Deliverable D2.1

Version:	FINAL
Date:	30/04/11
Author:	ICLARM
Dissemination status:	PU
Document reference:	Deliverable_D2.1

Report on Guidelines on best practices for database custodians to facilitate the use of their data

STATUS: FINAL VERSION

Project acronym:	BioFresh
Project name:	Biodiversity of Freshwater Ecosystems: Status, Trends, Pressures, and Conservation Priorities
Call and Contract:	FP7-ENV-2008-1
Grant agreement no.:	226874
Project Duration:	01/11/2009 – 30.04.2014 (54 months)
Co-ordinator:	Leibniz-Institute of Freshwater Ecology and Inland Fisheries at Forschungsverbund Berlin e.V., Germany
Partners:	RBINS, Royal Belgian Institute of Natural Sciences, Belgium
	BOKU, Universität für Bodenkultur Wien, Austria
	ICLARM, International Center for Living Aquatic Resources Management, Malaysia
	IRD, Institut de Recherche pour le Développement, France
	UDE, Universität Duisburg-Essen, Germany
	IUCN, International Union for Conservation of Nature, Switzerland
	UOXF.AC, Oxford University, UK
	UB, Universitat de Barcelona, Spain
	UFZ, Helmholtz Zentrum für Umweltforschung, Germany
	UCL, University College of London, UK
	EAWAG, Eidgenössische Anstalt für Wasserversorgung, Abwasserreinigung und Gewässerschutz, Switzerland
	UCBL, Université Claude Bernard - Lyon 1, France
	UPS, Université Paul Sabatier- Toulouse 3, France
	ECOLOGIC, Ecologic GmbH Institut für Internationale und Europäische Umweltpolitik, Germany
	EC-ERC, Commission of the European Communities - Directorate General Joint Research Centre, Italy
	UD, University of Debrecin, Hungary
	NRM, Naturhistoriska riksmuseet, Sweden

BIOFRESH

Biodiversity of Freshwater Ecosystems: Status, Trends, Pressures, and Conservation Priorities

Project no. 226874

Large scale collaborative project

Deliverable number	D2.1
Deliverable name	Guidelines on best practices for database custodians to facilitate the use of their data
WP no.	WP2
Lead Beneficiary (full name and Acronym)	WorldFish Center (=ICLARM)
Nature	Written report
delivery date from Annex I (proj. month)	M18
Delivered	Yes
Actual forecast delivery date	2011-04-30
Comments	

Project funded by the European Commission within the Seventh Framework Programme Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

This project is supported by funding from the specific programme 'Cooperation', theme 'Environment (including Climate Change)' under the 7th Research Framework Programme of the European Union.

Name of the Authors	Name of the Partner	Logo of the Partner
Bailly Nicolas	WorldFish (formerly ICLARM)	
Schmidt- Kloiber Astrid	BOKU	
De Wever Aaike	RBINS	

In case the report consists of the delivery of materials (guidelines, manuscripts, etc)

Delivery name	Delivery file name	From Partner	To Partner
Report on data encoding (this document)	BF_WP2_D2.1_DatabaseGuidelines_Report_110430_v1_final.doc	WorldFish, BOKU, RBINS	All

Guidelines on best practices for database custodians to facilitate the use of their data

Nicolas Bailly (WorldFish Center)
Aaike de Wever (RBINS)
Astrid Schmidt-Kloiber (BOKU)

SEVENTH FRAMEWORK PROGRAMME

Table of contents

WorldFish Center: formally: ICLARM – International Center for Living Aquatic resources. Malaysia (Philippines)
RBINS: Royal Belgian Institute of Natural Sciences, Belgium
BOKU: Universität für Bodenkultur Wien, Austria

INTRODUCTION

This document presents a number of information, guidelines and recommendations that should help the partners of the BioFresh project to integrate their dataset in the BioFresh portal. It is not a standard that should be followed strictly, but rather a number of indications to facilitate the management of datasets considering the multiplicity of situations.

This document summarises the experience gathered by the partners of the Workblock 1 (Workpackages 1 to 3), both during the first third of the European project BioFresh, and from past experiences in such projects on Biodiversity Informatics.

This is the first version. The document will evolve along the project with new experiences acquired.

The first section makes the difference between a dataset and database, acknowledging the fact that today the word ‘database’ means computerized dataset using a particular type of software called Database Management Systems (DBMS). The second section suggests choices of a DBMS. The third section explains what is a database structure based on the relational model. The fourth section gives some advices on the data quality control in databases. And the fifth section proposes ways to disseminate the database and its information.

What is a database?

Types of databases

A database is primarily a data repository. Beyond that, it is a fuzzy concept.

Databases evolved from (multi-)indexed texts as in books and monographs, to card files, to worksheets, and finally to electronic databases.

The principal goal when building a database is that data must be easily searchable/retrievable. One collateral aspect is that it must be simple to order data according to various ways.

To meet that goal, tables are the real prototype of databases, either printed or electronic (as worksheets, e,g,. MS-Excel). It implies a notion of standardised structure of the data.

A table is a set of cells arranged in rows and columns:

- Rows: usually the items (individuals); also named records;
- Columns: usually the characteristics of the items (variables, characters); also named fields;
- Cells: the piece of data, corresponding to a couple row/column or item/characteristic (value, character state, …); also name field characters.

In oral expression, we often use “character” for “character state”, “field” for “field content”. It may lead to misunderstanding in discussion, and the vocabulary should be précised in many cases.

A database can be a dataset fixed at one time, a snapshot of values that will never change, or require new editions of the complete dataset (e.g., printed repositories). But most often, we want to update the database by adding, editing, and records, attributes, and values.

In addition, if the content management and the access is conducted by several persons (and possibly simultaneously), not all parts of the database may be available for everybody for all usages, and access rights may be defined for the following actions: Read, Add, Edit, Delete.

Support of datasets

There are differences between:

- The physical support (printed, electronic, …);
- The management on the physical support; and
- The logical/semantic structure.

Examples:

- Tables can be printed or under electronic format, electronic files can be under word, worksheet, or database format;
- Cards can be printed/hand-written or electronic (HyperCard);
- Files on disks may be managed in different ways;
- For the same dataset, several semantic structures may be set up.

Issues are or should be independent.

Definition: Database vs. Dataset

A database is a dataset of structured and editable data easily accessible, searchable and re-orderable simultaneously by several users with security.

Although various supports, structures, and organizations are evocated in the previous section, it is clear that this definition is primarily about the electronic files that are managed under various types of Database Management Systems (DBMS). Hereafter, we will use database strictly in that meaning, while we will speak about datasets for all others forms of data repositories.

Digitization

A special recommendation is to digitize data as far as it is possible when they are only in manuscript or printed form. Even under a semi-structured way in a text- or word-processor.

Choice of one database system

Moving towards electronic relational databases

Many data are kept by researchers under text files or worksheets, or notebooks. Although it is reasonable to handle small datasets in this way during the research work in a limited time framework, it is tedious to continue that way with huge datasets, because DBMS include many intrinsic controls to maintain the integrity of the dataset that should be developed “manually” otherwise. And for the management of data in a long-term perspective, datasets are much more exploitable (querying, ordering) through DBMS than through text files; worksheets on the other hand should be seen as short-term tools for dedicated purposes.

As a rule of thumb, it is tedious and error-prone to handle a worksheet with more than 10,000 cells (100 rows × 100 columns; 200 × 50; 500 × 20; 1000 × 10) or with more than 1,000 rows or columns.

Although data can be disseminated through the other types of repositories, data are easier to use from electronic databases, even if the actual data exchange is done through csv, xml, or worksheet files. Exporting the latter from the former is easy, the reverse is time consuming and error-prone.

Database models: the relational model is currently the most popular

Several models were developed to handle large datasets, with associated structure models and languages to develop user interfaces:

- Hierarchical / Network database model based on records (CODASYL model / COBOL language)
- Relational model based on interlinked “relations”=tables (ERM – Entity Relationship Model / SQL = Standard Query Language)
- Object Oriented model based on objects and their hierarchical classes (Java language)
- No-SQL / Triple Store / Doc Store are other less-known attempts.
- XML is also used sometimes not only for exchanges but also as data repositories.

The relational model developed by Codd (1970; 1990 for the last update) is the most used in the world. It is based on a formal algebra and keeps separated the semantic representation of data from the physical management of files on hard disk (which was not the case for the CODASYL model for instance).

Data are structured in tables, tables are structured in records, records are structured in fields. Tables linked through designated fields (fig. 1 next section). The database is searchable through SQL queries.

Note: tables are called “relations” in the original model, hence the name “relational. It does not designate the link between tables as often misunderstood.

The model has some limitations however and was adopted only in the early 80s when software became available. The oriented-object model allows a much better knowledge representation (semantic structure) but is more complicated to implement and thus less easy-to-use; and well-maintained software are not available.

Software recommendation

It is not possible to give a correct advice for all possible cases that are also dependent on software availability, background and history of the database manager and his institution.

But in general, and in the view of the current development of the systems, we recommend:

- MS-Access for small to medium datasets: although it is a proprietary format, it remains reliable and easy-to-use software. Databases are easy to create, and the functionalities to create queries and develop interfaces are intuitive and user-friendly.
- PostgreSQL (Postgres in short) for larger datasets: this is the most advanced open-source DBMS for handling bigger datasets. The difference with MS-Access is that its installation and configuration require some skills in informatics, and that the development of sophisticated interfaces requires skilled programmers. It is preferred to MySQL, the other open source DBMS available because PostgreSQL handles the full Unicode, and has GIS functionalities (PostGIS).

Database structure

Using the relational model

In essence, the relational model helps to split a unique huge table with many duplicated information, in a set of interlinked smaller tables (fig. 1), which minimizes the duplications that are potential source of inconsistencies, reducing at the same time the space used on the hard disk and the data management.

There is a formal way to describe the dataset structure before its implementation in a database (ERM: Entity-Relationships Model). However, this step can be skipped for simple datasets where experimentation can be just used instead.

There is no unique best solution for a given dataset.

As far as possible, existing structures should be re-used.

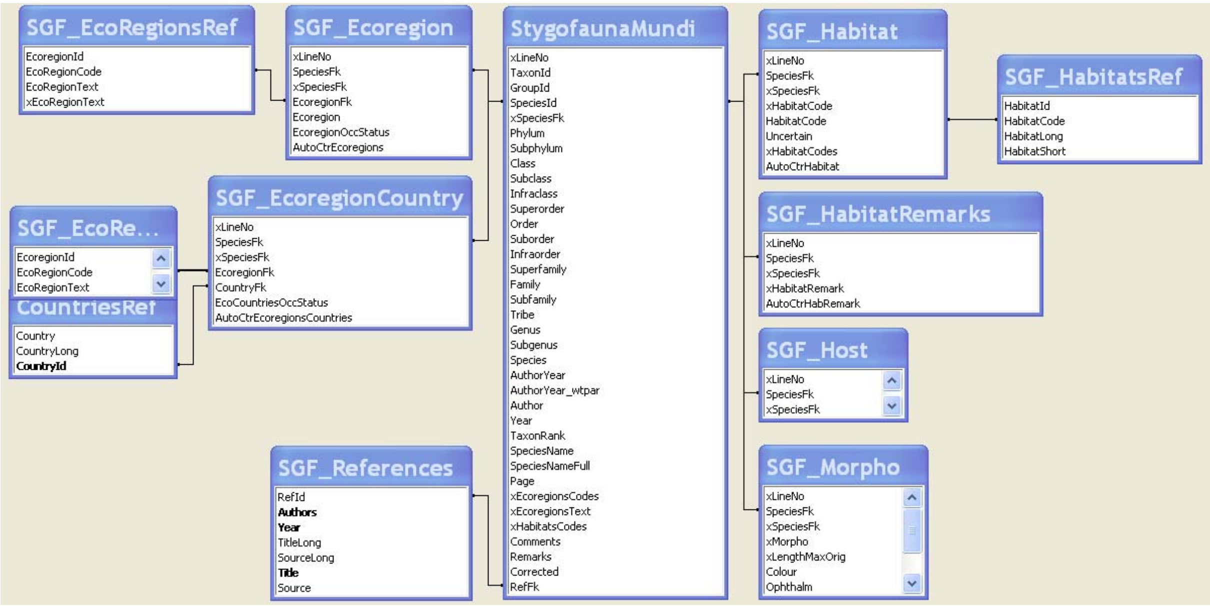

Figure 1. – Example of a relational database structure for the Stygofauna Mundi database developed for BioFresh. Usually, the is one main table driving the entire structure, here the taxon table called StygofaunaMundi.

Balance between theory and practicality: keep it simple

The complete realisation of the relational model in what is called “normal forms” should not be a goal, especially since the multiplication of tables slows down significantly the performance of available software during queries.

A compromise must be found, which also reinforces the idea that there is not a unique best solution for a given dataset, but only heuristic ones balancing database integrity controls and database simplicity.

The step of specification is essential but must be kept in reasonable time limit. Biodiversity is a complex domain, multidimensional (almost fractal), it is not possible to plan in advance all exceptions. For example, 50 cases over 100,000 records do not deserve the creation of a dedicated field. It is quite easy to spend a lot of funds to define a database, but content matters, and must be balanced in the funding. It is better to publish a reasonable complete small dataset, than an empty structure.

Data standards

It was a common practice to rebuild the authority file tables, e.g., countries, taxonomy including classification and names, bibliographic references. Today, many standards are available on the web (in particular those developed by TWDG, see http://www.tdwg.org/standards/, ISO, and many others).

Before developing one's own structure, it is preferable to re-use what exists, adapting it for specific purposes if needed, which allows to free time for more data entries in order to reach more quickly a critical mass of data: donors may be sensitive to that aspect.

Data structure templates for freshwater biodiversity

As long as the portal will include more and more datasets, it will be possible to copy the structure for the same type of data. We may define some general templates that would link directly to the standards used in BioFresh for taxonomic, geographic, and other common attributes.

Some specificities in biodiversity informatics and research

- Always keep trace of the scientific name used as valid for the information (either from its own results or from the bibliographic reference used);
- As much as possible, keep trace of the unique identifiers of the name and the taxon in the reference list, which in BioFresh should be FADA identifiers;
- Use standard check-lists for names such as FADA, Catalogue of Life, PESI, etc., and make drop-down lists in the encoding interface;
- Always keep trace of the bibliographic references used: each piece of information must be linked to a reference;
- Consider the copyrights issues (especially for pictures and pdfs);
- Keep trace of the correct reference of the standards used;
- Keep trace of the units used;
- GIS is an alternative in some cases but should remain linkable to databases;
- GIS data should use the file shapes provided in the BioFresh portal (countries, ecoregions, etc.).

Research versus for-public databases

The database structure may be drastically different if the database is intended for research only, or for public at large dissemination. This is an important decision to be taken at the start because in addition to the structure, the data management (and usually nowadays the website management) are completely different:

Data (unitary piece of data like measurements)	vs	Information (synthetic)
Valid in real time	vs	Valid at a given time
Complex	vs	Simplified
All values from the past	vs	Only last updated value
Need translation for public	vs	Understandable language
Computations possible for further analyses	vs	Computations not necessary
Bottom-up approach (from data to knowledge)	vs	Top-down approach

As far as the BioFresh European Project is concerned, the research database are first targeted, and data should be stored in such a way that they can be used in further analyses.

This aspect is quite important to get. At the same time we are asked at the same time store data for research purposes, and raise the awareness of all publics about the freshwater biodiversity. But databases behind are not the same and are not managed the same. Putting out scientific data and knowledge is not that straightforward. Targets and evaluation must be adapted for each case to avoid misunderstandings.

Database Quality Control

What is database quality control?

It is a set of standards (used and/or checked against), methods, and protocols to assure the reliability of data, at structural (technical, integrity) and semantic (content, consistency) levels of the database.

Data quality may refer to a slightly different equation: Data quality = reliability + accuracy + precision.
For instance, for occurrence data, the reliability resides in who caught and identified the specimens, how the information was handled over years, etc., while accuracy and precision concerns primarily the localization of the catch or the observation, and the rank of the identification.

The quality control is not only some procedures that have to be run, but that should be intrinsically managed along all the life-span of the database, and in a cyclic way (i.e., be sure that data were controlled before each usage and when new entries or updates were made). Each action in a database may compromise the overall quality.

Some references are given at the end focusing on biodiversity data.

There is also the recent development of a standalone web-based tool acquired by Google that is quite efficient to control the most basic typos and inconsistencies in a database (Google Refine).

Some recommendations

The quality control may be organized by taking the step of the data encoding (= data entry but with standardization process) as the central point.

<u>Quality control before the data encoding</u>

- Good database structure minimizing redundancy and with a set of controls included in the table definitions, and in the encoding interface;
- Reuse already verified sources, in particular formal and de facto standards, incl. controlled vocabularies (see section 4.3);
- Develop encoding interface with as many multichoice drop-down menus as possible (= controlled vocabularies).

<u>Quality control during the data encoding</u>

- Having a proper work environment. It includes:
  - Ergonomics: seat, mouse, monitor, light, …;
  - Quiet environment;
  - Force oneself to stop 5 min every hour: the number of errors (typos, wrong field, etc.) increase significantly with the time spent in the front of the computer.
- Follow the same field filling/encoding order;

- Have one's "data encoding notebook" (just like laboratory work requires an "experiment notebook"). It should be used to document any important change in the signification of fields, controlled vocabularies, or automatic corrections;
- Remark text fields are useful to handle exceptions to avoid too complex structures, but the way to fill them must be documented: they should be standardized as much as possible;
- Respect the standards. If needed reconsider their definition at a later stage. It may imply to review the remarks fields and to finally structure them or part.

Quality control after the data encoding

- Check field by field: null, min, max, respect with controlled vocabularies (or external standard);
- Check data content between relevant pair of fields;
- Check data content between relevant n-set of fields;
- Check fields between tables;
- As far as possible, develop null queries, i.e., queries that must give no result when everything is correct;
- Run these queries regularly and before important usage or release.

Dissemination of the database

Choose a telling name and acronym

The name of the main topic of the database should be in the full database name and in the acronym (e.g., FishBase). Changing the name after the database is well-known must be avoided by all means, hence a correct choice at the start is important.

A simple logo is very helpful when the database is used in third parties website: it helps to cite the source and make back deep-links.

Metadata

It is important to give a good description of the database and its content that can be used for indexation in metadatabases.

Several standards exist, and BioFresh has its own metadatabase that all partners must use to register their databases/datasets.

See other reports for the WP2 for more details.

If the database is made available on the web, access logs should be put in place.

Publish the structure and the metadata

It is recommended to publish a paper that describes the genesis of the database, its main general structure, the standards used, the controlled vocabularies, the ownership and the conditions of availability, etc.

As results, the content of data may be synthesized in terms of number of records more or less detailed by field and attribute (e.g., how many species per country, how many records for temperature ranges, etc.).

Also, example of usage of the database should be given, explaining the type of queries that are possible, the datasets that you can extract from, etc.

Making the database available on the web

The database itself should be disseminated together with the paper described in the previous section, a clear mention of Intellectual Property Rights and Copyrights, the way to cite the database.

References

Chapman, A.D., 2005. Principles of Data Quality, version 1.0. Report for the Global Biodiversity Information Facility. Copenhagen: GBIF. 58p. [www2.gbif.org/DataQuality.pdf]

Chapman, A.D. 2005. Principles and Methods of Data Cleaning – Primary Species and Species-Occurrence Data, version 1.0. Report for the Global Biodiversity Information Facility. Copenhagen: GBIF. 72p. [www2.gbif.org/DataCleaning.pdf]

Codd, E.F., 1990. The Relational Model for Database Management, Version 2 Addison-Wesley.

GBIF training manuals. [www.gbif.org/participation/training/resources/gbif-training-manuals/]

Google-Refine [code.google.com/p/google-refine/]