Guideline for usage of Microplastic Database

The database (DB) is used to commonly collect data on microplastics (MP) from the projects BONUS MICROPOLL (BONUS) and MicroCatch_Balt (BMBF). The purpose of this guideline is to ensure that a new user has the required understanding of how to enter, extract and interpret data.

Please carefully read the following sections in order to understand how the database is designed and how it should be used. **It is absolutely critical that the data within the database is entered, extracted and interpreted in a consistent manner.** If not, the extraction of information from the database can yield wrong information that lead to wrong interpretation of any kind of results.

Whenever in doubt, contact the following people:

Natalja Čerkasova	KU	DB creation and maintenance	natalja.cerkasova@apc.ku.lt
Kristina Enders	IOW	Sampling, processing	kristina.enders@io-warnemuende.de
Robin Lenz	IOW	Sampling, processing	robin.lenz@io-warnemuende.de
Josef Brandt	IPF	Analysis	brandt@ipfdd.de
Franziska Fischer	IPF	Analysis	fischer-franziska@ipfdd.de

Table of Contents

Guideline for usage of Microplastic Database 1
DB structure 2
Main tables 2
Auxiliary tables 3
Linking tables 3
Data entry 4
Upload of a sample 4
Upload of a particle 9
Data retrieval 11
Extraction from a single table (example) 11
Extraction from multiple tables (example) 12
Data interpretation 13
Additional information 15

DB structure

The DB consists of tables where data is entered (Figure 1). Each table has got defined columns, which tell what data can be entered (e.g. “size”), if applicable what unit is expected (part of the name, e.g. “µm”) and which data type the entry requires (e.g. “float”). An entry in a table means a line filled with coherent information about one object.

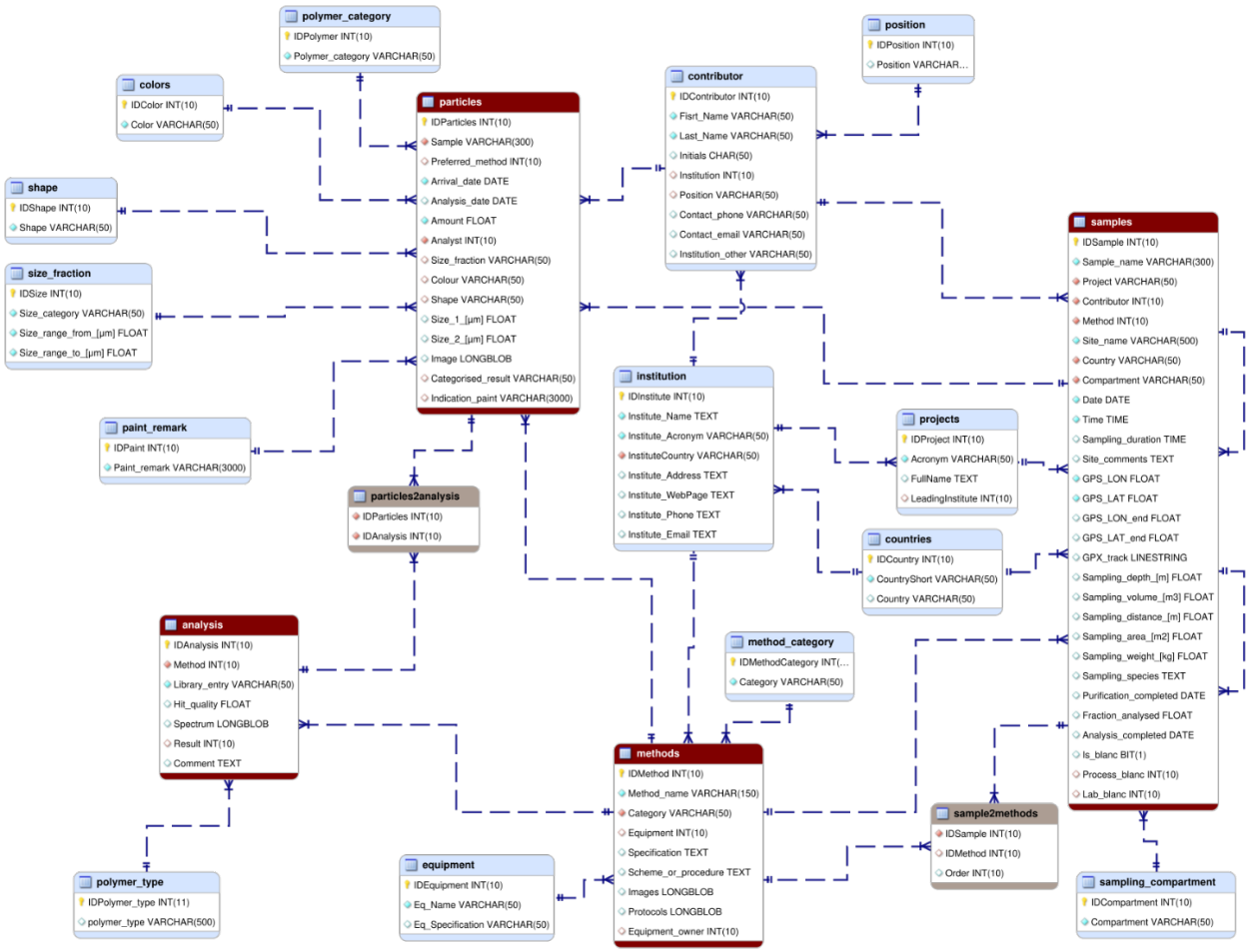

Figure 1: MP-database structure

For simplicity, the tables can be grouped into 3 categories: main, auxiliary and linking tables.

Main tables

→ contain the actual data about MP a user typically would like to do research on. They inherit a lot of attributes from the auxiliary tables (see below).

- **Samples**
  - each entry describes properties of one sample (e.g. sediment taken at a specific time and location)
  - for information on the “Upload of a sample” (page 4) and specifications of this table see the respective section

- **Methods**

- every method, from sampling, processing and pre-analysis treatment to final particle identification is listed here
- this avoids inflating the tables “samples”, “particles” and “analysis” with repetitive entries of methodological details
- all relevant information has to be listed (e.g. mesh sizes, laser wave lengths, chemicals used, etc.)
- files (sketches, photographs and detailed protocols) can be included

- **Particles**
- an entry in this table stands for one analysed suspected plastic item
- if a sample was analysed and no plastic was found the *Particles* table should receive only one entry containing the link to the sample ID, the arrival and analysis date (and no actual results)
- the column “categorised_result” contains a decision for what polymer type and group the particle is made of based on the analytical results

- **Analysis**
- each entry details the specific analytical results of a successful attempt to identify the polymer type of a particle

Auxiliary tables

→ Many of them are simple lists that provide predefined entries for certain columns of the main tables

- **“shape”**
- **“colour”**
- **“polymer_category”**
- **…**

→ Some of these tables also provide additional information on the entered object to better describe it

- **“contributors”** is a list of people that is used to link entries from “methods” or “samples” to the responsible person. Besides the name it provides various contact information.
- **“projects”** lists projects (currently just BONUS MICROPOLL and MicroCatch_Balt), along with their leading institutes
- **“institutes”** provides names, locations and contact information of involved institutes
- **...**

Linking tables

→ are used in places where it is necessary that it is possible that several entries in one table can be linked to several entries of another table

- **“samples2methods”**: any method can used by many samples but also every sample is treated by several methods (e.g. sampling method and purification method)
- **“particles2analysis”**: every particle can have several analytical results (e.g. multiple measurements or complementary analyses on different instruments)

Data entry

Data entry tools

Entries to the DB can be made manually or, as in most cases, for multiple entries an entire data set can be uploaded to the DB. One option to prepare a data set for a group of samples is to set up a spreadsheet (Excel or Calc) and save it to comma separated file (.csv).

It is advisable to download the current version of the target table in the database as a csv file. This file can then be used to insert the new data entries in order to ensure the correct formatting and column structure. The csv containing the new entries can then be merged into the DB table by use of a MySQL management software (e.g. free tools, like the HeidiSQL, MySQL Workbench; or commercial software like Navicat for MySQL, etc.) or available MySQL interface for programming languages (e.g. R, Python). An example of how to include new data entries via csv import in HeidiSQL is given at the end of this section.

Another way is to open the current version of a DB table using the Microsoft Excel addon "MySQL for Excel" where entries can be added manually or by pasting new content and committing the changes when done. An explanation video can be found at: https://drive.google.com/open?id=1hdwP6AmJ1AAYQ-JbPyR0Sf7XGZGH_I5Z

Data entry tables and fields

Several fields are mandatory for an entry to be accepted by the DB, others do only apply to certain sample or particle types and are therefore optional. Generally, this information can be found in the DB table information in a column typically named "Nullable" or "Allow Null" (NO – mandatory, YES – optional) and is also described in Figure 1.

You do not have access to all tables. Especially ***auxilliary tables***, e.g. holding data about possible colours or shapes, are fixed. If you wish to have more entries there, please consider the absolute need for such additions (discuss this in a group with your colleagues and consider how much this will benefit your data representation accuracy) and contact Natalja to add them. The current entries in the ***auxilliary tables*** have seen much scrutiny, so we feel that some additions are not required.

Data backup/ Accidental manipulation of data

Generally, try to avoid changing data entered by others. However, if you accidentally manipulated or deleted data or notice any other kind of mistake, please contact the responsible person and the DB creator (Natalja) for clarification. A backup of the DB is made every week, in accordance with the data management plan.

Upload of a sample

A sample contains all information regarding the sampling and pre-treatment procedure and is required to add analysed MP particles. Each particle is assigned to a sample, so without having the sample in the DB, no particles can be entered. Make sure sample data is entered at the latest when samples are transferred to the analyst.

Please agree with all people involved on **suitable and consistent naming conventions** and clarify WHO is supposed to upload a sample. There were cases, in which samples were added twice – once in regular

spelling, once in all lowerCase letters. Such duplicates are very hard to identify and can lead to serious errors. **Make sure each sample is present in the DB exactly once!**

The database is structured to distinguish blank samples from environmental samples. **Blanks are critical to assess the reliability of any final analysis.** We consider two different types of blank samples: Process_blanks and Lab_blanks. Process_blanks are samples of MP-free milliQ water that were treated identically to an actual environment sample (including workup-procedures). They indicate any contaminations coming from sample workup. Lab_blanks, on the other hand, are MP-free milliQ-samples that were created in the analysis lab and can indicate contaminations coming from the analysis lab environment.

The database stores ALL FOUND MP PARTICLES. So, to get a particle report of a certain environmental samples, one has to retrieve all particles of the sample of interest, but also check the corresponding blanks. Ideally, the blanks only contain negligible amounts of MP particles. If, however, in rare cases there was a contamination, the blanks will indicate these and show, for instance, 200 PE particles. Consequently, any PE content of the corresponding environmental sample is highly questionable and should probably be discarded.

To add a sample, you have to add an entry within the samples-table in the DB. For the samples table the following information is required for each entry (mandatory <> optional):

mandatory attributes:

- **IDSample:** ID automatically generated by the DB
- **Sample_name:** What name describes the sample best? → uniqueness and consistency!
- **Project:** To which project does the sample belong (project financing the campaign)? → choose ID of linked entry in the ***projects*** table!
- **Contributor:** Who is responsible for the sample? This is the person who sampled and/or processed the sample and is thus in charge of the quality of the data entry. This person has to agree on the usage of the data and results by other DB users. → choose ID of linked entry in the ***contributors*** table!
- **Site_name:** Where was the sample acquired? → If you got replicate samples, the site_name is the same for all. The replicates should be distinguishable by the sample_name
- **Country:** In which country the sample was taken? → choose abbreviation of linked entry from the ***countries*** table!
- **Compartment:** Which environmental compartment / matrix is sampled (e.g., biota, beach, soil)? → choose ID of linked entry of the ***sampling_compartments*** table!
- **Date:** What date the sample was taken?
- **Time:** What time the sample was taken? → Accuracy limit: min. 1 hour, if possible
- **GPS_LON:** geodetic longitudinal coordinates in decimal degrees → Accuracy limit: 4 decimal places, positive numbers are east, negative numbers are west

- **GPS_LAT:** geodetic latitudinal coordinates in decimal degrees → Accuracy limit: 4 decimal places, positive numbers are north, negative numbers are south

optional attributes:

- **Sampling_duration:** How long did the sampling take? → Required for samples listed under the compartment "water", possibly "waste water" and "biota"

- **Sample_comments:** Are there any additional information that could influence during sampling that could influence the result? Alternatively, the field can also be used to inform the analyst on the existence of sample splits, by the analyst to inform about additional information retrieved during analyses (e.g. internal standard particles) and other information important for the specific small set.

- **GPS_LON_end:** In case of transects being sampled: What is the longitudinal coordinate endpoint in decimal degrees → Accuracy limit: 4 decimal places, positive numbers are east, negative numbers are west, required for e.g. Manta net sampling

- **GPS_LAT_end:** In case of transects being sampled: What is the latitudinal coordinate endpoint in decimal degrees → Accuracy limit: 4 decimal places, positive numbers are north, negative numbers are south, required for e.g. Manta net sampling

- **Sampling_depth_[m]:** In which depth layer the sample was taken? → In case of water, biota and sediment samples = water depth; soil and beach samples: depth from surface, if not surface

- **Sampling volume_[m3]:** What is the volume of the matrix sampled? → Required for samples listed under the compartment "water", possibly "waste water" and "biota"

- **Sampling_weight_[kg]: What is the weight of your sample?** → only required for samples listed under the compartment "biota", "sediment", "soil", possibly "beach". Apart from "biota" it always refers to the dry weight

- **Sampling_species: Which organism (biota) has been sampled?** → only required for samples listed under the compartment "biota"

- **Purification_completed:** What date the purification/pre-treatment is completed, and the sample is ready to be transferred to the analyst?

- **Fraction_analysed:** If only a certain percentage of the sample was analysed (e.g., due to too high particle numbers), indicate the fraction. Example: 50 % sample translates to fraction of 0.5.

- **Analysis_completed:** Date of the finished analysis, i.e., date of upload to database

- **Is_blank:** Enter a 0 (=False), if the sample is an environmental sample. Enter a 1 (=True), if the sample is a blank sample.

- **Process_blank:** If the sample is an environmental sample, indicate the sample index of the corresponding process_blank sample (if existing).

- **Lab_blank:** If the sample is an environmental sample or a process blank, indicate the sample index of the corresponding lab_blank sample (if existing).

Connect sample to all relevant methods:

To understandably track the history of each sample, please use the **sample2methods** table to explain what methods were applied in what order.

Example: Your sample is already entered and has the sample-index 125. The sample was sampled with the "Rocket sampling" method, underwent a "Density Separation" and was digested with H2O2. Go to the **methods** table and find out the indices of the used methods. Then go to the **samples2methods** table and add the following entries:

IDSample	IDMethod	Order
125	8	1
125	12	2
125	13	3

Final upload procedure on example of HeidiSQL

To download the current samples table in HeidiSQL open the table (just click on it) and activate the data pane. Then go to "Tools" > "Export grid rows". A dialogue window pops up. Chose a path where to save the file and CSV as output format. Click ok.

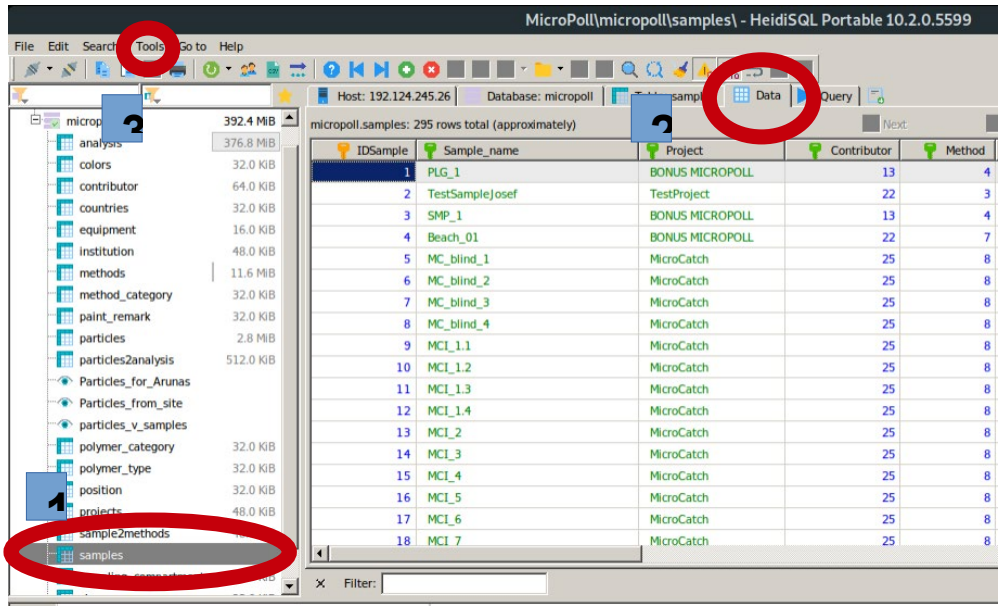

Figure 2: HeidiSQL user interface

You can then go to the downloaded file and open it for example in MS Excel or LibreOffice Calc. Replace the existing data entries with your new entries but don't change the column structure. For empty fields enter NULL, which will indicate for the DB that no entry has to be made there (beware: this is not possible for mandatory fields).

It is advisable to delete the column "IDSample" to have it set automatically by HeidiSQL when the data is uploaded.

When the new data has been formatted and inserted + saved into the previously downloaded csv snapshot of the samples table, the new csv file is ready to be uploaded.

Open HeidiSQL and log on to your account, got to **Tools > Import CSV file**…

A dialogue will open where you specify the path to your csv file. Set all options as shown in the screenshot below. Remove the check mark in front of “IDSample” if you have deleted this column as advised. Then click on import.

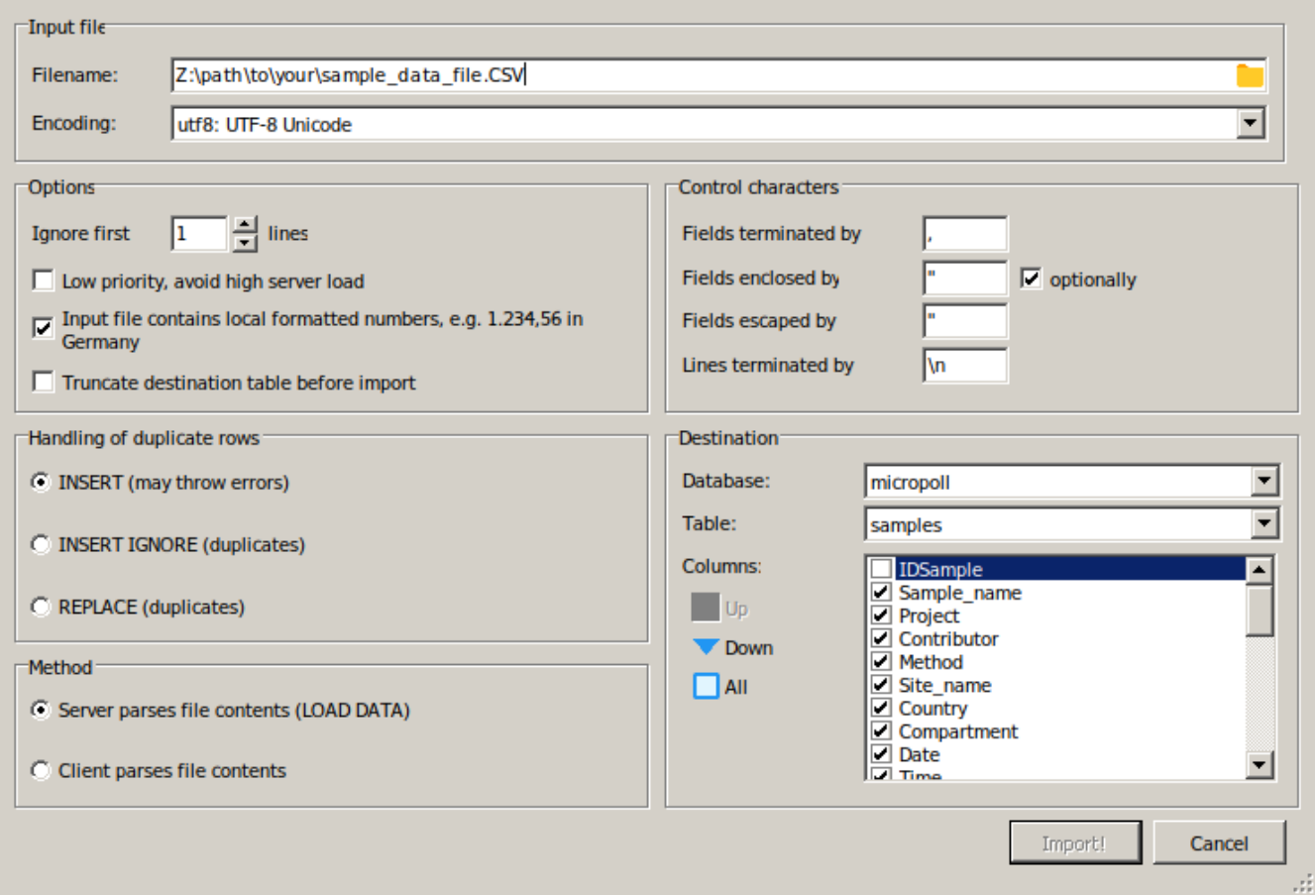

Figure 3: HeidiSQL Data import user interface

Go back to the data view of the samples table and press F5 to refresh. If the import worked your data entries from the CSV file will now be visible at the bottom of the table and should have received automatically generated entries in the “IDSample” field.

Upload of a particle

Uploading a particle is closely related to uploading an analysis. The concept is, that the **particle table** holds any "coarse" information, such as the size and a categorized-result (polyolefin, polyester, epoxy resin). This info is sufficient for any modelling purposes, but some analysists would like to have more detailed info about the exact measurement conditions and results.

At this point, the **analysis table** comes into play. It holds exact details of an analysis (i.e., exact assignment, method used, spectrum). The following figure illustrates the concept. Thereby, each particle can be linked to any number of individual analyses. The link is realized by using the **particles2analysis table**.

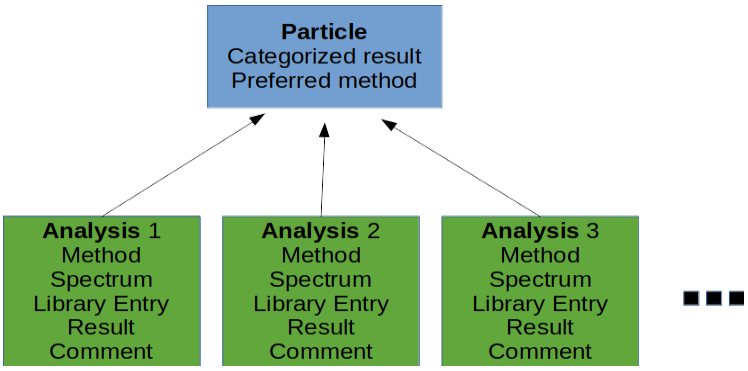

Figure 4: Particles and analysis data structure diagram

Let's go step-by-step and start with the **particle table**:

- **IDParticles:** Id of the new particle, will be automatically generated by the DB
- **Sample:** Name of the sample the particle belongs to.
- **Preferred_method:** Index of the analysis method that yields the final result. If only one analysis was done, its method is automatically the preferred one. → Chose index from **methods table.**
- **Arrival_date:** When did the particle arrive for analysis?
- **Analysis_date:** When was the analysis finished (i.e., uploaded to the DB)?
- **Amount:** If you have two particles that are identical **in every aspect**, you just need to upload it once and indicate the number of identical particles here.
- **Analyst:** Who did the analysis? → choose Index from **contributor table**.
- **Size-fraction:** To what size-fraction does the particle belong? → choose category text from **size_fraction table**.
- **Shape:** How would you describe the particle's shape? → Choose category text from **shape table**.
- **Size_1_[µm]** and **Size_2_[µm]:** Long and short size of the particle, respectively. Given in µm.
- **Image:** If you have an image of the particle, you can upload it here.

- **Categorised_result:** Choose text from **polymer_category table**. Two entries there are worth explaining: “Unknown” means that you can see/feel that it is a plastic particle, but don’t have the possibity to analyze it more in detail. “Uncertain” means that you have analyses at hand with ambiguous results.

- **Indication_paint:** Used to indicate, whether the particle is likely to originate form paint applications. → Choose text from **paint_remark table**.

Let’s come to the **analysis table**. Please create for each particle *at least* one analysis entry.

- **IDAnalysis:** Unique id, will be created automatically by the DB.
- **Method:** Method index of the used analysis technique → Choose from **method table**.
- **Library_entry:** free text of whatever your spectra library reported
- **Hit_quality (optional):** Hit quality of the database search
- **Spectrum (optional):** Upload the corresponding spectrum
- **Result (optional):** Index of the polymer result → Choose from **polymer_type table**
- **Comment (optional):** Free text of anything you want us to know about the analysis.

Finally, you have to define what analysis/analyses belong to what particle. This is done analogously to the methods-to-sample-assignment, but now we use the **particles2analysis** table.

Example: You entered a particle, which got assigned the id 324. Then you entered two analyses, which received the ids 640 and 641. Go to the **particle2analysis** table and enter:

IDParticles	IDAnalysis
324	640
324	641

Data retrieval

When extracting data from the DB the responsible person shown in the field **"contributor"** should be contacted **to verify the results and to give permission** on its usage. This person will be able to forward you to others who were involved in the result generation and might be of help.

In the following two examples are described on how to extract data from the DB yourself. If you encounter problems, every partner (for internal usage) can make a request to the DB creator to provide certain views/scenarios which will contain the requested type of data.

Extraction from a single table (example)

Scenario: I need every sample, from the MicroCatch project from compartment "water" from location between 12.17 – 12.2 and 53.95 – 55.0 (Lat, long) in year 2018.

Explanation: when extracting data from single table you only need to "SELECT" the data. After the "SELECT" based on certain criteria, the user may specify where to store the data and in what format.

Base syntax: SELECT * FROM tableName WHERE condition;

Example:

```
SELECT * FROM `samples`
        WHERE
        samples.Project = 'MicroCatch' AND
        samples.Compartment = 'water' AND
        year(samples.Date) = 2018 AND
        samples.GPS_LON >= 12.17 AND samples.GPS_LON <= 12.2 AND
        samples.GPS_LAT >= 53.95 AND samples.GPS_LAT <= 55.0;
```

Note: every query ends with a semicolon (;)
Export everything to csv: either by using the export tool in HeidiSQL (tools -> Export grid rows); or by executing a statement (for this, delete the last semicolon and update the query as follows):

```
INTO OUTFILE 'D:\full_path\samples_table.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Note: keep in mind that different separators are used to indicate a table (`) and an entry (').

Extraction from multiple tables (example)

Scenario: I need every particle, from location between 12.17 – 12.2 and 53.95 – 55.0 (Lat, long) in year 2018, with a size fraction of 11-100 micro-meters.

Explanation: when extracting data from multiple tables you need to create a new joint table, which is called a View, where all the related data entries are linked. After this, the new table can be filtered as in example 1.

Step 1. CREATE VIEW NewName AS SELECT ColumnName FROM tableName INNER JOIN tables

Example:

```
CREATE VIEW `particles_v_samples` AS
SELECT p.`Sample`,s.`Date`, s.`GPS_LON`, s.`GPS_LAT`,p.`Size_fraction`
        FROM `particles` AS p
```

```
                INNER JOIN `samples` AS s
                        ON p.`Sample` = s.`Sample_name`;
```

Step 2. Filter the new view table as you see fit (like in example 1)

```
SELECT * FROM micropoll.particles_v_samples
        WHERE
        year(particles_v_samples.year) = 2018 AND
        particles_v_samples.GPS_LON >= 12.17 AND particles_v_samples.GPS_LON <=
24.2 AND
        particles_v_samples.GPS_LAT >= 53.95 AND particles_v_samples.GPS_LAT <=
59.0 AND
        particles_v_samples.Size_fraction = '11 - 100';
```

Step 3. Follow the instructions on data export, described previously in this document.

Scenario II: All particles, type of polymer, size, all locations in the compartment water. I also need to know the number of plastic particles per sample and sample volume.

Example

Step1:

```
CREATE VIEW `particles_v_samples_volume` AS
SELECT p.`Sample`, p.`Amount`, p.`Categorised_result`, p.`Size_fraction`,
s.`GPS_LON`, s.`GPS_LAT`, s.`Compartment`, s.`Sampling_volume_[m3]`, s.`Sam-
pling_weight_[kg]`
         FROM `particles` AS p
                INNER JOIN `samples` AS s
                        ON p.`Sample` = s.`Sample_name`;
```

Step2: Filter the new view table as you see fit (like in example 1)

```
SELECT * FROM particles_v_samples_volume
        WHERE
        particles_v_samples_volume.Compartment = `water`;
```

Step 3. Follow the instructions on data export, described previously in this document.

Data interpretation

Handling of blanks:

As already mentioned in the section about sample-upload, the handling of blanks is critical for ensuring validity and reliability of the entered data.

We agreed on uploading ALL MP-PARTICLES to the DB that we find in the respective samples. So, any particles that are most likely contaminations by any of the used methods, will be present in the DB as well. Consequently, when retrieving particle data of a particular sample you have to retrieve the particle data of the corresponding blanks as well and check for possible MP-contaminations.

The reason behind our considerations is, that the decision whether to include any particles that might be due to contaminations, is a very delicate one.

Example:

Your sample of interest has 40 PE particles, the corresponding blank shows:

(a) 0 PE particles. → Great, no questions here!

(b) 30-50 PE particles. → You most likely consider neglecting the PE particles.

(c) 5-10 PE particles. → You probably consider subtracting the number of PE particles. Or you keep the number? Or you neglect all?

As you can see, case (c) can lead to different opinions in how to treat the conflict. Unfortunately, we don't have any reliable data at the moment to tell if case (c) occurs rarely or frequently. Hence, we decided on including *all data* in the first place to be able to decide on a more profound database later on.

Please communicate with persons connected to a certain sample if you have any doubts in how to handle the information from the corresponding blank samples.

There exists two kind of blanks (or also called blinds)

- analytical laboratory blanks
  - there exists one blank per sample analysed with µRaman and / or µFTIR
  - it represents potential contamination introduced during the filtration of the purified sample onto the spectroscopic substrate and during spectroscopic measurements
  - can be seen as a control where detected MP particle counts should be close to zero
- process blanks
  - one or several process blanks show the expectable contamination for a larger group of samples (typically all samples processed in that way)
  - represent sampling and purification steps of a sample as close as possible without containing actual sample material
  - moderate MP particle counts are expected here and define the lower detection limits

Recommendation for blank data implementation – Concept of MP phenotypes

After extraction of a data set (e.g. all MP particles found in one sample) a list of frequencies of each MP phenotype should be made.

Definition: MP phenotype is used here as a unique combination of the entries of polymer type (depending on your needs either categorised results or specific polymer match)

Table 1: Example of an MP phenotype frequency list

Phenotype			Frequency
Green	Polyamide	Fibre	19
White	ABS	Flake	44
Blue	Polyurethane	Foam	62
etc.	etc.	etc.	...

In the same way MP phenotype frequencies are determined for the corresponding lab and process blanks.

To yield MP data corrected for the respective detection limits, the frequencies of each phenotype of the process blanks are subtracted from those of the sample. Furthermore, we recommend to remove any particle counts with phenotype frequencies < 2, to avoid inclusion of potential measurement errors (a MP rarely comes alone). If there is significant contamination present in the corresponding lab blank a subtraction may be done in the same way as for the process blanks.

Finally it is important to report the corrected number of particles found in a sample as a concentration, e.g. MP counts per unit sample. This could be per volume, weight, area, individual, or other depending on the kind of sampling compartment (you can find the needed information in the sample table in the respective columns, e.g. Sampling_volume, Sampling_weight, etc). Also verify in the database whether the whole sample has been processed and analysed or if subsampling or partial analysis has been applied and correct the numbers accordingly. (take a look at the “fraction_analysed” field of the samples table).

Two notes on sizes

1. In addition splitting of the total sample, there may be cases where the samples was size fractioned on spectroscopic filters of different pore sizes (e.g. a 10 µm filter and a 50 µm filter) and only the larger fraction was analysed due to time restraints. In this case, the listed particles may still include MP < 50 µm, which however should be omitted, as is it only by chance whether an individual particle came to rest on the 50 µm filter or went through to the 10 µm filter. It is, however still important that those particles are listed in the DB, in case the 10 µm fraction gets analysed later to have a more comprehensive dataset.

2. The database is intended to be used for particle based MP analysis. Reported data (i.e. numerical MP concentrations) are nearly useless without a detailed analysis of sizes. For samples analysed by microspectroscopy two size dimensions are reported (shorted and longer axes of a fitted ellipse around the pixels of the particle). Only reporting the size range of MP your study focusses on is not sufficient for a scientific evaluation of the pollution problem. As a minimum a size spectrum

histogram should be presented with the numerical concentrations. An approximation of volume (from sizes values) or mass concentrations (from sizes and polymer density) can be calculated.

You may contact the responsible person of a sample to verify your data analysis or if in doubt about anything concerning the sample.

Additional information

Use of the database for internals: When using other project contributors data – please get in-touch with the responsible person and confirm that he/she allows you to use the data. Always acknowledge the persons behind the work and respect your collegues!

Use of the database for externals: The meta-data is visible for other DB that are connected with our BONUS MICROPOLL DB. The actual data can be requested by them. This request must be answered/allowed by the responsible person that is the author of the respective data.