figure 3: Screenshot of the GBIF.org modal (split in two in the figure) presented to users when they choose to download data as a cube. Dimensions and measures can be defined with a number of dropdown menus and checkboxes. Users also have the option to further edit their query in SQL.

Clicking the download button will trigger the service to create a download with the selected parameters. As with regular GBIF downloads, these downloads have metadata regarding the selected parameters (i.e. the SQL query) and a DOI that can be used in citations. Users will be notified by email when a download is ready. They can find all requested downloads under their personal downloads page (requires login).

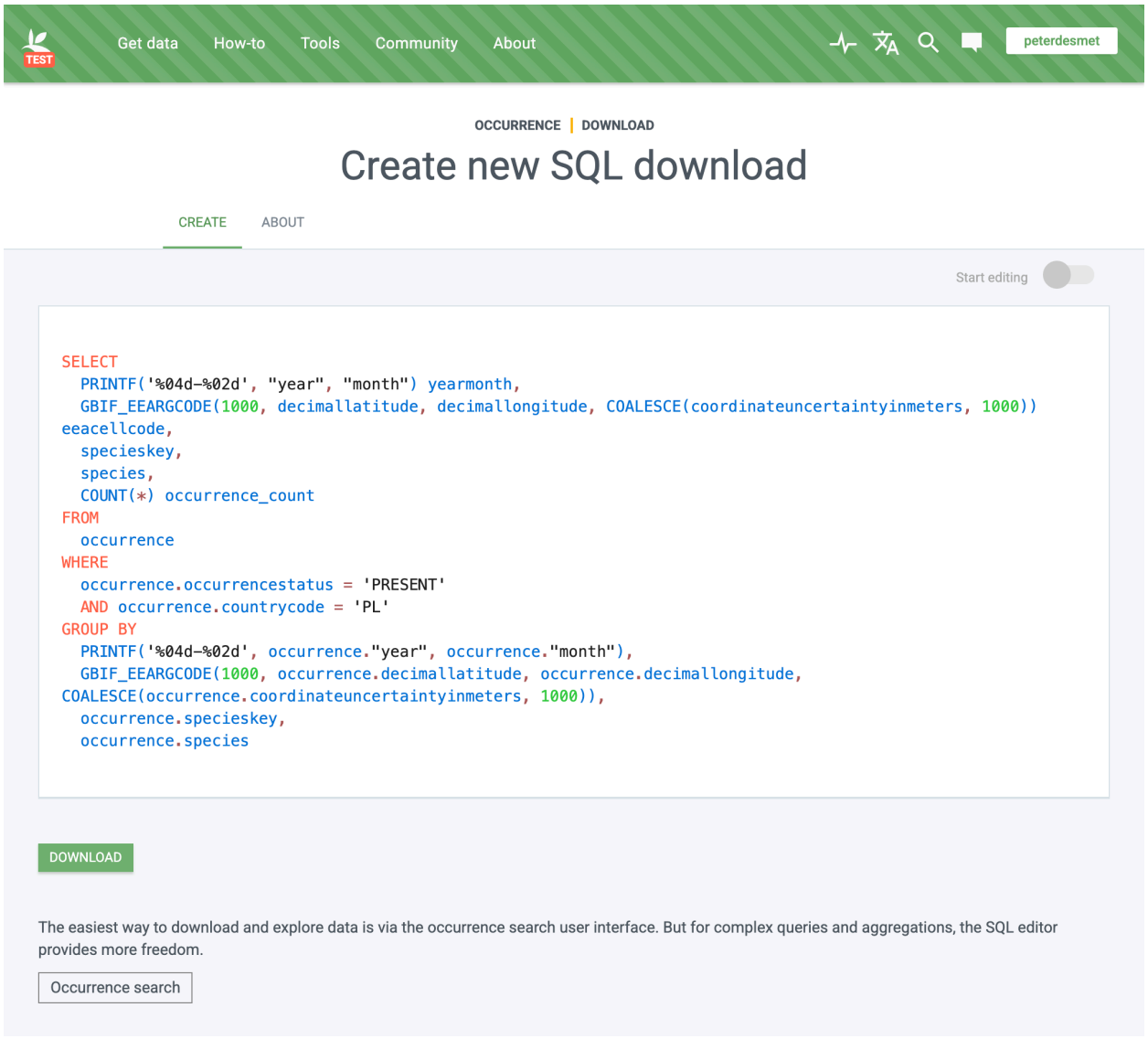

Figure 4: Screenshot of the GBIF.org SQL editor presented to users if they want to customize their query in SQL.

5. User support

Users of the new service are supported in a number of ways.

Training events

We organized or will organize training on the SQL Download API at the following venues:

- **Technical support hour for GBIF nodes**: online helpdesk, July 2024.
- **Biospace25 conference**: in-person workshop, February 2025.
- **Datos Vivos conference**: in-person workshop (abstract submitted), October 2025.
- **GBIF Data Use curriculum**: a specific module on data cubes and SQL downloads, mid-2025.

Tutorials

In addition to the service documentation, we have created the following tutorials on how to use the SQL Download API:

- **GBIF SQL downloads**: blog post on the GBIF data blog, September 2024.
- **How to build a species occurrence cube from a GBIF checklist**: tutorial on the B3 documentation website, February 2025.

We will add more tutorials if the need arises.

User feedback

Users can ask questions or request suggestions via:

- **GBIF helpdesk**: accessible via the “Feedback and questions” feature on GBIF.org.
- **GitHub issues**: feature requests can be submitted in the public occurrence cube software repository on GitHub.
- **Contacting project partners**: project partners can be contacted directly.

6. Next steps

GBIF will announce the service's availability and promote its capabilities beginning in March 2025. These announcements will be distributed through GBIF’s communication channels including its main website (GBIF.org), social media, and project partner networks. Announcements will be redistributed through B3’s communication channels where relevant.

Once announced, we plan to:

- **Listen for user feedback**: we will monitor and respond to user feedback and suggestions. This will guide future improvements.

- **Support the EBV NetCDF format**: one of the requested output formats is EBV NetCDF, which is the standard used for essential biodiversity variable datasets on the EBV Data Portal. This format is quite specific and cannot be used for an SQL query. It will therefore be supported as a downstream (CSV to EBV NetCDF) in the ebvcube R package (Quoss et al. 2024). This also allows for better maintenance, since the R package and EBV Data Portal are maintained by the same organization. We will facilitate them with this transformation, for example by adding features they deem critical (e.g. structured metadata on the selected dimensions in addition to the used SQL query).
- **Reach TRL 8**: we will improve the service and monitor its uptake, so it can reach TRL 8 (System complete and qualified) by the end of the B3 project.

7. References

Blissett M, Robertson T, Desmet P (2024a). Occurrence cube implementation. B3 project deliverable D2.2. https://b-cubed.eu/storage/app/uploads/public/65e/1d0/3d2/65e1d03d2d9d5647000672.pdf

Blissett M, Robertson T, Desmet P (2024b). Occurrence cube functions (cube-0.2.0). Global Biodiversity Information Facility (GBIF). https://doi.org/10.5281/zenodo.10607133

Desmet P, Oldoni D, Blissett M, Robertson T (2023a). Specification for species occurrence cubes and their production. B3 project deliverable D2.1. https://b-cubed.eu/storage/app/uploads/public/64d/1f7/5a2/64d1f75a2fc96997998232.pdf (online version at https://docs.b-cubed.eu/guides/occurrence-cube/)

Desmet P, Blissett M, Robertson T (2023b). Software has been tested by partners and feedback is collected. B3 project milestone M5. https://b-cubed.eu/storage/app/uploads/public/676/124/d96/676124d960efa529805380.pdf

Quoss L, Fernandez N, Langer C, Valdez J, Pereira HM (2024). ebvcube: Working with netCDF for Essential Biodiversity Variables. https://doi.org/10.32614/CRAN.package.ebvcube