BONUS-MICROPOLL

TASK 1.4

Marine Plastic Data Base (MPDB)

Documentation (v. 1.0)

By Natalja Čerkasova

Klaipeda University, Marine Research Institute

e-mail: [natalja.cerkasova@ku.lt](mailto:natalja.cerkasova@ku.lt)

# Database description and aim

MICROPOLL Database (MPDB) is a database for all data collected within the project. It allows general import/export and data exchange with other international and national Marine Litter databases (e.g. German Baltic Marine Litter data base), and provides tools for data quality insurance and assessment, spatial visualization and statistical analysis.

The database is developed using SQL as a widely spread and well-documented language with the necessary flexibility of storing data. Key points of the task:

* DB allows general export and data exchange (csv, ascii, xml);
* DB allows spatial visualization (with extra tools) and statistical analysis;
* An assigned person with sufficient IT skills (Natalja) will handle DB administrative management (thru the project implementation phase). Other users receive a user key (with limitation, to assure data quality and intactness).
* DB should be marinated following a technical maintenance plan which should include:
  + Data quality assurance;
  + Data backup plan;
  + Risk assessment plan;

# Database structure

The developed database structure is presented in Figure 1. The main tables are:

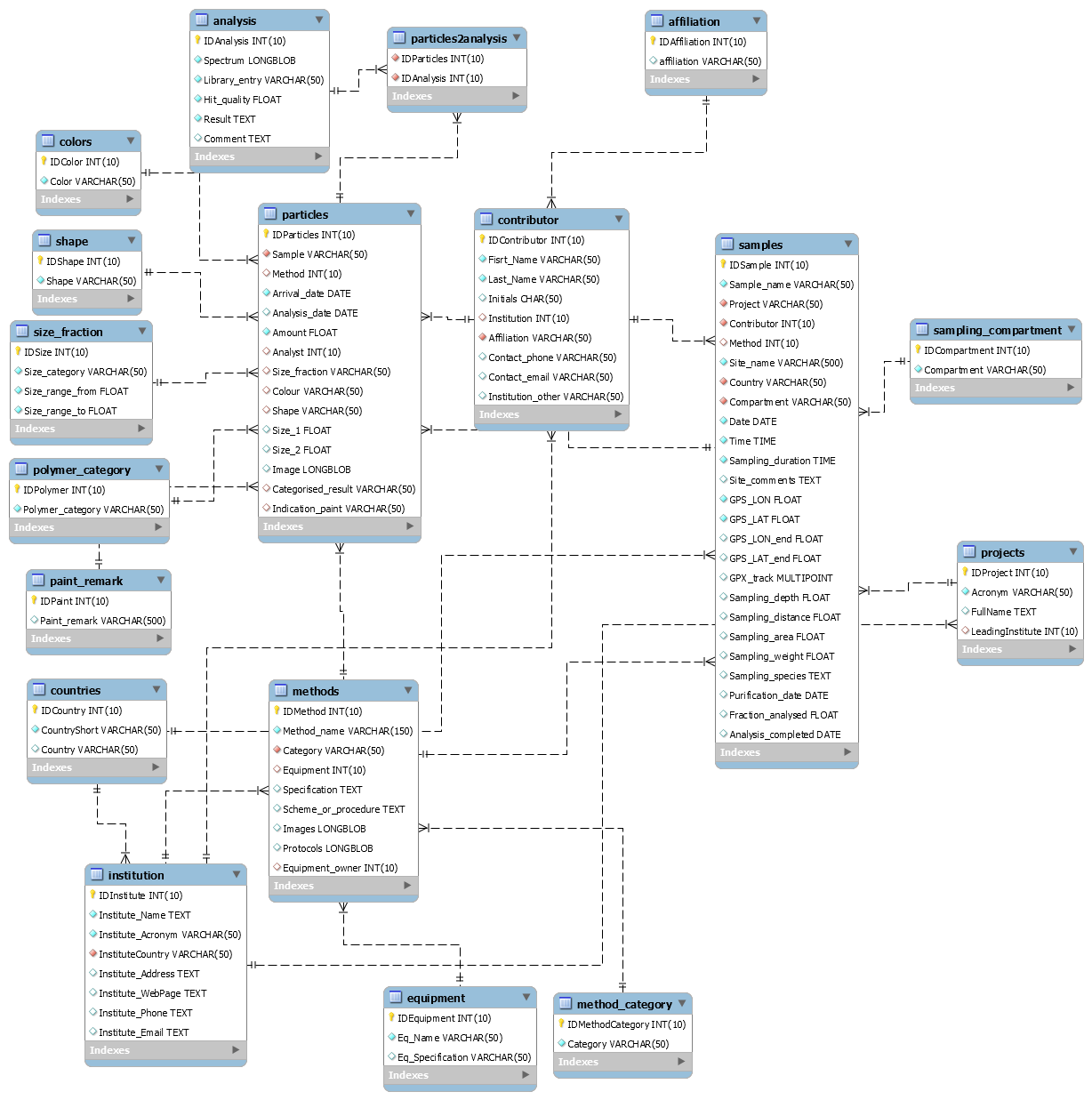
* Samples table;
* Analysis table;
* Particles table.

The supporting tables are:

* Colors;
* Shapes;
* Size fraction;
* Polymer category;
* Paint remark;
* Countries;
* Institution;
* Equipment;
* Method\_category;
* Projects;
* Sampling\_compartment;
* Affiliation.

The many-to-many relationship table is:

* Particles2analysis;



**Figure 1.** MPDB EER diagram

## Supporting tables

Information in the supporting tables has to be filled up prior to the main tables. The information to be filled should follow the logic:

1. User fills out his/hers institution in the “Institution” table, if necessary (to be completed only once);
2. User fills out his/hers credentials in the “Contributor” table, if necessary (to be completed only once);
3. Fills out the “Projects” table, if necessary (to be completed only once);
4. User goes thru all the supporting tables and fills out the information needed (to be completed once or multiple times, depending on the situation);
   * Equipment (which equipment was used);
   * Method (which method was used);

The tables “Color”, “Shape”, “Polymer category”, “Paint remark”, “Size fraction” are already filled up with the agreed-upon categories.

## Main tables

After completing the information input of the supporting tables, the user may proceed to the main tables. The data entry should follow the logic:

* A sample entry should be created in the “Samples” table, which provides the information on the collected sample/samples;
* An analysis method should be created in the “Analysis” table (if not created prior), by which these samples were analyzed;
* The Particles entries should be added to the “Particles” table. The user may specify how many on which category, shape, size, color, etc. particles were found in which sample analyzed by which method.

## Relationship tables

Because the basis of the relation database is the one-to-many data integrity and structure, for many-to-many relationships a separate table should be included, which will define this relationship. The table is called “particles2analysis” and defines the IDs of those particles or those particles, who were found by multiple analysis method. In short, this table provides the linkage between multiple particles found by multiple analysis and contains the unique IDs of said entities.

# Database access

The users may request the MPDB access via e-mail to: [Natalja.cerkasova@gmail.com](mailto:Natalja.cerkasova@gmail.com). Every user is issued with unique credentials for the DB usage.

The DB is currently hosted by the IOW server at 192.124.245.26 (micropoll.io-warnemuende.de).

The instructional videos for MPDB usage are available at google drive storage (link): <https://drive.google.com/drive/folders/1hdwP6AmJ1AAYQ-JbPyR0Sf7XGZGH_I5Z?usp=sharing>

# Future work and fixes

Future work includes:

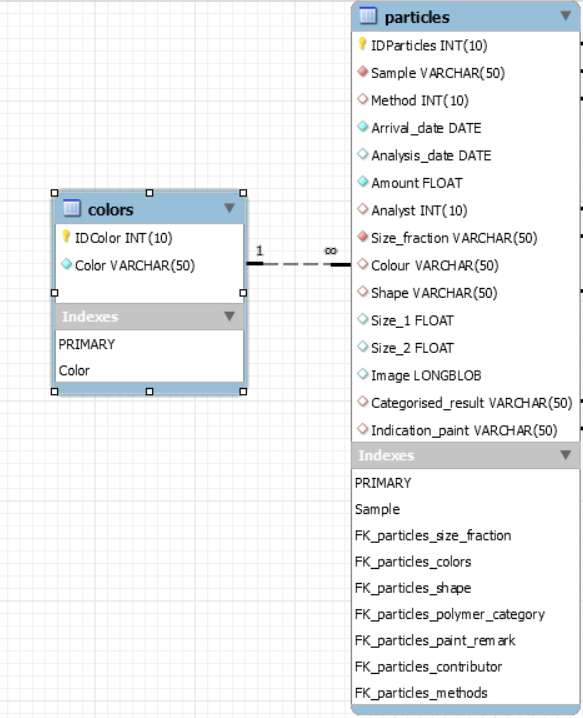
* The assessment of circular references within the database and possible separation of the tables;
* The assessment of fields and their constraints/requirements.

# Appendix 1. In-depth table descriptions

## “**Colors**” table

A supporting table with one unique ID for each color, describing the particle color.

The “colors” table is used as a supporting table for the “Particles” tables and is linked with a “one-to-many” relationship based on the name “Color”.

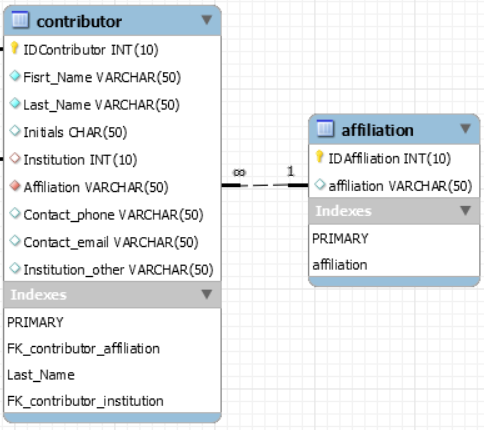


**Figure 2**. Colors table and related tables.

*Note: the relationship was chose not as ID, but as color. The name of the color is by itself unique, so there should not be any difficulties of problems with this relationship*.

## “**Affiliation**” table

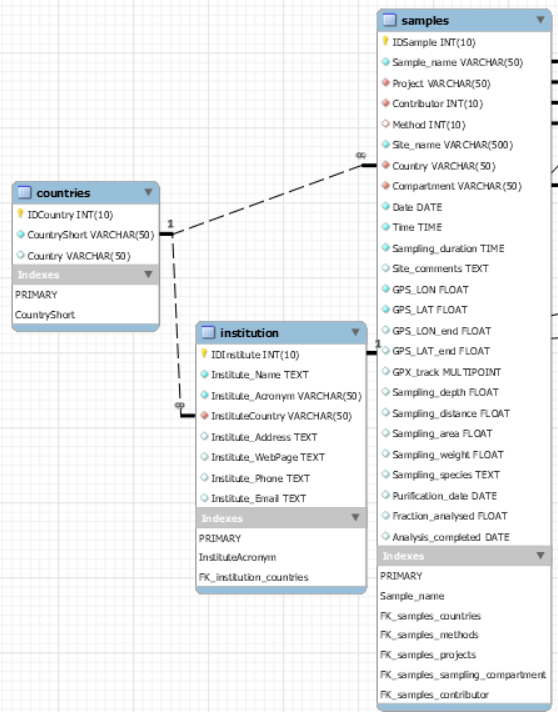
Affiliation table is a supporting table, which provides the user (contributor) with assigned affiliation. Contributor affiliation is not a required field, so the table and corresponding related field is optional.



**Figure 3.** Affiliation table and related tables

## “Countries” table

Countries table is a supporting table that is used in the “Samples’ and “Institution” tables in the field of the same name.

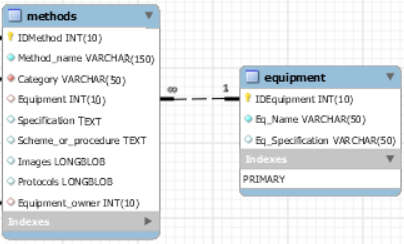


**Figure 4.** Countries table and related tables

The “Countries” table has the long name and a short name field, e.g.: Germany, DE. A key of the Short name is used in the other tables, as the key is always unique and a more user-friendly solution to the user, rather than the entire name or the ID. The relationship is, as with previous tables. “one-to-many”.

## “Equipment” table

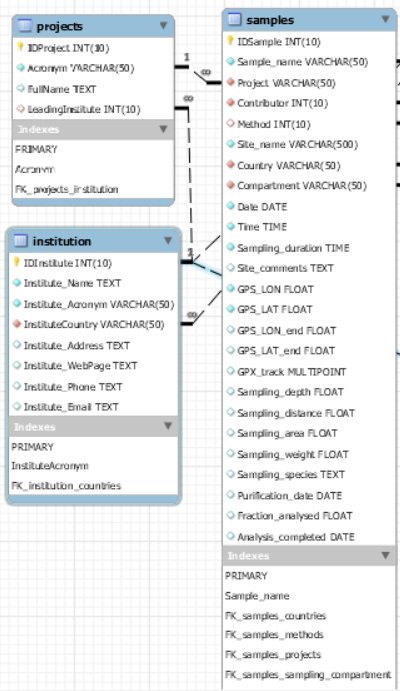
Equipment table is a secondary table, which stores the information about the used equipment. All the used equipment may have non-unique names, so the primary key is the ID of the entry. This table is linked to the “Methods” table, where the user can define an analysis method with the used equipment by its ID.



**Figure 5.** Equipment table and related tables

## “Projects” table

Projects table is a supporting table, which defines the data or sample origin based on project. As this DB may be linked to other projects, it is necessary to make the versatility of indicating to which project the collected data is attributed to. This is handled thru the fields in related tables (Samples). The “projects” table also contains one foreign key: a leading institution key, drawn from the “Institution’ table.

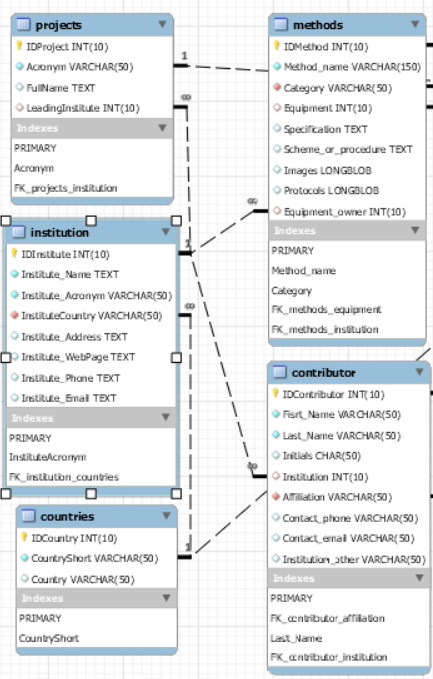


**Figure 6.** Projects table and related tables

The primary key of the project table is the Id, although the association is made via the acronym, which is supposed to be a unique name. This solution was made more for the user benefit, to make the DB more user-friendly.

## “Institution” table

Institution table is a supporting table with one foreign key (countries) and three related tables. The institution holds information on the participating or contributing institutions. It is related to the projects table via the project leading institution field; to the methods table via the equipment owner and to the contributor via the contributors main working place.

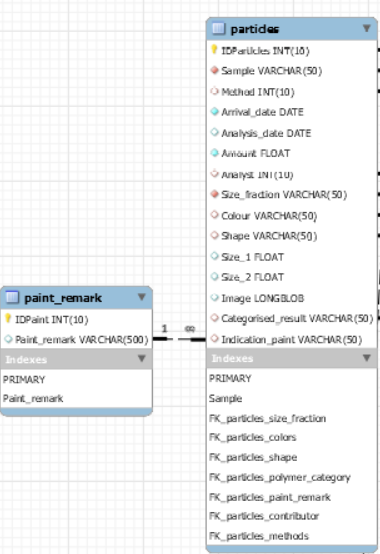


**Figure 7.** Institution table and related tables

The institution is related (or referred to) the other tables via its unique ID.

## “Paint remark” table

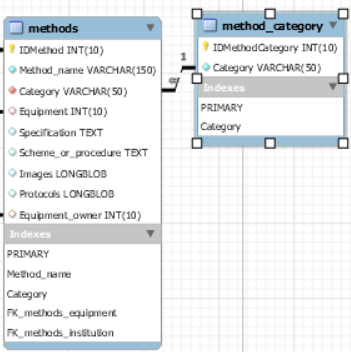
Paint remark table is a supporting table, which is related to the particles table and provides information on the specific paint remark, if a particle was identified as paint. The table is pre-filled and does not require any adjustment from the user. The associated field in the particles table is optional.



**Figure 8.** Paint remark table and related tables

## “Method category” table

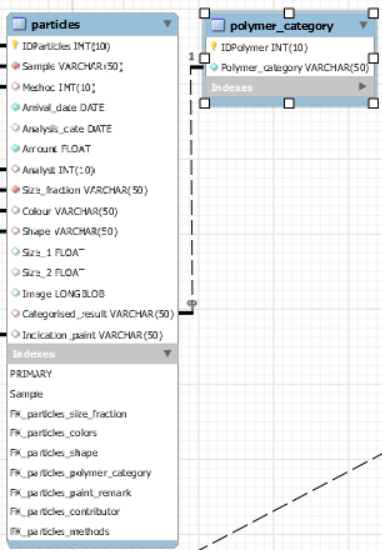
Method category is a support table, which defines multiple categories available. This table is pre-filled, so the user does not require editing this table. Each method in the “methods” table requires a category.



**Figure 9.** Method category table and related tables

## “Polymer category” table

Similar as the “method category”, the “polymer category” table provides information on the classification of the polymer. The related table “particles” is linked to the polymer category via the “Categorized result” field.

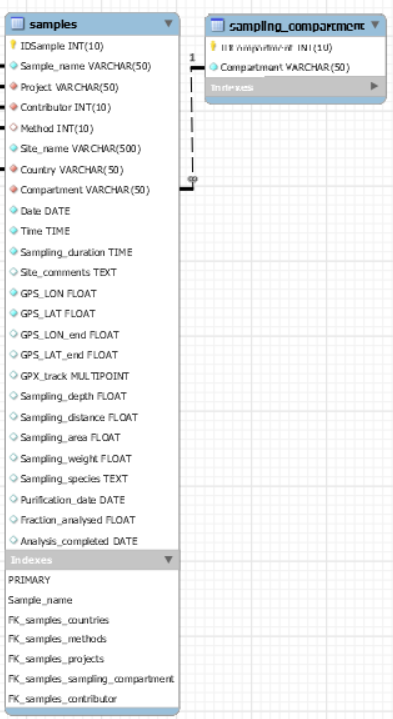


**Figure 10.** Polymer category table and related tables

This table, similar to previous, is pre-filled with agreed categories, so no user input is required. The user must select a particle category by the name (which is unique) for a more convenient table data population.

## “Sampling compartment” table

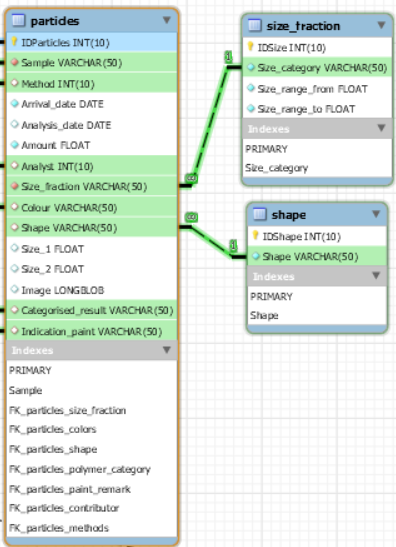
The Sampling compartment table is a supporting pre-filled table which provides a list of sampling compartments (water, beach, etc.). This table is related to the “Samples” table. The user must select the sampling compartment for each sample that is entered in the “Samples” table.



**Figure 11.** Sample compartment table and related tables

## “Size fraction” and “Shape” tables

“Size fraction” and “Shape” tables are supporting tables that store a list of categories attributes for the particles. Size fraction defines a size category for each of the fractions, whereas the “Shape” defines a shape category. Both tables are pre-filled with agreed categories, so the user does not need to enter any data to these tables.

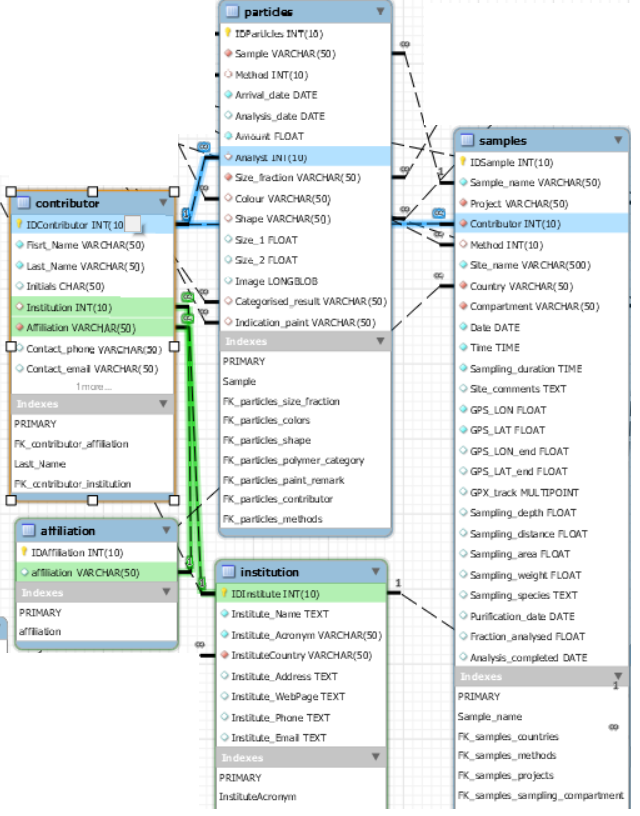


**Figure 12.** Size fraction and shape tables and related tables

Because the data entries are unique for each category, the key fields are the description fields (not the IDs). This, as with previous cases, is a solution for the users benefit.

## “Contributor” table

The “Contributor” table stores information on the people involved in the projects, sampling or analyzing the samples. The contributor contains two foreign keys: Institution and Affiliation, which are self-explanatory.



**Figure 13.** Contributor table and related tables

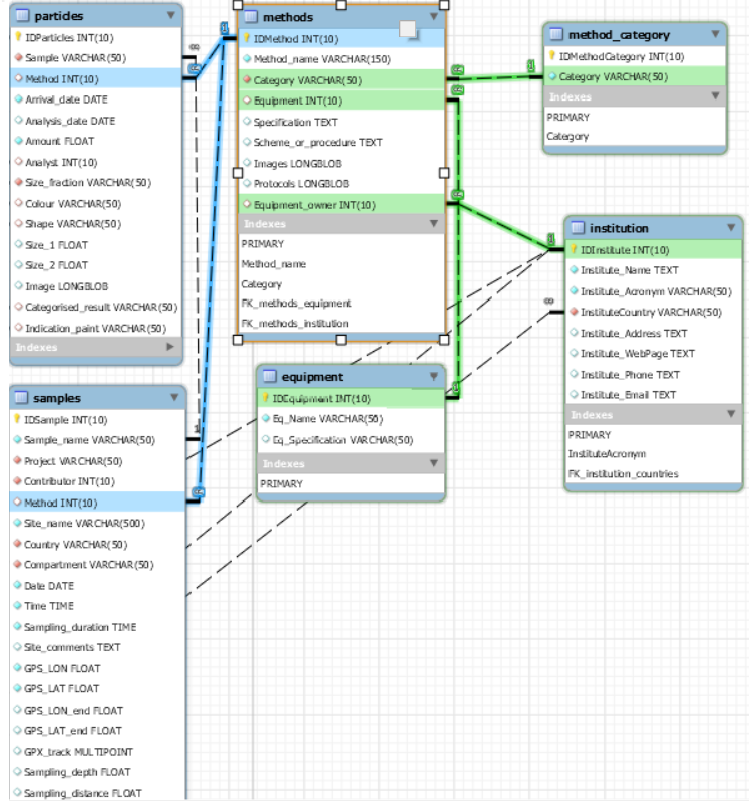
The “Particles” table uses the contributor as the analyst (person who did the analysis and found the particle). The table “Samples” uses the contributor as the person who did the sampling. Contributor is used in these related tables using their unique ID, because names, surnames, initials and so on may duplicate.

The user must always fill in the name and surname of the contributor before proceeding to other tables.

## “Methods” table

The methods table stores the information on various methods, used both in sampling and in the sample analysis.

The table has three foreign keys: Method category, Institution and Equipment. These tables were covered in this documentation and their information is self-explanatory.



**Figure 14.** Methods table and related tables

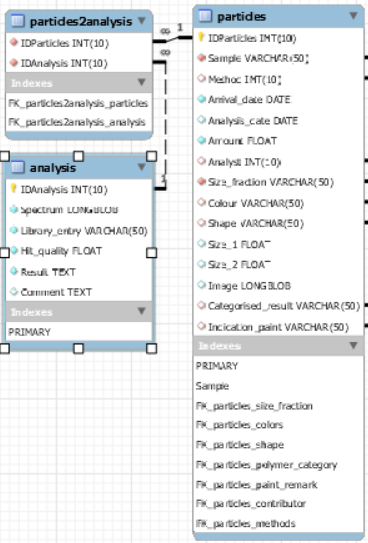
“One-to-many” relation defines the relationship of the methods table to two other tables: Samples and Particles. In the “Samples” table it refers to the sampling method, whereas in the “Particles” table it refers to the analysis method. Both fields are defined (linked) by the method IDs.

Note: this table and the relationship definition creates a circular reference, which is subject to further improvements.

## “Analysis” and “particles2analysis” tables

The “Analysis” table stores information on the particular analysis that has been conducted. This table is logically related to the particles table (Particles were found using some analysis), but as such connection can be defined as “one-to-many” in the relational database, it cannot be fully implemented in cases, where one particle could be detected by multiple analysis.

Thus, a particles2analysis table is the binding table, where information on both, the particles and the analysis, can be stored, and provided the many-to-many relationship possibility.



**Figure 15.** Analysis and particles2analysis tables and related tables

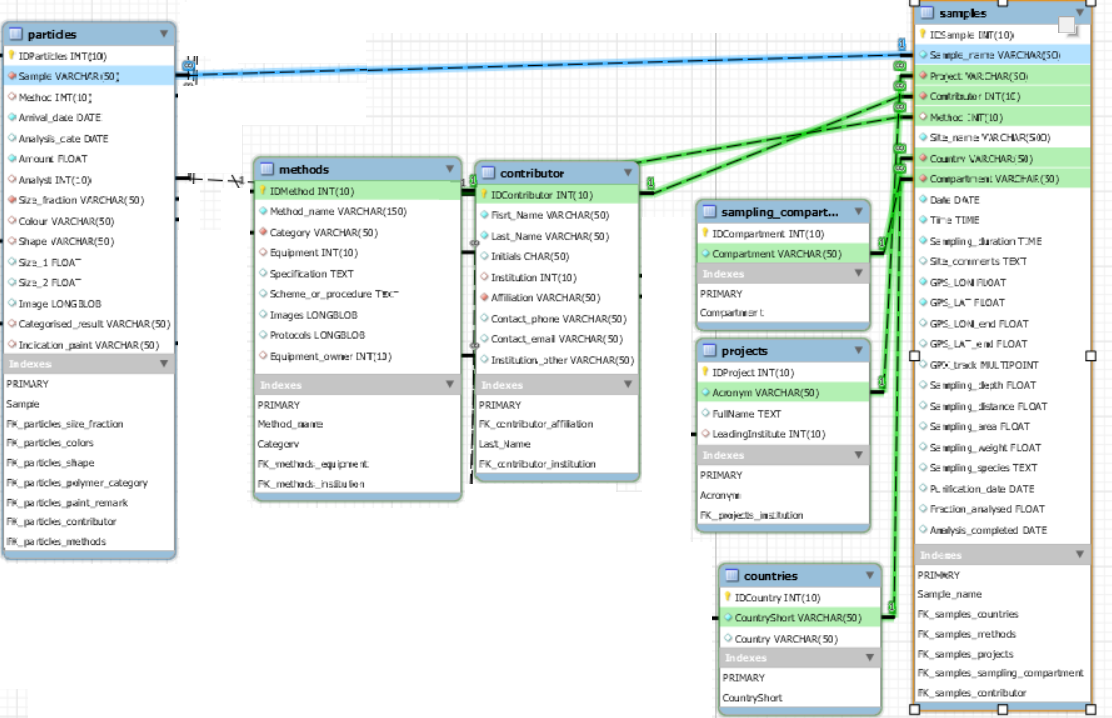
The user has to enter the analysis and the particles separately. Then the user is required to fill in the particles2analysis table, where one entry is a combination of related IDs. This way, a relationship between particles and analysis is made.

Note: this particular relationship might be undistinctive to the user, thus is subject to review and change.

## “Samples” table

The samples table is one of the main tables, which stores information about the collected samples. The table has five foreign keys:

* Contributor (the person who collected the sample). “One-to-many” relationship based on contributors ID;
* Project (the project associated with the sample). “One-to-many” relationship based on projects acronym;
* Method (sampling method). “One-to-many” relationship based on methods ID;
* Country (sampling country). “One-to-many” relationship based on countries short name;
* Sampling compartments (in which compartment was the sample taken). “One-to-many” relationship based on sampling compartment name.



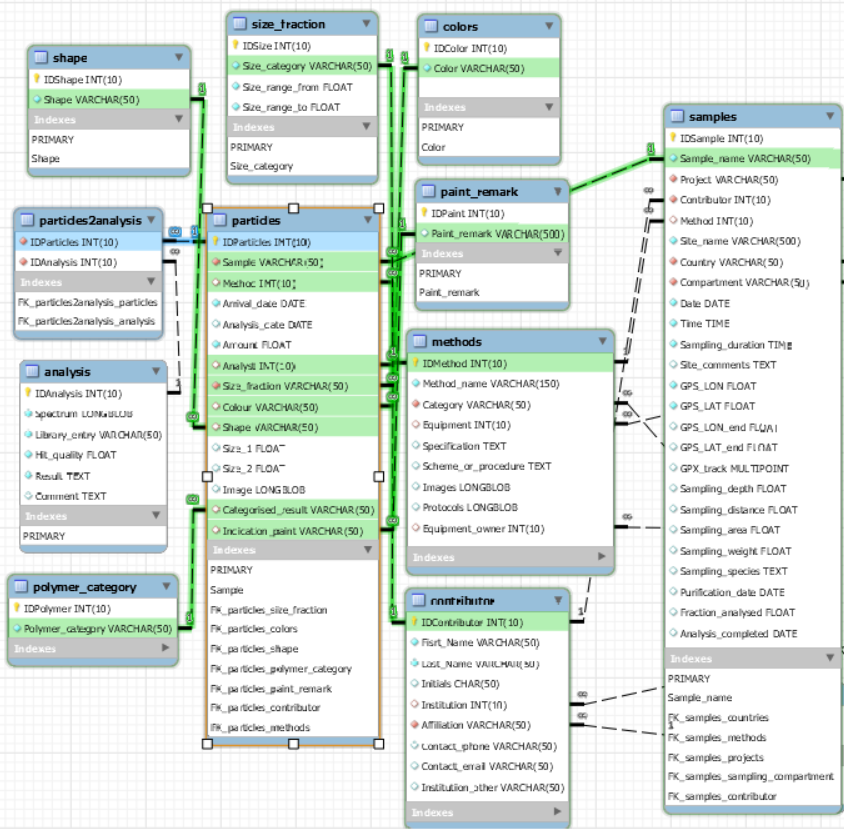
**Figure 16.** Samples table and related tables

This table is related to “Particles” table. The relationship is “one to many”, as the one sample may lead to many particles. On request, this relationship is defined by the field “Sample name”, which should be unique but user-defined.

## “Particles” table

The main table of the database, as it stores the information on the particles, found in samples by conducting an analysis (or many analyses). The particles table has seven foreign keys:

* Shape (the shape of the particle). “One-to-many” relationship based on shape name;
* Colors (the color of the particle). “One-to-many” relationship based on color name;
* Size fraction (categorized size fraction of the particle). “One-to-many” relationship based on size fraction category name;
* Paint remark (optional field). “One-to-many” relationship based on the remark text;
* Method (analysis method). “One-to-many” relationship based on methods ID;
* Contributor (the person, who performed the analysis). “One-to-many” relationship based on contributors ID;
* Samples (sample, which was analyzed and where the particles was located). “One-to-many” relationship based on samples user-defined ID;



**Figure 17.** Particles table and related tables

The related table is particles2analysis, which defines a many-to-many relationship with the performed analysis based on analysis ID.

# Appendix 2. MPDB EER Diagram

