http://efi-plus.boku.ac.at/

Project no.: 0044096
Project acronym: EFI+

Improvement and spatial extension of the European Fish Index

Instrument: STREP
Thematic Priority: Scientific Support to Policies (SSP) - POLICIES-1.5

D2.1 – Central Database Description
D2.2 – Central Database Manual

Due date of deliverable: **End of month 10 (31.10.2007)**
Actual submission date of report: **13.12.2007**

Start date of project: **01.01.2007** Duration: **24 months**

Organisation name of lead contractor for this deliverable**:**

Swedish Board of Fisheries
Freshwater Research Institute
SE-178 93 Drottningholm
Sweden

Revision [1]

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	X
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)	

Contents

Deliverable 2.1 Central database description

Deliverable 2.2 Central database manual

The purpose of this document is to formally notify the European Commission of the first launch and the initial content of the EFI+ central database and central database manual.

Deliverable 2.1 Central database description

Description of work

General work process

Due to the fact that such a great amount of data will be kept within the database, the database has constantly changed in structure (see version history). The change of version is mostly related to the fact that it has been very important to keep the database normalized and secured, and to do so even when the amount of data has increased. The choice has also been to keep the database environment in MS Access for a more user friendly approach.

To secure data integrity and to assure that data is not change by mistake, there have been many quality controls on all work process levels:

1. Data collection and first upload process (M2-M10)
2. Updating (optimization) process (M10-M12)
3. Second upload process (M10-M12)
4. Database management after final upload (M12-M19).

These checks have been numerous and are therefore not possible to describe in detail. Though, an overview of the different aspects of work (with focus on data integrity) will be presented (figures 1 and 2).

Data collection and first upload process (M2-M10)

Data collection has been done between February and October 2007, the basic principles and procedures behind are documented in **tblmetadata** and **tblmeta_variable** in the database.

At first, input of data was partly controlled by preformatted Excel files and Access databases. These objects were distributed to the participants, so they could upload their data directly in the preformatted files (see EFI+ Excel Access Input Manual for more details).

Some problems occurred due to the fact that the formatting was erased during the copy and paste handling. Therefore, it was early comprehended that he preformatted Excel files were not a full measure to keep data integrity. The Access databases had better structures for data input, but it was not enough to secure data integrity (since input from Access forms took too long). Due to these problems, VBA-modules were developed for both Excel and Access users (see EFI+ VBA Manual). The modules checked for wrong values in all tables. Also it checked for empty cells and mismatches between runs (mismatches between tblcatch and tbllength). It was widely used, but it had a very time consuming run time for countries with very large datasets. During the later part of the work process, the VBA-modules were abandoned in favour to direct (and faster) checks within a test database structure.

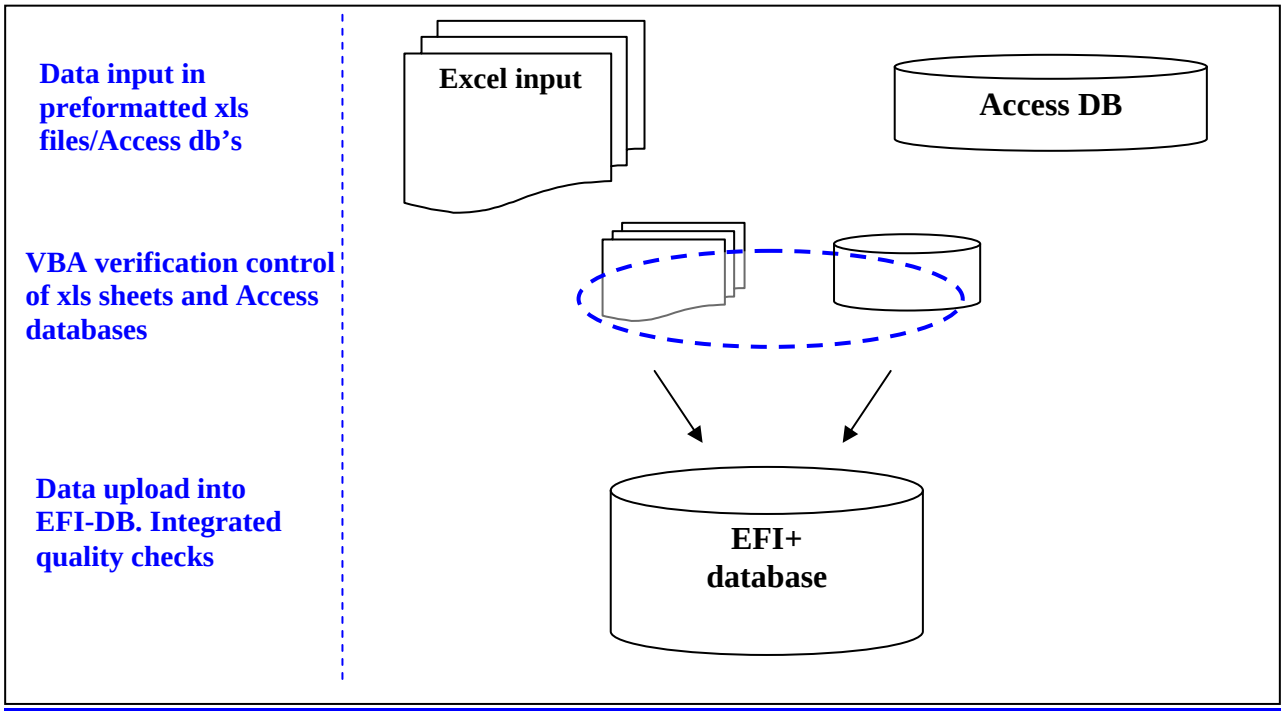

Figure 1. The general work process during first upload.

During the first upload of data (into EFI+ database version 3.1), upload and quality checking modules were initially redeveloped (since the rejection of the database modules in earlier version 2.1). This was done to better secure data input. The modules had to be changed during several times and were later abandoned. One of the main reasons for the rejection of the modules in version 3.1 was the high amount of formatting errors that occurred in excel (too complicated to be handled by the modules). Also, whenever a serious problem occurred the modules stopped working. All the formatting issues had to be dealt with manually. Therefore, other tools (like advanced union queries in SQL) were used instead.

After each fully checked data set, a status report was delivered from the DB team (to each participant). Then the data was mostly changed directly in the EFI+ database (by the DB team). Sometimes, when there were too many changes, a whole new data set was once again loaded into the EFI+ database. Between 3 and 4 status reports were distributed to each participant.

During the first upload process, missing data were discovered in a high amount of obligatory variables.

Updating (optimization) process (M10-M12)

After input into the database, the data sets were checked with statistical programmes (R and SPSS) in order to fully check for errors and missing values. These programmes could also check (by statistical means) for input faults, e.g. extreme values. To check for data inconsistencies, data were tested directly within the test databases. These test databases were also checking faulty values and empty cells. Preformatted queries and specific key structures in the database checked the different tables for mismatches and duplicates. The French team now showed statistics on the high amount of missing data. Therefore, during this second upload process, all participants were informed to complete all obligatory variables (missing data was not allowed in compulsory variable fields).

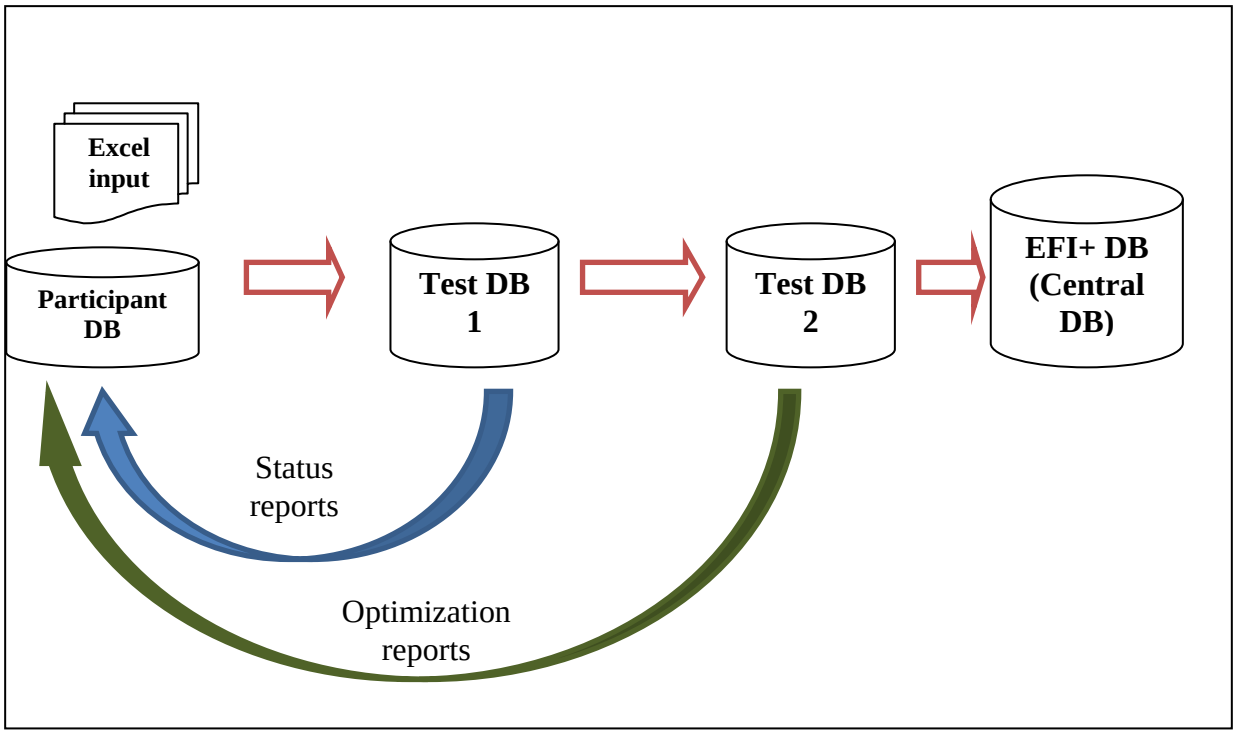

Figure 2. The general work process during the first and second data upload. Straight arrows indicate data flow from the participants to the database team (DB team). Curved arrows indicate the responses from DB team back to the participants.

After data controls in the test database, optimization reports were distributed to all participants for a second upload. To secure data integrity during upload a specific Access database were distributed to each participant (uploaded into the file manager). The database consisted of the most recent uploaded data by each country. By using a database structure (with intact primary keys in each table), the relation integrity were kept, which enhanced data quality. Some countries chose Excel update instead of Access. To minimize formatting errors and faulty values, each participant got an Excel outtake from each table in the updated EFI+ database.

Figure 2 shows the flow of incoming data from the participants, and the DB team's distribution of status and optimization reports. **Status reports** were mainly used in response to the participants' first data input, and showed structural inconsistencies and value faults, missing values and other errors. **Optimization reports** were the result of other types of data controls, as a response to the participants' updated data.

Test databases were established to check data consistency continuously during the whole work process. The test databases had a specific structure that tested import of data in a very restricted way, and the closer the data came to be uploaded into the final database the more restricted this structure became. Update and append queries forced data into tables with different restrictions considering key violations. If data were to pass, it also had to "fit" the different restricted tables. Each participant's data set was checked in several test databases.

Second upload process (M10-M12)

The second upload process started whenever each participant downloaded their new database from the EFI+ website (the file manager) and started to update their data according to the optimization reports. The optimization reports were compiled by the French, Austrian and Swedish teams. There have been good and quick responses from all the different participants. Later in the process it was also possible to mail updates to the database team. During this period there was a direct correspondence between the participants and the database team (figure 2).

All participants had to update their data sets, but some countries also had new data that would have to be added to their previous data sets. The DB team used different methods to check the newly added data and the updated data. The newly added data had all those errors that were revealed during the first update process. Once again, this was a very time consuming work. Table 1 shows the amount of records that have been included in the central database.

Database management after final upload(M12-M19)

This work process will be reported to the European Commission after M19.

Problems and solutions during work process

Many errors and some problems occurred during the work process. Much had to be manually solved, and this complicated the way data was handled. More time was spent to overcome faulty values, formatting errors etc. Partly, these problems also led to the rejection of the automated database modules in version 3.1.

Table 1. Number of records included in the central database's different tables.

	tblreporter	tblfishdata_owner	tbldiadromous	tblsite	tblfishing_occasion	tblcatch	tbllength (real numbers)*	tblmetadata
Austria	5	1	0	938	1172	6294	326039	46
Finland	1	1	257	530	530	2207	0	46
France	1	9	65700	1155	6570	62576	3896905	46
Germany	1	7	27240	803	1817	18543	648243	46
Hungary	1	1	246	193	193	2094	0	46
Italy	5	18	0	652	1152	4238	62847	46
Netherlands	1	4	11850	182	790	5903	135934	76
Poland	9	5	3480	919	978	6926	73140	414
Portugal	1	3	7384	923	923	45227	60431	47
Romania	2	1	0	263	323	1671	27722	46
Spain	7	34	10010	4252	5189	14092	233344	312
Sweden	2	1	7607	615	5652	16751	426826	92
Switzerland	2	29	0	717	969	2781	171583	46
UK	1	1	22134	1987	3162	16361	241111	48
Total	**39**	**115**	**155907**	**14129**	**29420**	**205664**	**6304125**	**1434**

*Real numbers (tbllength) indicate the amount of measured individuals (before normalizing the database). After normalisation the number of records diminished, but the number of individuals is still the same.

In the first updating process, there were mostly formatting errors that caused timely delays of data upload. Formatting errors were often hard to overcome, since the cut and paste techniques (used during data input) mixed different formats within the same columns. Date issues were especially hard to solve. Also, some variables (fields) were not able to upload correctly. There was faulty formatting also from Access input (e.g. mixture of date formats).With the help of several different database techniques, formatting errors was later solved.

Missing data and empty cells was first checked with the VBA control, but for countries with large data sets the VBA modules worked very slowly. Also, the VBA check was only used during the first update process. Generally the VBA control modules were widely used during data input, and to great results.

Missing data (as NoData or -999) was discovered in a high amount in compulsory variables. This was not allowed according to the EFI+ agreements in Bratislava. Therefore, during the second upload process, participants had to minimize the amount of missing data. After the second upload most countries had considerable minimized the amount of missing data in their data sets.

One serious matter was that the number of rows sometimes changed between different status reports. This showed that the participants sometimes deleted rows of data or (more seriously) added more rows to their data sets. This addition was often not announced, and sometimes created many hours to check and recheck again.

One serious problem that Access cannot solve easily is the issue with large and small letters. Other computer programs, e.g. SPSS, can easily distinguish character differences within a data set. The main problem is when characters are faulty within a primary key (that depends on other tables). These problems were not so many and were addressed by the French team.

During the work process, the manuals, input files, and database structures (e.g. data types, table values and field names) changed continuously. This was sometimes confusing, since there could be several distributed files in different versions, and some participants used the old ones instead of the newly updated ones. These problems became rarer later in the process, and there were generally good sensitivity among the participants (for major changes in database objects).

Database management and development process

To preserve data integrity, secure the data during upload procedure, and to harmonize the different objects within the database, it has been imperative to constantly change

the database structure. This has been done by constructing test databases, run program routines and specific queries in order to minimize structural corruption.

Database management consists of several different steps towards a complete and harmonized database:

1. Data input and quality assurance
2. Optimization of input data
3. Harmonization of database structure and its objects
4. Normalization of database structure
5. Storage and handling of data within the central database
6. Outtake of data

Data input and quality assurance

The EFI+ database deviated from the FIDES database structure (FAME project database, MS Access) since it was necessary to include new type of data: Large Floodplain River fish data, historical fish data, new natural descriptors and human pressure data, new fish guild classification, metadata, GIS-analysis calculations, and fish data ownership.

As mentioned before, an important sub-task is data quality assurance, as data are derived from many different sources. Data is partly checked in respect to standards defined in WP1, plausibility of data values and completeness of data sets. To assure data quality there has been a constant development of tools and objects to create a safer handling of data (input and storage). See Version history for more details.

Optimization of input data

Measures have been taken to minimize the amount of faulty data. The Swedish Team has been working in close contact with the BOKU and French Team in order to optimize the participant's data sets.

Harmonization of database structure and its objects

The database has to be harmonized with the collection of existing fish data from field surveys of participating countries according to field sampling methods defined in WP1. Also, harmonization was needed in order to integrate the work from other parts of the EFI+ project (e.g. calculated GIS data). A more detailed and harmonized metadata description has also been developed and integrated in the database structure.

Normalization of database structure

There is always a need to normalize the structure of a database in order to make it respond to queries in a logical way. Since the EFI+ central database is not going to be uploaded with more data, some difficult problems of normalization have now been solved. The final database is therefore optimized in this respect. The normalization process included adding a number field and regrouping all records in table length, and also erasing all duplicates within the tbllength. Therefore, the number of records in tbllength diminished, but the number of individuals is still the same. Earlier in the work process, there had been problems with relation integrity between table length and catch (exactly like in FIDES), see version history. This was urgently solved by finally normalize tbllength.

Storage and handling of data within the central database

See the EFI+ Central database manual in appendix for more details.

Outtake of data

This will be reported to the European Commission after M19.

Produced objects

Deliverables

- EFI+ Central database
- EFI+ Central Database Manual

Accessory objects for input from participants

The accessory objects for input from participants are not included in the previous specified deliverables. It includes developments of:

- Access input database (first input)
- Excel input files (first input)
- VBA module for self-generated quality control
- Optimised databases
- Optimized Excel files

Help manuals, instructions and reports

- **Help manuals.** Several help manuals have been produced to harmonize data management during the input, optimization and upload procedures. These manuals include:
  - EFI+ Central Database manual
  - EFI+ Excel Access Input Manual
  - VBA Manual

- **Instructions.** The following instructions were produced:
  - Instructions of how to use the file manager (for data upload)
  - Instruction for Access user,
  - Instructions for how to handle the data sets during the optimization process (Optimization process – Instructions)

- **Reports.** The following reports were produced:
  - Status reports for each participant (sometimes several reports per participant)

- Optimization reports for each participant (sometimes several reports per participant)
- Generalized reports to BOKU Team

Objects for database development

The objects for database development are also not included in the previous specified deliverables. It includes developments of:

- **VBA-modules**. Modules were developed in the previous versions of the central database, to include automatic processes for quality assurance of data sets during input. This especially concerns the modules developed for version 2.1 and 3.1 (see version history). These were developed but later rejected due to structural errors. Smaller modules were left intact in the version 4.1 (to help the form design of the existing central database.

- **Uploaded Access databases**. In the beginning of the second update process, the participants' data sets were uploaded in 11 different databases (one for each participant).

- **Uploaded Excel files.** In the beginning of the second update process, some of the participants' data sets were uploaded in 3 different zipped Excel folders (which included 8 Excel files – one for each table).

- **Test databases**. Test databases are used to check for faulty data and structural errors. The test environment had largely the same structure as the central database, but relations and tables were manipulated in order to check for data inconsistencies, value errors and missing values. Several different test db's (from loose to stricter relation integrities) were used, see general work process.

- **Special queries and procedures** for data input, quality assurance, and structural development were developed and used by the DB Team.

Database description

As previously stated, the database structure is based on the previous FIDES database structure. Though, we decided early to rebuild the database and skip some of the redundancies that were present in the Access version of FIDES. This is to enhance quality control and make administration easier. Also, during the development of the database structure, we encountered several issues concerning harmonization with the different objects in the database. Therefore, it has been a constant change of structure during the development of the database. A short description of the version history will follow.

It was important to manage the data partly according to the FIDES database structure, and keep some of its input structure (see “EFI+ Excel Access Input Manual.pdf”). Therefore, all the first input files (both in Excel and Access, preformatted by the participants) are still using the same old structure as before, but with differences concerning e.g. data types and field names. For the final database, though, we needed to normalise the database structure for better control of the incoming data, and also for more effective and correct storage of data.

Version history

Version 4.1 (existing version)

The existing version of the EFI+ database (version 4.1) has a more slimmed structure than earlier version. The heavy programmed modules are no longer within the structure, and this forced us to do more manual control work. Though an updated version of the VBA control (ver 4.1) helped us to better secure and preserve data integrity within the input data. Also, the French Team and BOKU team checked the data in special statistical programs (R and SPSS) for the optimization process (update process).

The structure (figure 3) now includes more tables (e.g. tblsite_gis and the different metadata tables), and has a better normalized structure than FIDES. Since the version 4.1 will not be uploaded with more data after the last upload, we could finalize the

normalization process. This leads to a better consistency between data objects within the database, than e.g. in FIDES.

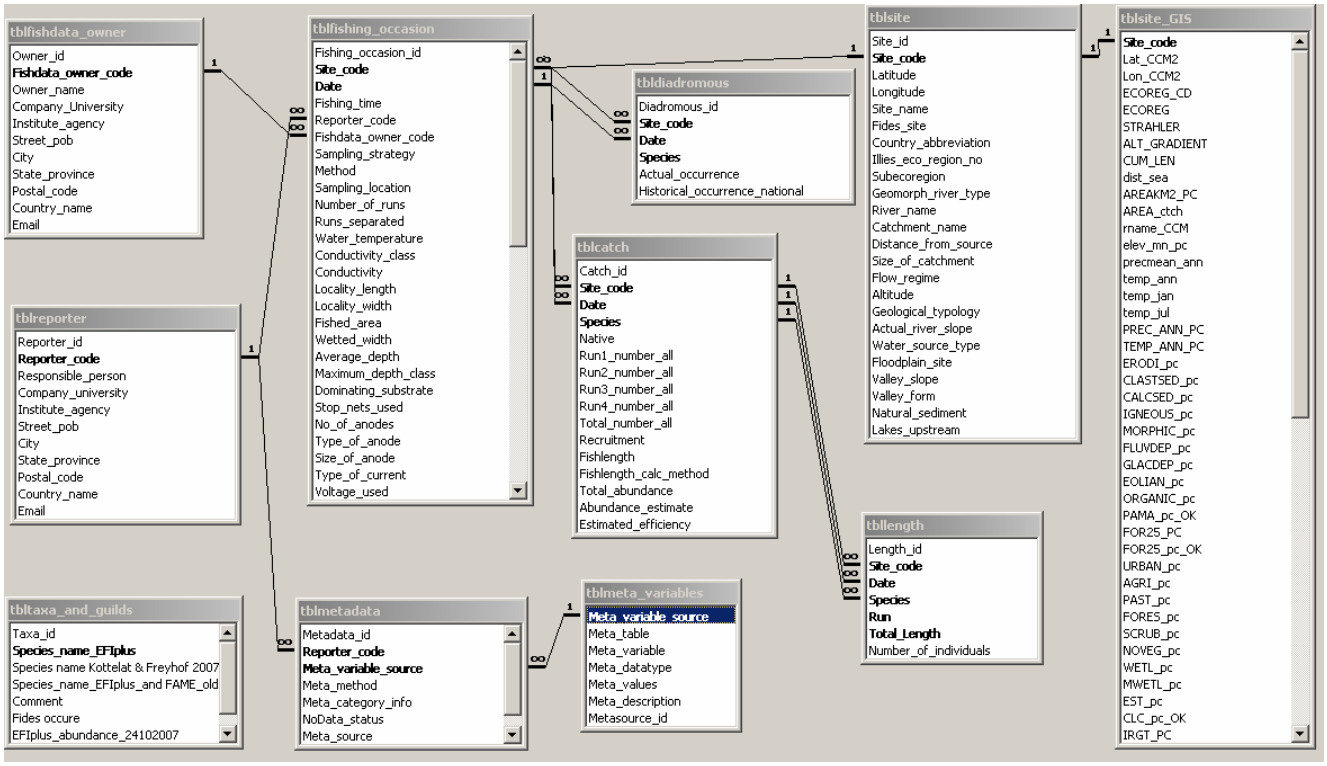

Figure 3. EFI+ (version 4.1) relation schedule in MS Access 2002.

Version 3.1 (smart modules for upload)

Version 3.1 also worked (like version 2.1) with the same data flow as shown in figure 5, though the structure was slightly changed from version 2.1. A simpler module approach was developed, and more tables were included / changed within the structure (figure 4.). Most of the changes from earlier versions was the programmed module, that also normalized the database (divided the incoming data into the different tables). This version also considered the possibility to make input at a later stage into the database, so different input forms and modules were again developed for this new version.

A VBA routine to control the input data at the source of input was developed. This had to be done since the modules that previously checked data quality (from ver 2.1) could no longer be used. The VBA routine was developed for both Excel and Access users.

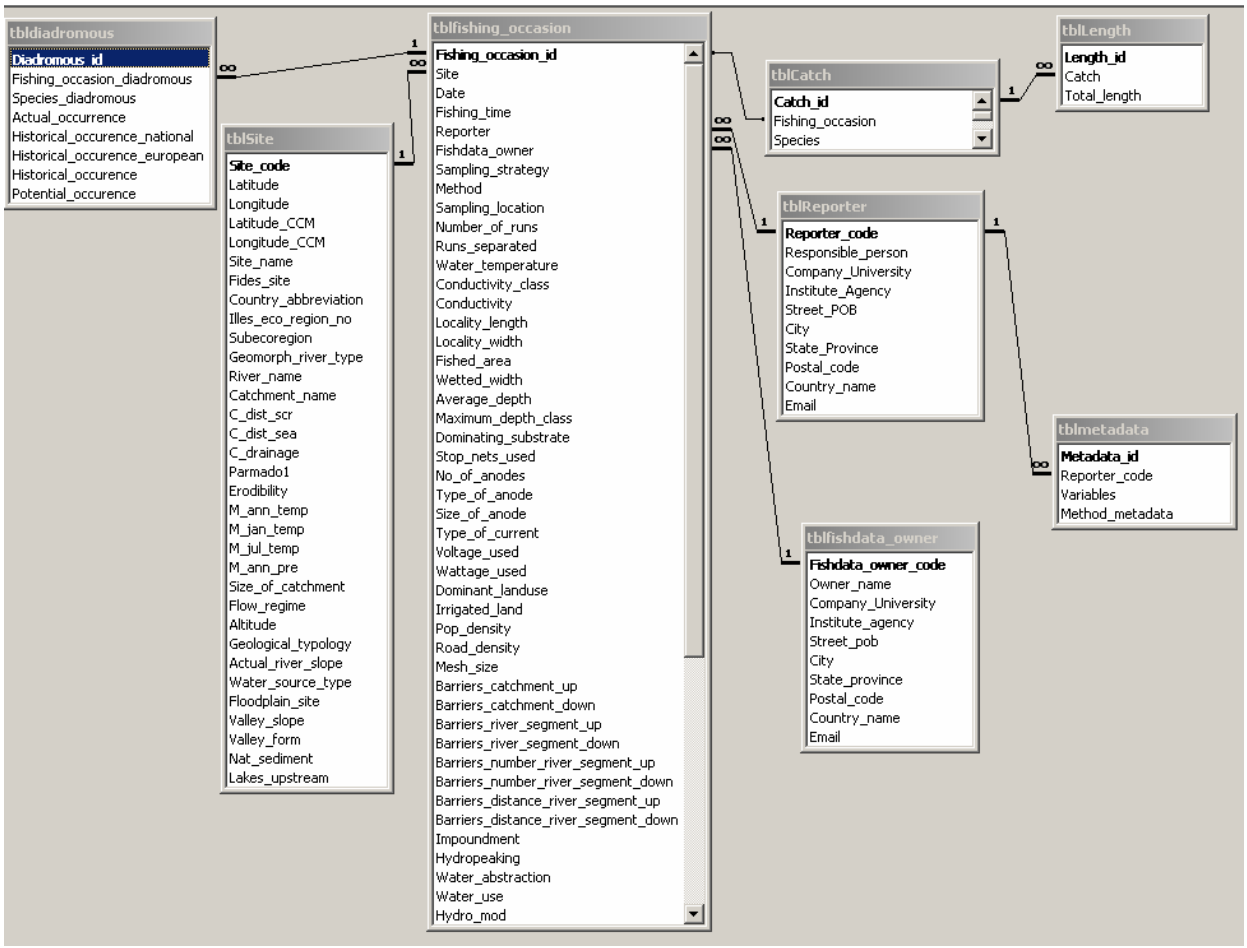

Figure 4. EFI+ (version 3.1) relation schedule in MS Access 2002.

Version 2.1 (full modules)

The database structure has been better adjusted for data input and outtake in version 2.1 (compared to FIDES), see figures 5 and 6. There is now a much simpler, and better normalized, structure (with e.g. single and not composite primary keys in the tables) in the database. The modules in the database include programmed updating of the different key values within the database (which FIDES lacked).

So, version 2.1 included heavy development of a module controlled environment (for better data security), which means that nearly no import of data was done manually. This enhanced data integrity and secured input data from faulty changes.

The general idea was to use a front-end database (EFIprog.mdb), and (by means of VBA controlled modules) transfer data packages from the front-end database to the

back-end (sharp database). The structure has earlier been tested for other databases with much less amount of data (figure 5).

However, there were difficulties to produce the right structure by using these heavy modules. The modules worked fine for smaller databases, and for better harmonized input data. Though, specifically due to the large amount of data, and also to the amount of formatting errors during data input, we needed to change the input modules radically. Therefore, we concentrated on another version (version 3.1), with completely new types of input modules.

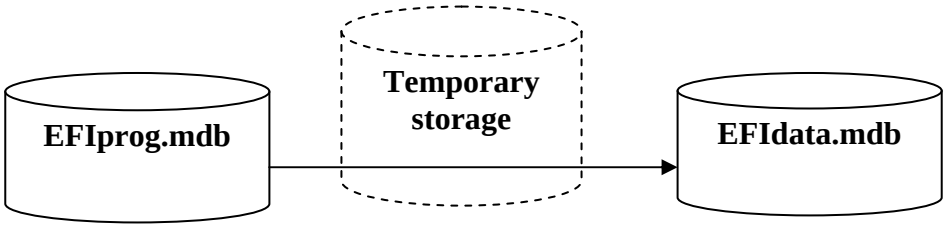

Figure 5. Description of data process from EFIprog.mdb and EFIdata.mdb in version 2.1.

Also, forms were developed to make direct input of data into the database (both from Excel and manually). This was later abandoned due to the large size of the database. It was also decided that the EFI+ database was not going to be used as an input database.

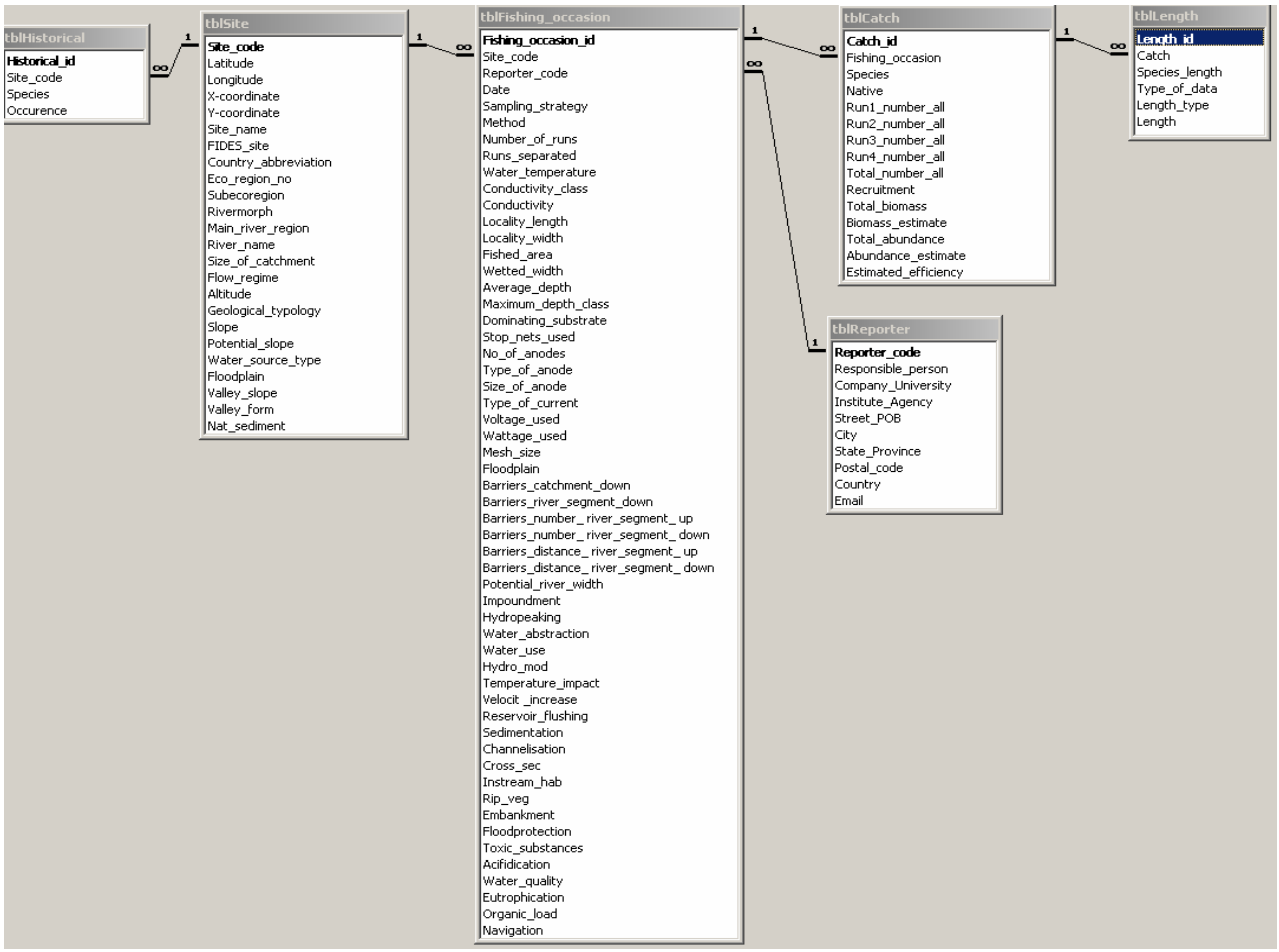

Figure 6. EFI+ (version 2.1) relation schedule in MS Access 2002.

2.1.4. FIDES (for comparison)

FIDES has a very simple, but good database structure. It is simple since its tables follow the design of the input files (Excel). Though, there are problems with this structure since it has not been normalized according to all the 4 basic normalisation rules.

According to the relation schedule for FIDES (figure 7), the length tables are not keyed (primary keys are missing). This will cause great problems when querying the database, and there will be difficulties to extract data from the database.

The contents of FIDES were later moved to a more stable and normalized platform (MS SQL Server).

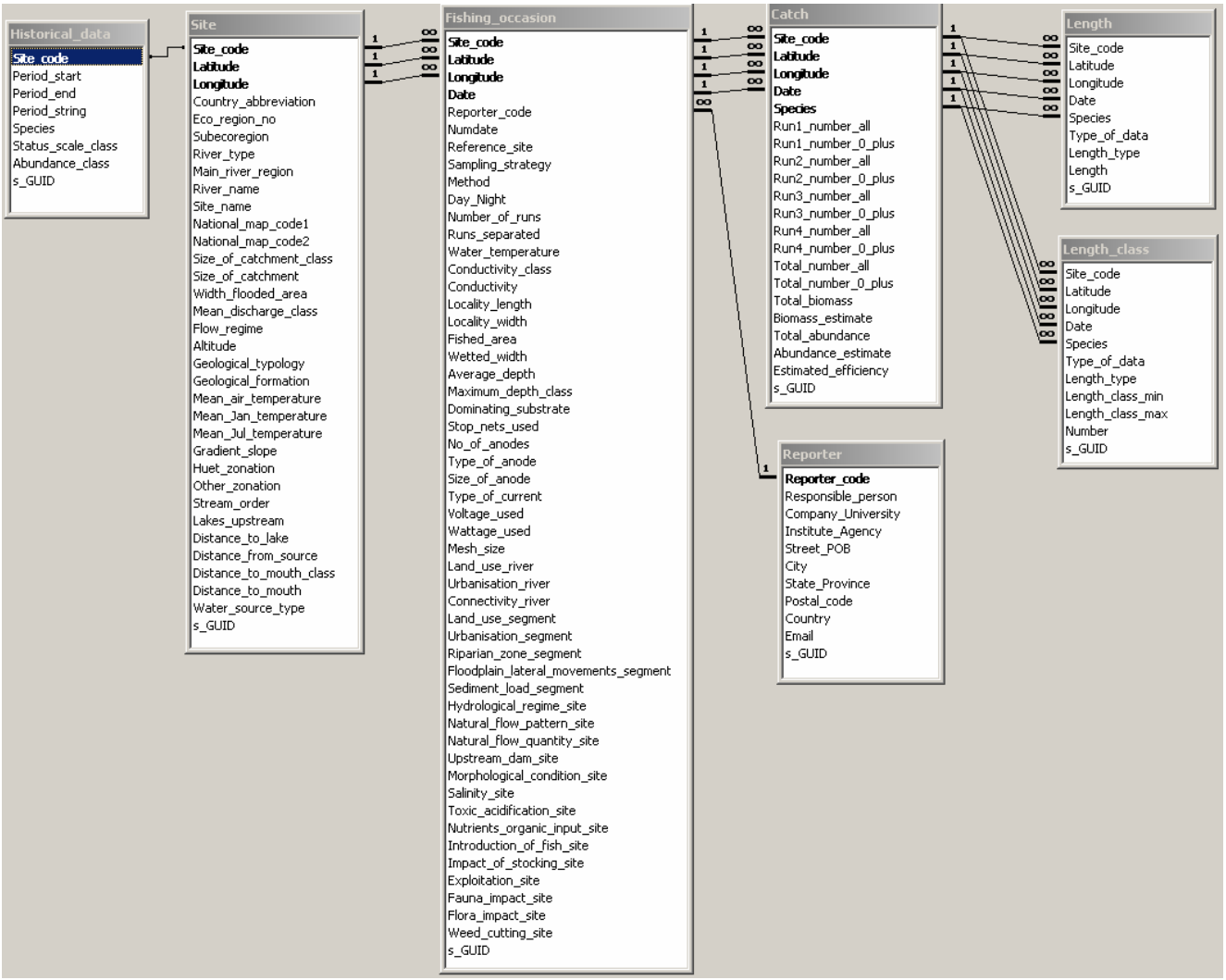

Figure 7. FIDES' relation schedule in MS Access 2002. Primary keys are missing in the two length tables.

Database structure - overview

An important part of the EFI+ database development has always been to protect the sharp data from corruption during data management. Therefore, it is normal procedure to divide the database into 2 different parts:

1. Front-end (where all modules, queries, and forms is located)
2. Back-end (where the sharp data is located)

By dividing the database in a front-end and back–end database, we are able to better control the change of sharp data and to simplify the structure. Also, there is a smaller chance to change sharp data by mistake. This will further facilitate the data management procedures for the database administrator.

Front-end database (EFIprog.mdb)

This unit handles forms and queries (there are no forms or queries in the back-end database at all). However, forms and queries have yet to be developed during the management process of the EFI+ central database. The possibilities to query and present data from the EFI+ database are nearly unlimited, and depend on the present questions at hand. Therefore, it is up to the database administrator to make the right queries and forms, and then present the data in accurate reports (this is not within the scope of WP2).

Figure 8. The start-up form.

The start-up form opens up the introduction form for the EFI+ database (figure 8). From this form you are able to manage the “sharp” (back-end) database. You are also able to search the back end database from this form. You handle the control of tables etc. with the menu bar on top of the start-up form (figure 9).

For further details of the front-end database, see EFI+ central database manual in appendix.

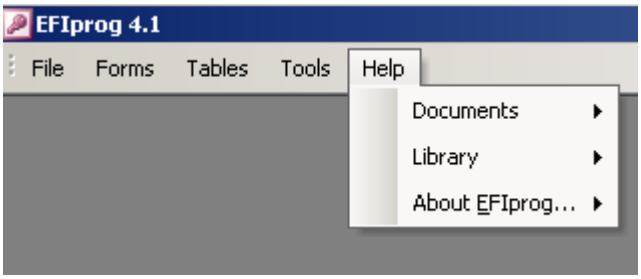

Figure 9. The menu bar on top of the start-up form.

Back-end database (EFIdata.mdb)

This database only consists of ”sharp” data, which is managed mainly from forms and queries in the front end database. The back-end does not contain any forms, reports or modules, only sharp data tables and help tables. In order to keep the sharp data intact, it is not good database manners to handle data directly within the back-end. Instead structures for data handling (e.g. queries, forms and reports) have to be built in EFIprog.mdb in order to export data from the EFIdata.mdb. Therefore, the sharp tables in the back-end (EFIdata) are linked to the front-end (EFIprog). Changes of table structures in EFIdata will therefore also change the table structures in EFIprog.

Locking the sharp back-end db (with a keyword), for security reasons, has been discussed. Unfortunately, this may cause problem for programming modules and during database update. The best control of unauthorized personnel is by creating authorization levels (which will be done by the database manager).

For further details of the front-end database, see EFI+ central database manual in appendix.

Overview of table structure and relations

Version 4.1 has increased the amount of integrated tables since the first version (2.1). It also has a normalized structure according to the newly integrated objects (tblsite_gis and the two metadata tables).

As shown in figure 3, the relations within the central database are linked between the tables with primary and secondary keys. It is important that each table have primary keys that can link other tables on lower (or higher) hierarchical levels. It is also important that all cells within each table's fields do not have a faulty value. Validity checks have been done on several occasions during the database development process, which also includes checking for empty fields. Otherwise the outtake of data will be very difficult, since Access can stop its queries when confronted with empty cells. The normalisation is partly about this, and this part of the normalisation has thereby been solved. Most of the redundancies from FIDES have been solved, but since the structure is similar to the FIDES structure (and input sheets) there are a few redundancies left. E.g. the fields "Date", "Site code" and "Species" are present in more tables than one. There are situations when a full normalised structure can be rejected, and this has been the issue of the EFI+ central database.

Summary and prospects of the central database

The EFI+ central database has been created continuously throughout the project. It has been a coordinated work process between all the participants, the coordinator team, and the responsible database team. The central database is providing the basic ecological and fish data, needed for the EFI+ project's analytical progress. The central database is the main data storage and access point for the project's complete data sets, for use by the project partners and the advisory group.

Deliverable 2.2 Central database manual

Introduction

This manual contains instructions on how to use the EFI+ central database (db). The central db is providing the basic ecological and fish data, needed for the EFI+ project's analytical progress. The central db is the main data storage and access point for the project's complete data sets, for use by the project partners and the advisory group.

Only the main events to describe the database (db) structure will be presented. All other db management instructions will not be discussed within this document, since it is considered as general knowledge for db administrators.

We have decided to divide the EFI+ central database in two parts, a front-end db and a back-end db. The front-end db is created for security reasons (so you do not change data by mistake too easily). Also, it is not good procedure to work directly in an environment of sharp data. Instead, you access all the sharp data by forms and queries in the front-end db. The queries then "ask" the specific tables in the sharp database, which send "answers" back to the front-end db (figure 1).

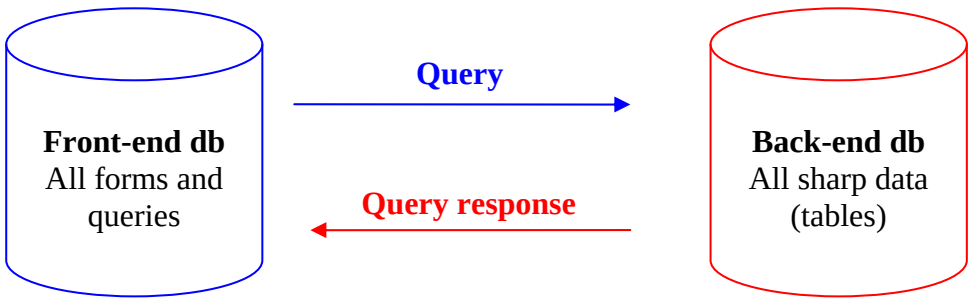

Figure 1. The EFI+ central database consists of two databases.

If you consider the front-end database difficult to understand and to use, you can of course use only the sharp database, and create all your queries etc in this db. Be reminded, though, that all your changes in the sharp db will not always be repairable. **Always make a copy** of the database before you start to query it or go through its sharp data.

Basic instructions

Character encoding and national language settings

Before you begin to use MS Access, we emphasise that you follow some instructions to enable use of standard national characters in the database's tables. Many of the TrueType fonts included in Microsoft Office 2002, supports a number of languages with different characters. **Arial Unicode MS** included in the Office package is a complete font containing all of the 40 000 alphabetic characters, ideograph characters and symbols which are defined in the Unicode standard 2.1. The EFI+ central database will be prepared using this font. If you did not install the font **Arial Unicode MS** when you installed Microsoft Office 2002 or another Office program, you can **reinstall Office** and choose add/remove functions. Press the plus sign (+) beside Office tools and then on the plus sign beside International support. After that, click the icon at Universal fonts and finally choose the alternative you wish. Reinstall Microsoft Office.

Also, to view the tables in the central database in a standardised way (which is necessary within the multi-lingual EFI+ project), please use the English national characters. In windows, you access the national characters by the control panel/National settings. Choose "English (Great Britain)". Otherwise there is a possibility to mix e.g. commas and points in number fields.

Database location

In order for the database programming routines (located in the front-end database) to work properly, the database has to be put on a specific location (**C:\EFI**). There is always a good idea to place the database on a network, for both security and access reasons. In doing so, you have to change the code within the front-end's modules (see Security and preservation of data integrity). For a description of important locations, take a look in the library (access from the menu-bar in the front-end database.

Note! If you choose not to use the front-end database, you do not need to put the database on a specific location. Therefore, if you only use the sharp database, you can place it wherever you want. Be reminded, though, that all your changes in the sharp db will not always be repairable.

Understanding the database

The database structure is based on the previous FIDES database. Though, we have decided to rebuild the database and skip some of the redundancies that were present in the Access version of FIDES. This is to enhance stability, quality control and make administration easier (figure 2).

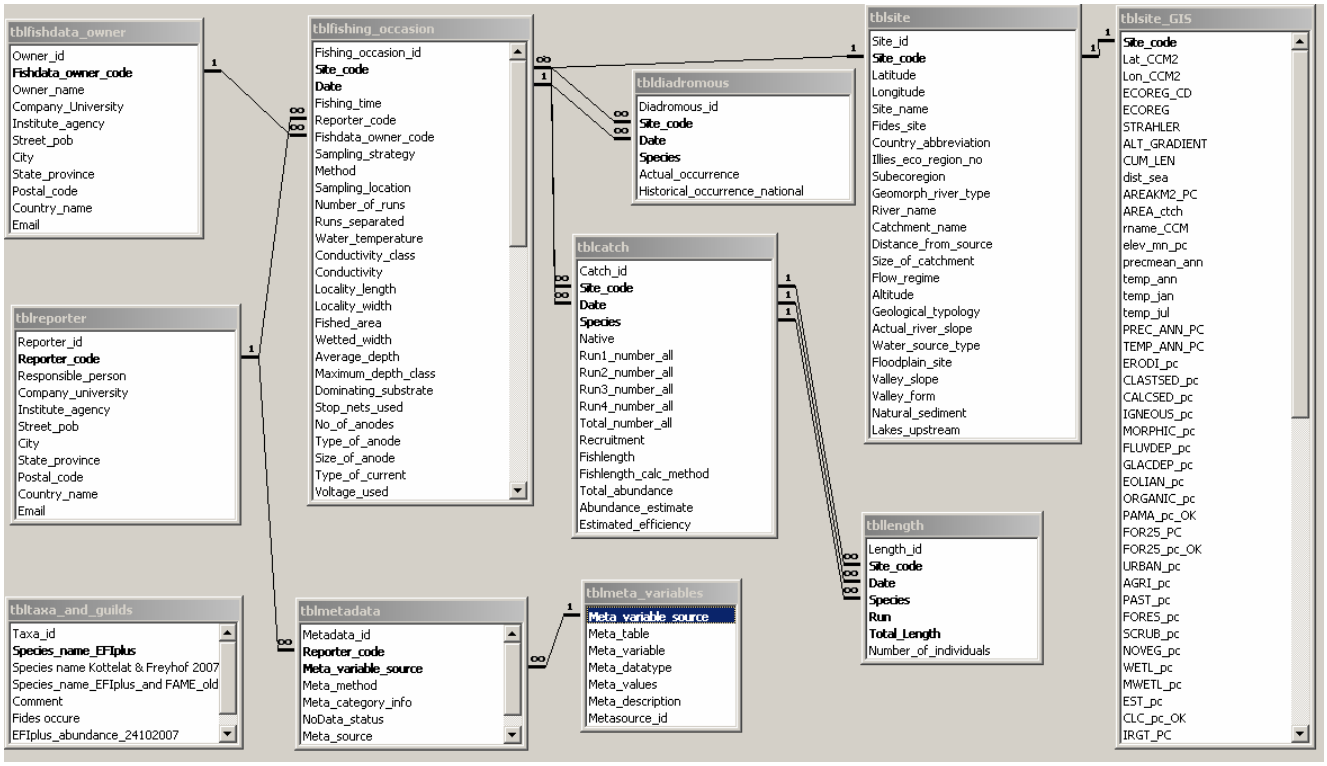

Figure 2. Table of relations in the EFI+DB (Access 2002). Primary keys in the tables are marked in bold.

Figure 2 shows the **relations** between all the tables in EFI+ central database. It also shows if the **relation types** are 1-n, n-1 or 1-1. You can also see which field that belongs to the **primary key** in each table (marked in bold). The primary key is vital for Access to know how to connect two (or several) different tables.

The EFI+ central database has been normalized, and all tables have **referential integrity with cascade deletion and update** (figure 2a) between each table. Due to this strict referential integrity, all the primary key data that are linked to the data you update or delete will also be affected. For example, if you delete a site code (primary key) in tblsite, all connecting fishing_occasions, catches, and lengths will also be

erased, since site code is also a primary key in these tables. Another example is if you change a species name in tblcatch (primary key), the same species name will also change in tbllength (if the two records also are connected by site code and date). By using the referential integrity, much is simplified concerning updates and deletions in the db's primary keys. But remember to be careful whenever you change sharp data.

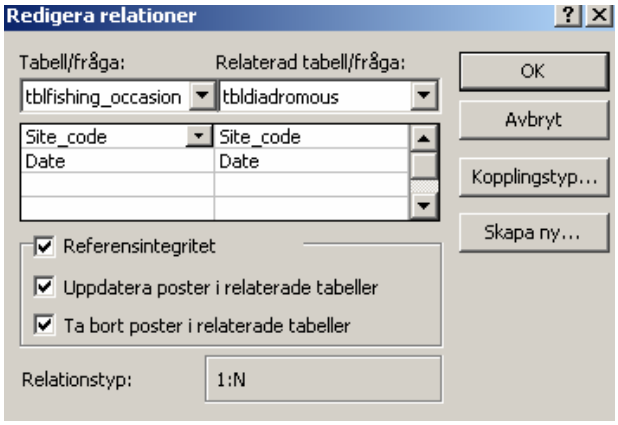

Figure 2a. The reference integrity between tblfishing_occasion and tbldiadromous.

Note! Be careful when deleting or updating sharp data. Due to the **referential integrity**, deletions or updates will also be made in related tables.

An important part of the EFI+ database development has always been to protect the sharp data from corruption during data management. Therefore, it is normal procedure to divide the database into 2 different parts.

The **front-end** database (**EFIprog.mdb**) handles all the forms, queries, reports etc. It is good procedure to manage all the sharp data using automated forms and queries.

The **back-end** database (**EFIdata.mdb**) contains all the tables with "sharp" data. It is not recommended that users handle data directly within the back-end db. Instead all database administration should be done from the front-end db.

By dividing the database in a front-end and back–end database, we are able to better control the changes of sharp data and to simplify the structure between the different database objects (tables, queries, forms, reports, macros and modules). Also, this will further facilitate the data management procedures for the database administrator.

The front-end database (EFIprog.mdb)

If you choose to use the front-end database you strictly work with forms and queries. All tables from the back-end db are linked to the front-end db. This means that you can query the tables with no restrictions. Though, you cannot change the structure (data type etc.) of the back-end db's tables from the forms in the front-end. Preformatted forms, queries and a menu will give you an idea of how you can use the front-end db more effectively.

Forms in the front-end db

- **Start-up form.** When you open the front-end db (**C:\EFI\EFIprog.mdb**), the start-up form introduces you to the EFI+ central database (figure 3). From this form you can manage the "sharp" (back-end) db. Please note that you have to keep the EFIprog.mdb and EFIdata.mdb on the same location (**C:\EFI**).

Figure 3. The start-up form in the EFI+ central database.

- **Example query form.** The control button “More info” on the start-up form opens up a preformatted query form, which gives you an example of how powerful a combination of forms and queries can be (figure 4).

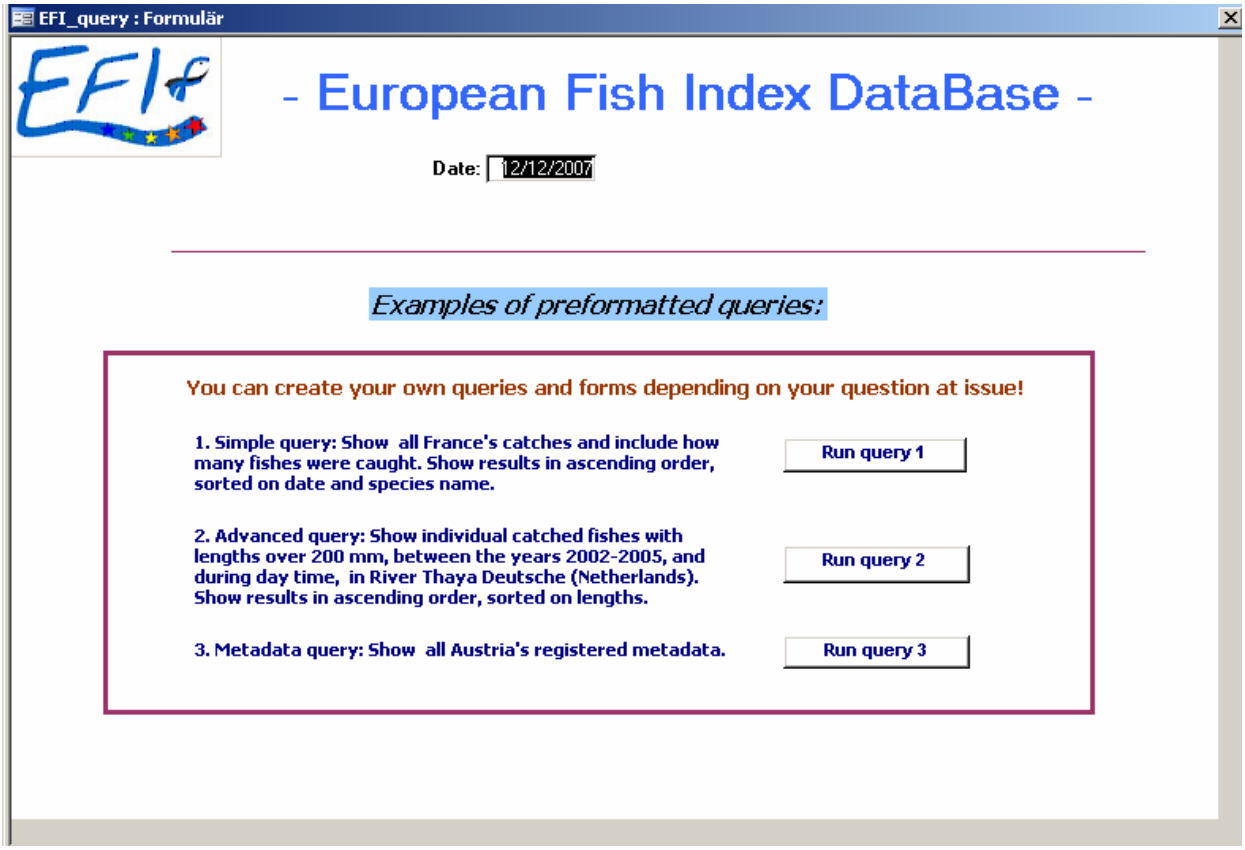

Figure 4. The example query form in the EFI+ central database.

Queries in the front-end db

By creating queries in the front-end db, you can directly view, change, update or delete sharp data in the back-end db. In the example query form (figure 4), there are 3 different queries already preformatted:

1. A simple query that shows France’s catches, which includes all caught fishes.
2. An advanced query that shows individual catched fishes with lengths over 200mm, between the years 2002-2003, during day time in river Thaya Deutshe in the Netherlands.
3. A simple metadata query that shows Austria’s registered metadata (all their filled out variables).

When we specifically check the details of the different queries we also discover the power of queries. The first very simple query uses data from several different tables (figure 5). Also, we restrict the query to only choose the site codes that begins with FR* (all sites from France). If we wanted we could also restrict the query by using the same restriction in the field Country_abbreviation in tblsite. The query uses the same relations that are in the central database's structure (figure 2).

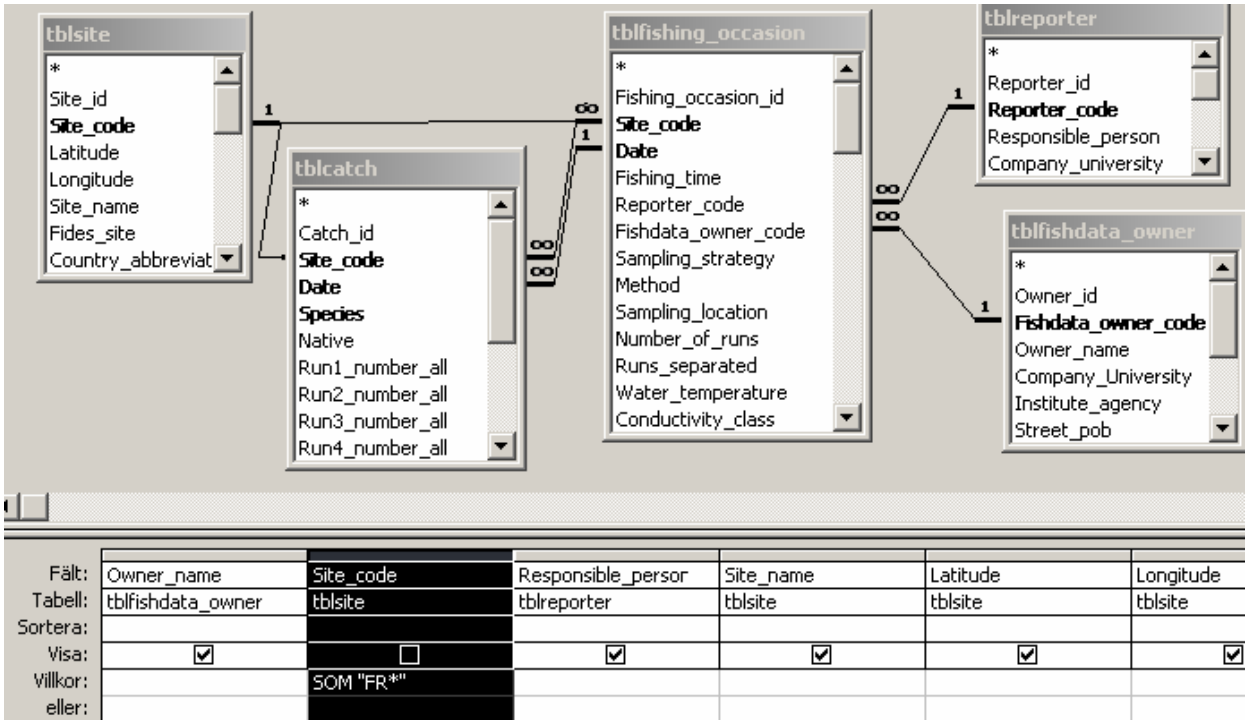

Figure 5. Example of the preformatted simple query in the example query form.

Note! If tbllength had been in the same simple query the result would consist of fewer records, since tbllength have fewer records than tblcatch.

Menu in the front-end db

By using the menu bar, you expand your controls within the front-end db. The menu bar is located on top of the front-end database window (figure 6).

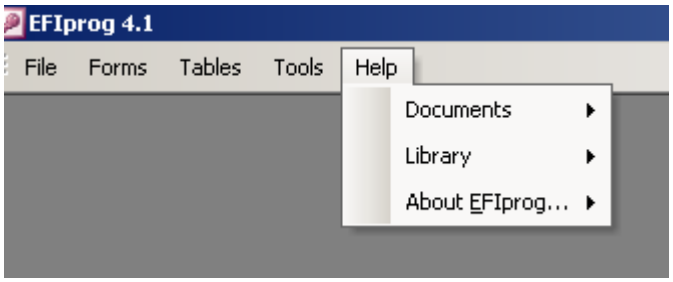

Figure 6. The menu bar on top of the EFIprog (front-end db) expands your controls.

The menu handles the following

- **File.** Closes the database.
- **Forms.** Consists of the 2 forms that you can move between (start-up form and query form.
- **Tables.** Here you can look through all the help and head tables within the sharp back-end database.

  Note! Do not change the data in the tables, since it will then also be changed in the sharp db. To lock these tables from editing, you have to do a “snapshot view”.

- **Tools.** Use tools to compress and repair the database.

  Note! You also have to compress and repair the back-end db if you change in a lot of tables.

- **Help.** Consists of fast access to **documents** (EFI+ central database manual.pdf). You have to put the document on the following location if you will be able to use the Help/Documents button **C:\EFI\Document.** Otherwise the database module will not find the right document. The **library** on the help menu shows the location of where to put all your files (figure 7). E.g. you must put all your databases on the location **C:\EFI**, and all the documents on

tblLibrary : Tabell

Object	Location
EFIprog.mdb	C:\EFI\EFIprog.mdb
EFIdata.mdb	C:\EFI\EFIdata.mdb
EFI Central Database Manual	C:\EFI\Document\EFI+ Central Database Manual.pdf
notebook (txt files)	C:\WINDOWS\system32\notepad.exe

Figure 7. The Library table shows where to put all your different files.

- **C:\EFI\Document**. **About EFIprog…** gives a short description of the version of the database, when it was built and by whom.

> **Note!** You must put all documents on the location C:\EFI\Document, and use the same name as the documents have during the distribution of the database. Otherwise the database will not find the correct document.

The back-end database (EFIdata.mdb)

The back-end database only consists of ”sharp” data. The sharp tables in the back-end db (EFIdata) are linked to the front-end db (EFIprog), and changes of table structures in EFIdata will therefore also change the table structures in EFIprog. The EFIdata.mdb does not contain any forms, reports or modules, only sharp data tables and help tables.

The sharp database (EFIdata.mdb) consists of 11 tables (figure 8). Two of these tables (tbltaxa_and_guilds and tblmeta_variables) are so called “help tables” with no specific function other than to describe the other tables’ data. There are 9 head tables (with specific contents relating to their names (tblsite_gis, tblsite, tblreporter, tblfishdata_owner, tblmetadata, tbldiadromous, tblfishing_occasion, tblcatch, and tbllength). See figure 2 for a detailed description of the tables’ relations.

The relations and tables have already shortly been described, and each table will be described in more detail below.

It is very important that the EFI+ central database is a “sealed and delivered” database, which means that it will not be updated with more data after the final upload. This is important because some parts of the normalization procedure depend on this “rule”. The db is not locked or anything, but you have to carefully follow specific routines if new data has to be uploaded. The main events to consider (for the new data set) are:

1. Basic checks of data integrity and consistency
2. Advanced checks of data integrity and consistency
3. Normalisation of tbllength
4. Control of metadata
5. Upload and checks against existing tables within the central database.

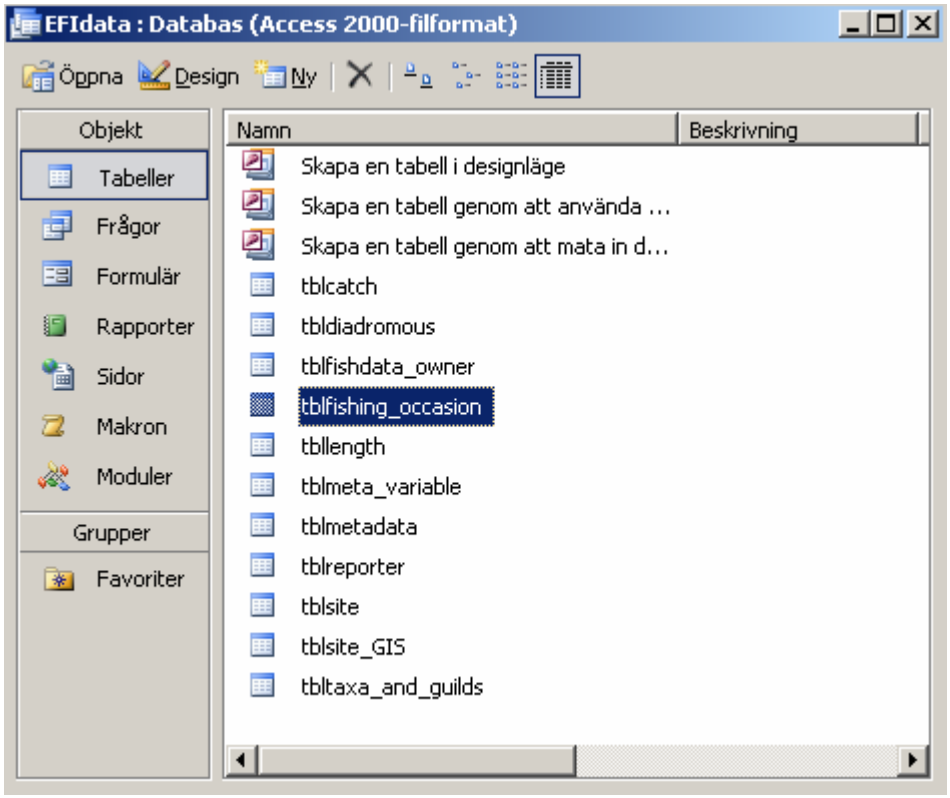

Figure 8. The 9 head and 2 help tables in EFIdata.mdb.

It is not in the scope of this manual to describe these advanced routines. Instead, every database administrator that manages the EFI+ central database have to go through the db structure and make adjustments accordingly.

Security and preservation of data integrity

Locking the EFIdata.mdb (with a keyword) has previously been discussed, but this may cause problem for programming modules and during database update. The best control of unauthorized personnel is by creating authorization levels (which will be done by the database manager).

One important security issue is where to locate the central database. If the database manager/administrator needs to let every worker within a specific institution query the db without changing any data, he/she could place the db on the network and distribute specific access rights.

The database team recommends that the front-end db is used to manipulate data in the back-end db. Therefore, there is a need to change the location of the database within the code. Remember that the location in the Library (a table in the EFIprog, see figure 7) also have to be changed to the new location. The code in the module "mdlallmanna_procedurer" (within the front-end db) has to be changed in the following places:

1. In public sub Tilldela_sokvag
2. In public function Open_Final_EFI_manual

Description of tables

Below follows a short introduction to every table in the central database (field name, data type, table description).

tblcatch

Field Name	Data Type	Description
Catch_id	AutoNumber	Counter id
Site_code	Text	Same site code as in tblfishing_occasion
Date	Date/Time	Same date as in tblfishing_occasion (DD/MM/YYYY)
Species	Text	Scientific name of species (tbltaxa_and_guilds)
Native	Text	Species native or not at the site (Yes, No; NoData)
Run1_number_all	Number	All caught individuals (incl 0+) of the species in run 1
Run2_number_all	Number	All caught individuals (incl 0+) of the species in run 2
Run3_number_all	Number	All caught individuals (incl 0+) of the species in run 3
Run4_number_all	Number	All caught individuals (incl 0+) of the species in run 4
Total_number_all	Number	All caught individuals (incl 0+) of the species in all runs (cumulative total)
Recruitment	Text	If the species reproduce in the segment (Yes, No; NoData)
Fishlength	Text	If fish length have been measured (and put into Length.xls). Yes/No
Fishlength_calc_method	Text	Method used to calculate fish length (All, Classes, Subsample, Min-max, Nodata)
Total_abundance	Number	Estimated abundance (no. of individuals) of the species per hectare (10 000 m2)
Abundance_estimate	Text	Method used for the abundance estimation.
Estimated_efficiency	Number	State the catch efficiency (p-value)

tbldiadromous

Builder Name	Data Type	Description
Diadromous_id	AutoNumber	Counter id
Site_code	Text	From tblfishing_occasion
Date	Date/Time	From tblfishing_occasion (DD/MM/YYYY)
Species	Text	Diadromous species according to list. Scientific name of species (see selected diadromous species list)
Actual_occurrence	Text	Actual occurrence of the selected diadromous species (Absent, Naturally, Stocked, NoData)
Historical_occurrence_nationa	Text	The historical occurence of diadromous species for each fishing occasion (Yes, Unknown)

tblfishdata_owner

Field Name	Data Type	Description
Owner_id	AutoNumber	Counter id
Fishdata_owner_code	Text	Your own personal code (e.g. ES0001)
Owner_name	Text	Name of person who ownes the fish data
Company_University	Text	Optional. Company or university connected to the owner name
Institute_agency	Text	Name of institute or governmental agency who owns the fish data.
Street_pob	Text	Optional. Street address or post box number
City	Text	Name of city
State_province	Text	State name
Postal_code	Text	Country code + postal code (e.g. SE 116 62)
Country_name	Text	Country in English
Email	Text	Email address to the data owner

tblfishing_occasion

Field Name	Data Type	Description
Fishing_occasion_id	AutoNumber	Counter ID
Site_code	Text	Same site code as in tblsite
Date	Date/Time	Date of sampling given as Microsoft date format (DD/MM/YYYY).
Fishing_time	Text	Time of fishing (Night, Day, Nodata)
Reporter_code	Text	Same as reporter code in tblReporter
Fishdata_owner_code	Text	Same as fishdata owner code in tblfishdata_owner
Sampling_strategy	Text	Define how the section was sampled (Whole, Partial1bank, Partial2banks, Partialrandom, Partialprop, Other, NoData)
Method	Text	Define how electric fishing was carried out (Wading, Boat, Mixed, NoData)
Sampling_location	Text	Where the sampling site is situated in relation to the river (Main channel, Backwaters, NoData)
Number_of_runs	Number	Number of runs (passages, removals) carried out
Runs_separated	Text	Is the catch reported separately for each run or as a total (Separated, Total, NoData)
Water_temperature	Number	The temperature of the water (depth 0.2-2 m) at sampling (degrees Celcius)
Conductivity_class	Text	Specific conductivity of the water at sampling. Given as mS/m (<10, <50,<500, >500, NoData)
Conductivity	Number	Specific conductivity value of the water at sampling (mS/m.)
Locality_length	Number	Fished length (sampled length) of the stream (m)
Locality_width	Number	Estimated fished width of sampled locality (m)
Fished_area	Number	Area of the section that has been definitely sampled (sampled length * sampled width) given in m2.
Wetted_width	Number	Wetted width in meters is normally calculated as the average of several transects across the stream.
Average_depth	Number	Average depth (m)
Maximum_depth_class	Text	Maximum depth in metres for the surveyed river section (<1, <2, <5, >5, NoData)
Dominating_substrate	Text	The dominating substrate type at the sampled area of the stream (Organic, Silt, Sand, Gravel/Pebble/Cobble, Boulder/Rock, NoData)
Stop_nets_used	Text	State if stop nets (block nets) were used to delimit the sampling site (Yes, Up, No, Down, NoData)
No_of_anodes	Text	Number of anodes used (One, Two, Three, Four, Multiple, NoData)
Type_of_anode	Text	Type of anodes used (Ring,Rectangular,Boom, Other, NoData)
Size_of_anode	Number	The maximum length of the anode (cm)
Type_of_current	Text	The type of current used for fish sampling (AC, DC, PDC, NoData)
Voltage_used	Number	The voltage used (Volt)
Wattage_used	Number	The wattage used (Watt)
Mesh_size	Text	Mesh size (not stretched net) of the net used for sampling fish (mm)
Barriers_catchment_down	Text	Presence of downstream barriers on the catchment scale (No, Yes, Partial, Nodata)
Barriers_river_segment_up	Text	Presence of upstream barriers on the segment scale (No, Yes, Partial, Nodata)
Barriers_river_segment_down	Text	Presence of downstream barriers on the segment scale (No, Yes, Partial, Nodata)
Barriers_number_river_segment_up	Number	Number of barriers upstream
Barriers_number_river_segment_down	Number	Number of barriers downstream
Barriers_distance_river_segment_up	Number	Distance to next upstream barrier in the segment (km)
Barriers_distance_river_segment_down	Number	Distance to next downstream barrier in the segment (km)
Impoundment	Text	Natural flow velocity reduction on site due to impoundment (No, Weak, Strong; NoData)
Hydropeaking	Text	Site affected by hydropeaking (No, Yes, Partial, Nodata)
Water_abstraction	Text	Is the site affected by water flow alteration/minimum flow (No, Weak, Strong; NoData)
Water_use	Text	Main purpose for water usage (No, Hydropower, Irrigation, Drinking water, Industrial water, Snow production, Fish ponc
Hydro_mod	Text	Seasonal hydrograph modification due to hydrological alteration (Yes, No, NoData)
Temperature_impact	Text	Is there an impact on water temperature (No, Permanent increase, Permanent decrease, Summer increase, Summer dec
Velocity_increase	Text	impact on flow conditions (mean velocity) due to channelisation, floodprotection, etc (Yes, No, NoData)
Reservoir_flushing	Text	If the fish fauna affected by flushing of reservoirs upstream of the site (Yes, No, NoData)
Sedimentation	Text	Input of fine sediment (No, Weak, Medium, High, NoData)
Channelisation	Text	Alteration of natural morphological channel plan form; intensity of straightening (No, Intermediate, Straightened, NoDa
Cross_sec	Text	Alteration of cross section (No, Intermediate, Technical crossec/U-profile, Nodata)
Instream_habitat	Text	Alteration of instream habitat conditions (No, Intermediate, High, Nodata)

Field Name	Data Type	Description
Instream_habitat	Text	Alteration of instream habitat conditions (No, Intermediate, High, Nodata)
Riparian_vegetation	Text	Alteration of riparian vegetation close to shoreline (No, Slight, Intermediate, High, Nodata)
Embankment	Text	Artificial embankment (No, Local, Continous permeable, Continous no permeability, NoData)
Floodprotection	Text	Presence of dykes for flood protection (Yes, No, NoData)
Floodplain	Text	If the river has a former floodplain - Proportion of connected floodplain still remaining (No, Small,Medium, Large, NoDa
Toxic_substances	Text	Category of toxic substances (No, Intermediate, High concentration, NoData)
Acidification	Text	Acidification (Yes, No, NoData)
Water_quality_index	Number	National water quality index (1-5)
Water_quality_name	Text	Name of the water quality (e.g. Good)
Eutrophication	Text	Artificial eutrophication (No, Low, Intermediate, Extreme, NoData)
Organic_pollution	Text	If organic pollution is observed (No, Weak, Strong, NoData)
Organic_siltation	Text	If organic siltation is observed (Yes, No, NoData)
Navigation	Text	Navigation intensity (No, Low, Medium, Strong, NoData)
Colinear_connected_reservoir	Text	Presence, absence of fish farms etc.

tbllength

Field Name	Data Type	Description
Length_id	AutoNumber	Counter id
Site_code	Text	Same as in tblcatch
Date	Date/Time	Same as in tblcatch (DD/MM/YYYY)
Species	Text	Corresponding to species in tbltaxa_and_guilds)
Run	Number	Indicate from which run (see Catch.xls) the measured fish was catched
Total_Length	Number	Length of individual in mm (NOT fork length)
Number_of_individuals	Number	Number of individuals of each total length (from a specific run)

tblmeta_variable

Field Name	Data Type	Description
Meta_variable_id	AutoNumber	Counter id
Meta_variable_source	Text	Variable name, unique and indexed (as named in tblmetadata)
Meta_table	Text	Table name of the variable
Meta_variable	Text	Variable name in the table
Meta_datatype	Text	Type of metadata (environment data, pressure data etc) that variable belongs to
Meta_values	Text	The different values from each variable
Meta_description	Text	Variable description in text
Metasource_id	Number	ID number from xls table

tblmetadata

tblmetadata

Field Name	Data Type	Description
Metadata_id	AutoNumber	Counter id
Reporter_code	Text	Same as reporter code in tblreporter
Meta_variable_source	Text	Specific variable name (same as in tblmeta_variable)
Meta_method	Text	Indicates which method has been used for deriving the
Meta_category_info	Text	Indicates how single categories have been compiled
NoData_status	Text	Indicates in which cases the category NoData has been used
Meta_source	Text	Indicates which sources that have been used to derive the variable
Meta_Editor	Text	Name of person who edited the metadata.

tblreporter

Field Name	Data Type	Description
Reporter_id	AutoNumber	Counter id
Reporter_code	Text	Your own personal code (e.g. FR0001)
Responsible_person	Text	Name of person responsible for data supply
Company_university	Text	Name of company/Univeristy if applicable
Institute_agency	Text	Name of institute or governmental agency
Street_pob	Text	Street address or post box number
City	Text	Name of city
State_province	Text	State name
Postal_code	Text	Country code + postal code (e.g. SE 116 62)
Country_name	Text	Country in English
Email	Text	Email address of the responsible person

tblsite

Field Name	Data Type	Description
Site_id	AutoNumber	Counter id
Site_code	Text	Country abbreviation + your own code of the site, e.g. GE0001
Latitude	Number	Latitude WGS84 referenced to CCM-rivernetwork
Longitude	Number	Longitude WGS84 referenced to CCM-rivernetwork
Site_name	Text	Your national name identifying the site (free text)
Fides_site	Text	Site occurs in FIDES database (Yes/No)
Country_abbreviation	Text	Country abbreviation (according to table)
Illies_eco_region_no	Number	Ecoregion according to Illies table
Subecoregion	Text	National ecoregions (free text)
Geomorph_river_type	Text	Information in 4 categories to be selected (Braided, Sinous, Meand regular, Meand tortous, Naturally constraint no mob, NoData
River_name	Text	National name of the river (free text)
Catchment_name	Text	English name of the river catchment (watershed). Do not use a numeric code, use the full name.
Distance_from_source	Number	Distance from source in kilometers to the sampling site measured along the river.
Size_of_catchment	Number	Absolute size of the catchment upstream of sampling site (km2)
Flow_regime	Text	Normal flow pattern for the river (Permanent, Summer dry, Winter dry, Intermittent, Nodata)
Altitude	Number	The altitude of the site in metres above average sea level
Geological_typology	Text	According to WFD (Siliceous, Calcareous, Organic, NoData)
Actual_river_slope	Number	Slope of stream bed along stream expressed as per mil (o/oo)
Water_source_type	Text	The source of the river water (Glacial, Nival, Pluvial, Groundwater, NoData)
Floodplain_site	Text	Presence of a former floodplain (Yes, No, NoData)
Valley_slope	Number	Length of the valley between two contour lines
Valley_form	Text	Form of the valley (V-shape, Gorges, U-shape, Plains, Nodata)
Natural_sediment	Text	Naturally dominant sediment Information in five categories (Organic, Silt, Sand, Gravel/Pebble/Cobble, Boulder/Rock, NoData)
Lakes_upstream	Text	Are there natural lakes present upstream of the site? (Yes, No, NoData)

tblsite_gis

Field Name	Data Type	Description
Site_gis_id	AutoNumber	Counter id
Site_code	Text	Country abbreviation + your own code of the site, e.g. GE0001
Lat_CCM2	Number	Latitude WGS84 referenced to CCM2-rivernetwork
Lon_CCM2	Number	Longitude WGS84 referenced to CCM2-rivernetwork
ECOREG_CD	Number	Code for ecoregions, accord. to WFD
ECOREG	Text	Name of ecoregion
STRAHLER	Number	Strahler order of CCM2 river segment
ALT_GRADIENT	Number	Relief energy of the CCM river segment (from-to/segm length)
CUM_LEN	Number	Cumulated length of upstream river network
dist_sea	Number	Distance to sea along CCM river network
AREAKM2_PC	Number	Area draining to segment = primary catchment between tributaries
AREA_ctch	Number	Area drained by segment (upstream area + primary catchment)
rname_CCM	Text	River name in CCM2 database
elev_mn_pc	Number	Mean elevation of primary catchment
precmean_ann	Number	Mean annual precipitation at the site
temp_ann	Number	Mean annual temperature at the site

Field Name	Data Type	Description
temp_jan	Number	Mean january temperature at the site
temp_jul	Number	Mean july temperature at the site
PREC_ANN_PC	Number	Mean annual precititation in primary catchment
TEMP_ANN_PC	Number	Mean annual temperature in primary catchment
ERODI_pc	Number	Weighted mean of erodibility in primary catchment
CLASTSED_pc	Number	Consolidated-clastic-sedimentary rocks (parmado1 code 1)
CALCSED_pc	Number	Sedimentary rocks (chemical, organogenic or biogenic in origin)
IGNEOUS_pc	Number	Igneous rocks
MORPHIC_pc	Number	Metamorphic rocks
FLUVDEP_pc	Number	Unconoslidated deposits (alluvium, slope deposits); clay, silt, sand
GLACDEP_pc	Number	Unconsolitadet glacial deposits/glacial drift
EOLIAN_pc	Number	Eolian deposits (loess)
ORGANIC_pc	Number	Organic deposits
PAMA_pc_OK	Number	Data quality of parental material in prim. catchm.; 100% - (0+NoData)
FOR25_PC	Number	Percentage of forest in upstream catchment (source: paneuropean forest map)
FOR25_pc_OK	Number	Data quality of forest map in prim. catchm.; 100%-(0+snow/clouds+NoData)
URBAN_pc	Number	Percentage urban area in pc (reclass 1)
AGRI_pc	Number	Percentage agricultural areas in pc (reclass 2)
PAST_pc	Number	Percentage pasture in pc (reclass 3)
FORES_pc	Number	Percentage forest in primary catchment (CORINE, reclass 4)
SCRUB_pc	Number	Percentage scrub and/or herbaceous vegetation in pc (reclass 5)
NOVEG_pc	Number	Percentage open spaces little or no vegetation in pc (reclass 6)
WETL_pc	Number	Percentage inland wetlands/marshes in prim catchm (reclass 7)
MWETL_pc	Number	Maritime wetlands (reclass 8)
EST_pc	Number	Marine waters, coastal lagoons, estuaries (reclass 9)
CLC_pc_OK	Number	Data quality of CORINE land cover in prim. catchm.; 100%-(0+NoData), rest is others
IRGT_PC	Number	Percentage of permanently irrigated agricultural land in pc (CORINE Code 212)
rddens1_pc	Number	Road density of primary routes in pc
rddens2_pc	Number	Road density of secondary routes and local roads in pc
rddens1_du	Number	Road density of primary routes in du
rddens2_du	Number	Road density of secondary routes and local roads in du
ERODI_du	Number	Mean of erodibility in upstream catchment (mean of ERODI_PC)

tblsite_gis (continued)

	PREC_AN_du	Number	Mean annual precitpitation upstream catchment (mean of primary catchments)
	TEMP_AN_du	Number	Mean annual temperature in upstream catchment (mean of primary catchments)
	ELEV_MN_du	Number	Mean elevation of upstream catchment (area-weighted mean of primary catchments)
	clastseddu	Number	Consolidated-clastic-sedimentary rocks (parmado1 code 1)
	calcsed_du	Number	Sedimentary rocks (chemical, organogenic or biogenic in origin)
	igneous_du	Number	Igneous rocks
	morphic_du	Number	Metamorphic rocks
	fluvdep_du	Number	Unconoslidated deposits (alluvium, slope deposits); clay, silt, sand
	glacdep_du	Number	Unconsolitadet glacial deposits/glacial drift
	eolian_du	Number	Eolian deposits (loess)
	organic_du	Number	Organic deposits
	fores25_du	Number	Percentage of forest in primary catchment (source: paneuropean forest map)
	urban_du	Number	Percentage urban area in du
	agri_du	Number	Percentage agricultural areas in du
	past_du	Number	Percentage pasture in du
	fores_du	Number	Percentage forest in drainage area upstream (CORINE, reclass 4)

	Field Name	Data Type	Description
	scrub_du	Number	Percentage scrub and/or herbaceous vegetation in du
	noveg_du	Number	Percentage open spaces little or no vegetation in du (reclass 6)
	wetl_du	Number	Percentage inland wetlands/marshes in du (reclass 7)
	mwetl_du	Number	Percentage maritime wetlands in du (reclass 8)
	est_du	Number	Percentage marine waters, coastal lagoons, estuaries in du (reclass 9)
	irgt_du	Number	Percentage of permanently irrigated agricultural land in du (CORINE Code 212)
	pama_du_OK	Number	Data quality of parental material in catchm.; 100% - (0+NoData)
	clc_du_OK	Number	Data quality of CORINE land cover in catchm.; 100%-(0+NoData), rest is others
	for25_du_OK	Number	Data quality of forest map in catchm.; 100%-(0+snow/clouds+NoData)
	snapdist	Number	Distance site had to be moved to CCM2 river network
	WINDOW	Number	Code of CCM2 data package
	WSO1_ID	Number	ID of the primary catchment drained by the river segment (CCM2)
	HYDROID	Number	ID for hydrological analysis (CCM2)
	datpack	Number	Sorting for input packages

tbltaxa_and_guilds

	Field Name	Data Type	Description
	Taxa_id	AutoNumber	Counter id
🔑	Species_name_EFIplus	Text	Latin species name according to EFI+ project experts
	Species name Kottelat & Freyhof 2007	Text	Latin species name according to Kottelat & Freyhof 2007
	Species_name_EFIplus_and FAME_old	Text	Latin species name of old FAME project
	Comment	Text	Comment
	Fides occure	Number	Fides occure
	EFIplus_abundance_24102007	Number	EFIplus_abundance_24102007
	NUM	Number	NUM