LabData

A Database and
Laboratory Information Management System
with some Special Features for Isotope Laboratories

Tutorial

Axel Suckow

2010-November-15

Isotope Hydrology Laboratory
International Atomic Energy Agency
Wagramer Strasse 5, A-1400 Vienna, Austria

IAEA
International Atomic Energy Agency
Atoms For Peace

and

Leibniz Institute for Applied Geophysics (LIAG)
S3: Geochronology and Isotope Hydrology
Stilleweg 2, 30655 Hannover, Germany
Axel.Suckow@liag-hannover.de

LIAG Leibniz Institute for Applied Geophysics

It is impossible to make anything foolproof because fools are so ingenious!

Build a system that even a fool can use, and only fools will use it!

A fool with a tool is still a fool, no matter how good the tool!

General Introduction

There are several facts about databases. Although widely known, I would like to emphasise some of them here:

- Every scientist today needs databases. Even if this is not directly obvious for the own work (a pencil and a sheet of paper works well for many scientists on the time scale of months or years), it is surely true with respect to the data of somebody else to be used in the own scientific work. It might be just an online literature search…
- Nobody likes to feed data into a database, although everybody likes to have them already in and get them out in a comfortable manner.
- Nearly nobody – with only a few exceptions like me – thinks it might be fun to program a database system.
- Nobody likes to work her/his way through a complicated, sophisticated database system that she/he has not designed on her/his own. I am no exception to this fact.
- If anybody has programmed an own database system, sophisticated or not, this person will surely state that it is the most suitable available for any purpose in discussion. Also here, I am no exception.
- But in some cases, the most important reason to create an own database system and to design it in a way that it finally looks complicated to anybody else is that nobody else can use it and that therefore nobody else can publish the own data in her/his publications. And yes, I want to be an exception to this fact. That's what this tutorial is all about.

But beside these statements there are several reasons, why a database system is necessary in a modern laboratory. Many of them were stated in the LIMS manual (http://water.usgs.gov/software/lims.html) by Tyler Coplen and in my earlier publication on the LabData table topology (Suckow & Dumke 2001). Here I do not want to state again these reasons, but I want to give an overview introduction how LabData does its job. I do this in several steps: a first step-by-step guide called the "LabData Tutorial" explains what the very beginner of LabData must do, to set up the system and to get the first data in and out. This is the document in front of you right now. In a second step the details and secrets (well many details and a part of the secrets) will be described. Probably this part is a never-ending story, since LabData evolves as every piece of software evolves that is in actual use by the one who developed it. This second step is the "LabData Manual", and believe me, this has much more pages. Besides these two, there will be "special issues" on special topics. One existing already is a manual on "Lumpy", the Lumped Parameter Modelling code integrated in LabData. We won't cover that here or in the manual at all.

Contents

Introduction

This tutorial tries to guide the user through the first steps in the use of LabData. The first paragraph describes hardware and software requirements for the use of LabData. Then a chapter “Installation” starting at page 5 explains how to install LabData on a computer, with some explanations on what will happen during this process. It also covers the first steps in learning how to modify data, how to input results, and it introduces the core of LabData which consists of the four tables tblSamplings, tblSubSamples, tblProcedures and tblValues. A thorough understanding of these four tables is necessary to understand LabData. In chapter “The organizational context…” beginning at page 23 the tutorial explains how to create Samplings and SubSamples for a new Set of Samplings coming to the lab. The following chapter then explains some geographical coordinate systems, how LabData deals with them, and the methods LabData offers to work with data in a geographical context (starting page 29). The final chapters following demonstrate in a step-by-step manner how data can be imported from spreadsheets into LabData (starting page 34) and exported from LabData into spreadsheets (starting 41).

Conventions

This Tutorial tries to use consequently the following conventions to enhance readability and usability:

Bold	Bold text denotes pre-defined items that you must select or click in the software, such as menu items and buttons in a dialog box or form.
`Courier`	Text in courier font denotes text or characters that you must enter from the keyboard. This e.g. might be a file path or a variable to be entered.
Italic	Italic font denotes names of objects like file names or table names.
>>	The symbols >> leads you through nested menu items and dialog box options towards a final action. For instance, the sequence **File>>Print** would direct you to pull down the **“File”** menu and select **“Print…”**.
&	The symbol **“&”** means that two keys should be pressed together. As a general Microsoft Windows© convention, available keyboard shortcuts are indicated by underlined letters. The example **File>>Print** above therefore also is available using the **“Alt&F>>Alt&P”** key combination, holding the **“Alt”** key and pressing the **“F”** and **“P”** keys.

Requirements

Hardware

LabData was first developed in the late nineties of the last century on a 400MHz PII with 256MB Ram and 4GB hard disc space. This explains the slightly outdated but still fully functional user interface with MS Access. So, any common PC available on the market today will fulfill the hardware requirements both for the server and client side. However, for ergonomic work, especially using the forms for noble gas analysis, a display with a resolution of 1280*1024 pixels or more is desirable. Also import and export of large amounts of data will create the wish for a network connection between client and server that is faster than the good old token ring. For some of the functionalities with higher numerical workload like lumped parameter models or Monte Carlo simulations of noble gas temperature errors, the CPU on the client should be faster and more Ram is desirable. A 2 GHz PC with 1-2 GB Ram as server and client and a 100 Mbit/s or 1 Gbit/s Ethernet surely enables you to get the results faster from LabData than your brain can contemplate about them.

Software

LabData is designed as a client-server database system, where the data reside on a central server accessible from the local area network. Every user has an own Graphical User Interface (GUI) on her/his PC that can be adjusted to the special needs and preferences of this user. This GUI connects to the database server via the network and a Windows "Data Source" that uses the Microsoft Open Database Connectivity® connection (ODBC). In principle server and client can be on the same machine, but then of course you will not have the possibility of having several persons working with the same data simultaneously. So LabData suggests that Microsoft® SQL Server is running somewhere in the local area network as database Server. The Microsoft® Office Professional software package must be working on every client where the Graphical User Interface (GUI) of LabData will be installed, and it is necessary that an ODBC connection can be established to the server computer running Microsoft® SQL Server. This manual assumes that these prerequisites are given. If this is not the case, you should ask either your local computer administration to complete the necessary installations. Or you must buy at least one license of Microsoft® SQL Server and Microsoft® Office Professional and install it on a network PC. During the export and plot procedures LabData sends its output (numbers and plots) to MS Excel. Therefore, the user must hold a licence of both MS Excel and MS Access, which are contained in the MS Office professional package. Further, if you want to get the benefit of seeing locations and colour-coded concentrations at a glance, you need to install Google Earth Pro.

Licence

The intellectual property (IP) of LabData was developed at the GGA-Institute in Hannover, Germany, and was transferred to me (Axel Suckow) in 2004. LabData is distributed under the GNU public licence agreement (GPL). In short this means that there is no warranty of any kind, and that the software may be freely distributed as is and may be modified if all modifications will be distributed under the same GPL agreement (share-alike agreement). Upon first startup there will be a dialogue window with the licence agreement which must be accepted. At the second startup this same licence agreement will show a checkmark "Do not show this message again". If this is checked (and the licence again accepted) the licence agreement will not show up again until you install a new version. The licence agreement can always be displayed from the startup screen when pressing the button Licence: 

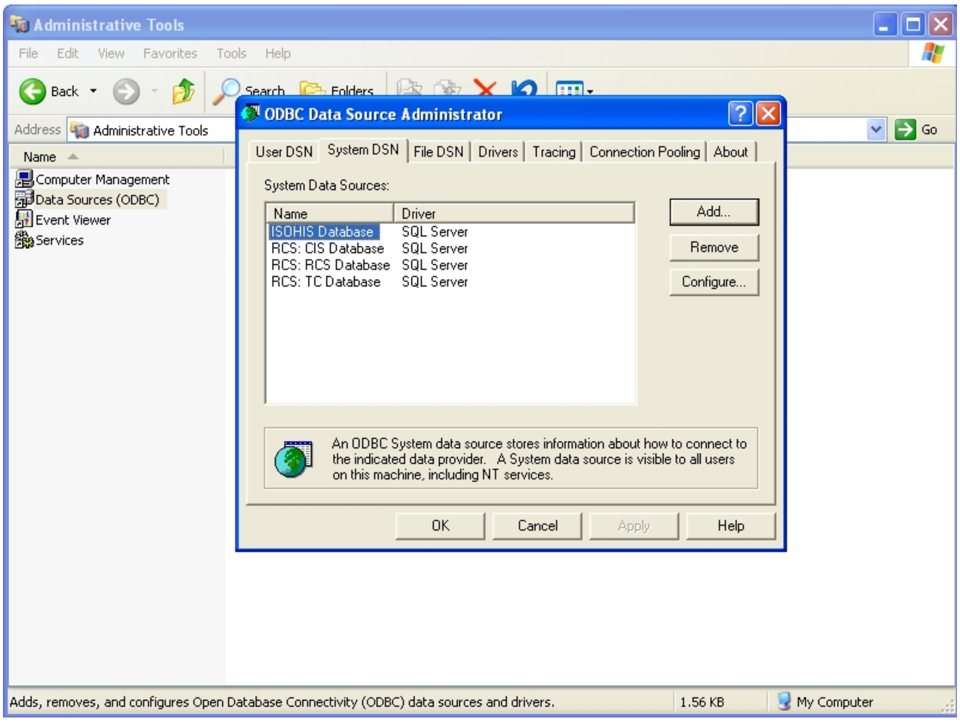
Installation

The Tutorial assumes that a drive `D:` is existing and that you copied the files completely to a directory called `D:\Samples`. If you do not have a drive `D:` or if you prefer to use other paths your mind should be able to change these variables accordingly throughout this Tutorial.

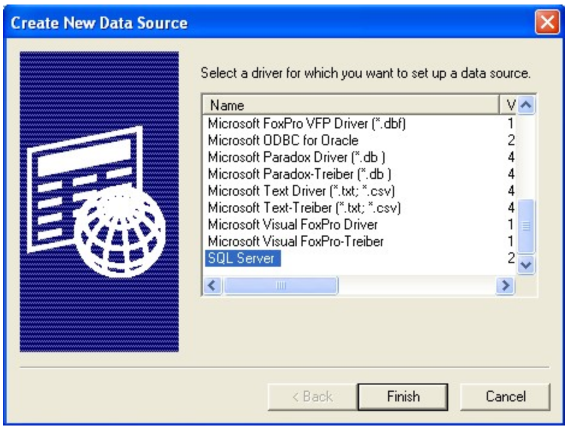
After you copied the installation files to your hard disk and after the Microsoft® SQL Server software is running, the very first step in installation is to create a database. This is done in the Enterprise Manager, a Sub-Program of Microsoft® SQL Server. In case you have a computer section that does all the database administration, they will probably do the creation of this empty database (they did in all places where I installed a productive LabData environment so far). If not so, you must open the Enterprise Manager (**Start>>Programs>>Microsoft SQL Server>>Enterprise Manager**). Here in the left part of the window you select the Server Group and Server of your choice. Right click on the folder **Databases** and select **New Database…**. Several dialogues then ask you for the name of the database, where you want to have the database and protocol files created, and what the step size is with which the database will grow as you put data in. If you are not familiar with these basic things, I would suggest to just take the default values as they are (they will do a pretty good job) and/or to read a book introducing Microsoft® SQL Server. In case no computer section administrates your database – things like backup, rights management and so on – you will wish to consult such literature anyway.

Creation of the ODBC data source

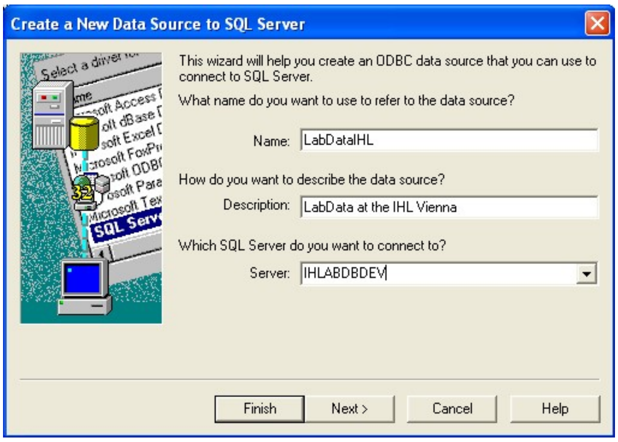
To connect between the graphical user interface ("GUI" in the following text) and the database server you need an ODBC data source. This four-letter acronym stands for the Microsoft **O**pen **D**ata**B**ase **C**onnectivity® standard of connecting clients and server. This ODBC data source must be created once for every user of LabData, therefore this tutorial will show the process in detail and step by step. ODBC data sources are managed in the Control panel: the command combination **Start >> Control Panel >> Administrative Tools>> Data Sources (ODBC)** will bring you to something like the following view:



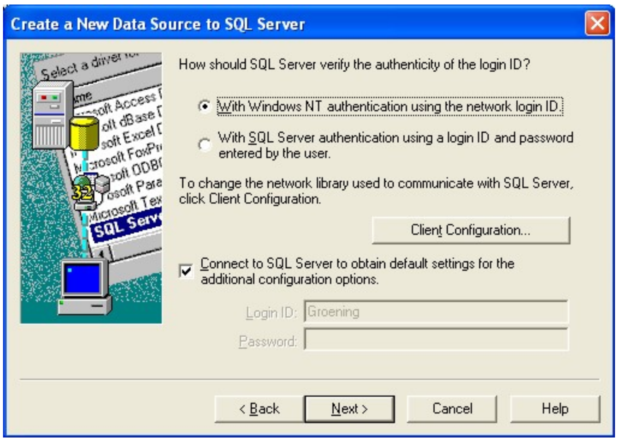
You should then select the panel **UserDSN** (for a **System DSN** you need admin rights, but both works) and press the **Add…** button. In the following selection you must choose SQL server as the according database driver:



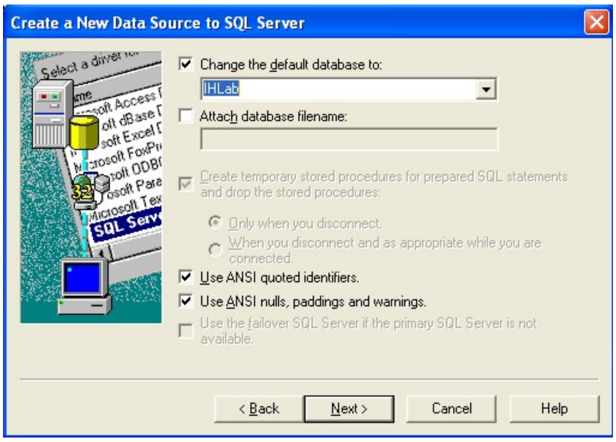
Press **Finish**. Five different definition steps will follow. In the following dialogue of step 1 you define a name and a description that is visible when you will select this data source in future and that should be meaningful in your work environment. The following picture shows what we in IAEA have chosen:



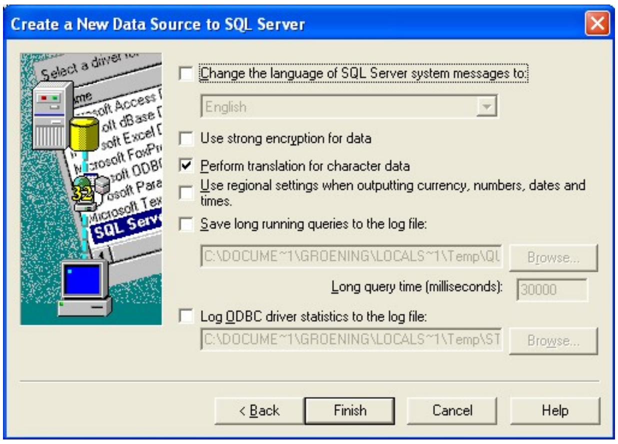
In case you plan (or anticipate or are afraid of) being the responsible administrator for several clients it’s a good idea to create a system data source and to use identical strings for *Name* and *Description* on every client: this way you don’t need to reconnect each time you distribute a new version of the GUI. In step 2 of the creation of an ODBC data source you must select how you want to organize your authentication to the database server. All LabData installations that I had to administrate so far used the Windows NT authentication with the network login. This is the easiest to administrate access rights to the tables.

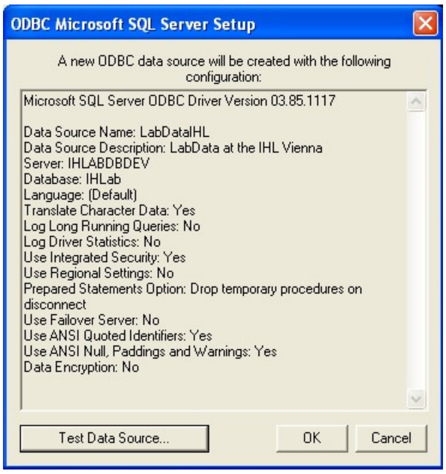


If you are interested in details on rights management you should consult the LabData manual and any book on SQL server. In this Tutorial it is assumed that you have full access with no restrictions to the SQL server database – or spoken in the SQL language: you are the **d**ata**b**ase **o**wner (dbo).
Step 3 of the creation of an ODBC data source asks you to select the default database. Here of course you must select the database you created during installation, page 5 of this tutorial. The database in the isotope hydrology lab in Vienna listens to the wonderful name `IHLab`:



In the last two steps you must define in which language you want the fonts etc. of LabData defined and you will be shown a summary and be told that the whole process was finished. The according screens follow:



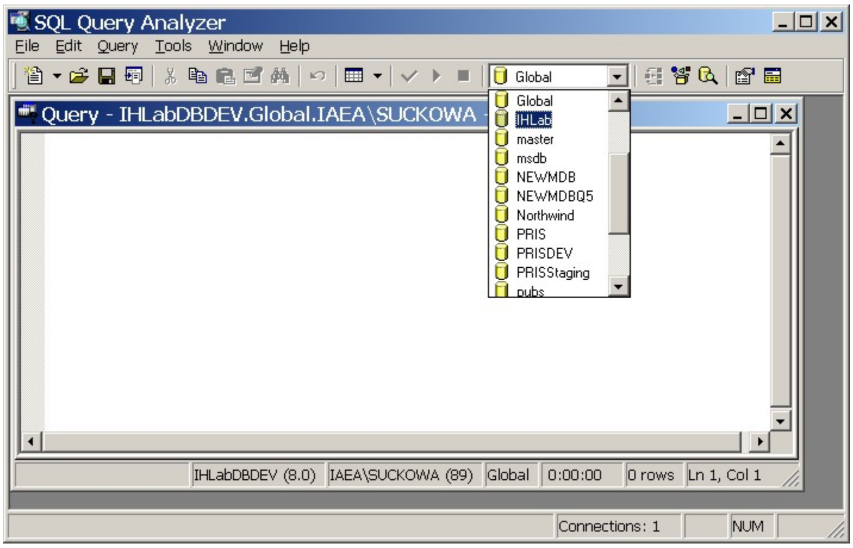


Of course, you can try to test what you did using the button Test Data Source… but I never experienced any problem with the process, therefore I skipped this picture. But please do not forget to press the **OK** button, because otherwise you must repeat all the steps shown…

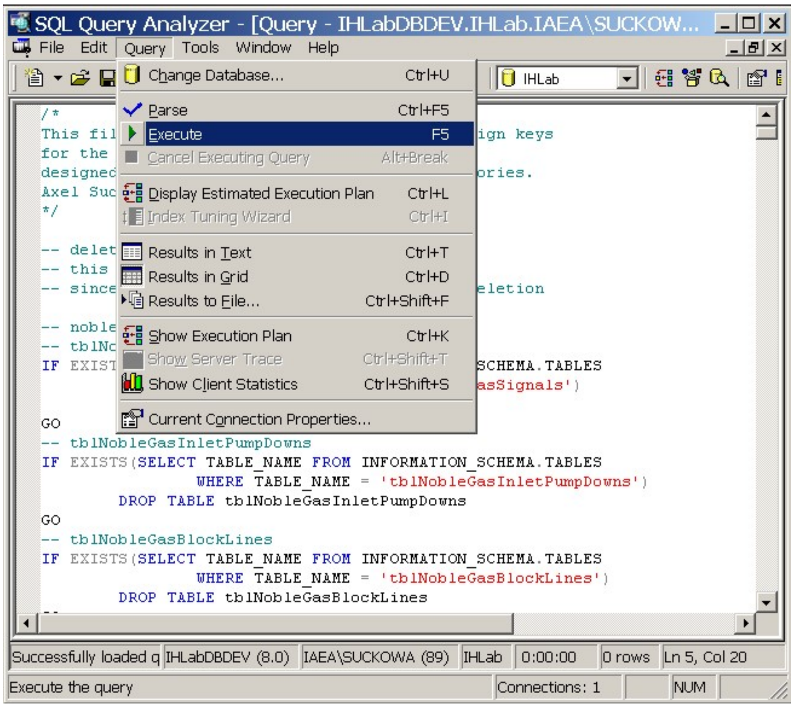
The following paragraphs show you in detail how to create the tables of LabData, how to establish a connection from the GUI on a client PC to the tables on the server, how to fill the tables with the first data and what kind of numerical procedures and triggers exist on the server side. For a newbie it is suggested to run through this process at least twice: once to test and see how the process works, and a second time to establish a new “juvenile” database structure to work with. In case you learn how to use LabData in an environment where an installation of LabData is running already I would strongly suggest creating a “training” database on your server and to run through all the tutorial steps at least once using an ODBC connection to this training database.

Creation of the database tables

If you have a look at the database tables (In the SQL Enterprise Manager select your database in the folder “Databases” and click on “Tables”) you will see that it does not contain any tables created by the user, but only tables the system itself needs for administration purposes. To create the user table structure of the database that will contain the data and the part of software that runs on the server machine, some programs must be executed. These programs are written in the “Structured Query Language” (SQL) and are started using the Query Analyzer, another Sub-Program of Microsoft® SQL Server. So please open the Query Analyzer now (**Start>>Programs>>Microsoft SQL Server>>Query Analyzer**). During opening, this program connects to your database server, but it does not necessarily connect to your database. Your screen should look similar to the following figure, and you can select your database (in case of Vienna it has the name `IHLab`) in the list box in the upper toolbar.



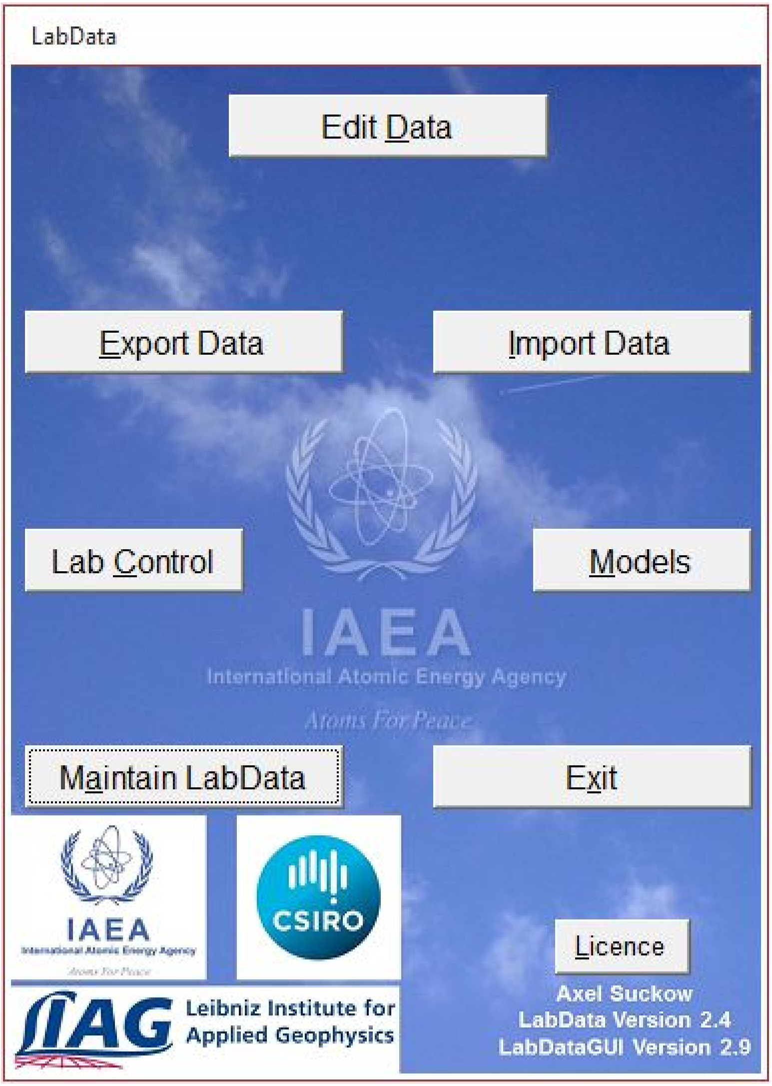
The tables of LabData are created running the program in file `LabData Tables Creation.sql`. So, open this file from the directory to which you copied the files (In SQL Query Analyzer: **File>>Open**, browse to the drive and directory and open the file). The program is executed selecting **Query>>Execute** from the menu or pressing **F5** or the toolbar button with the green triangle to the left: .



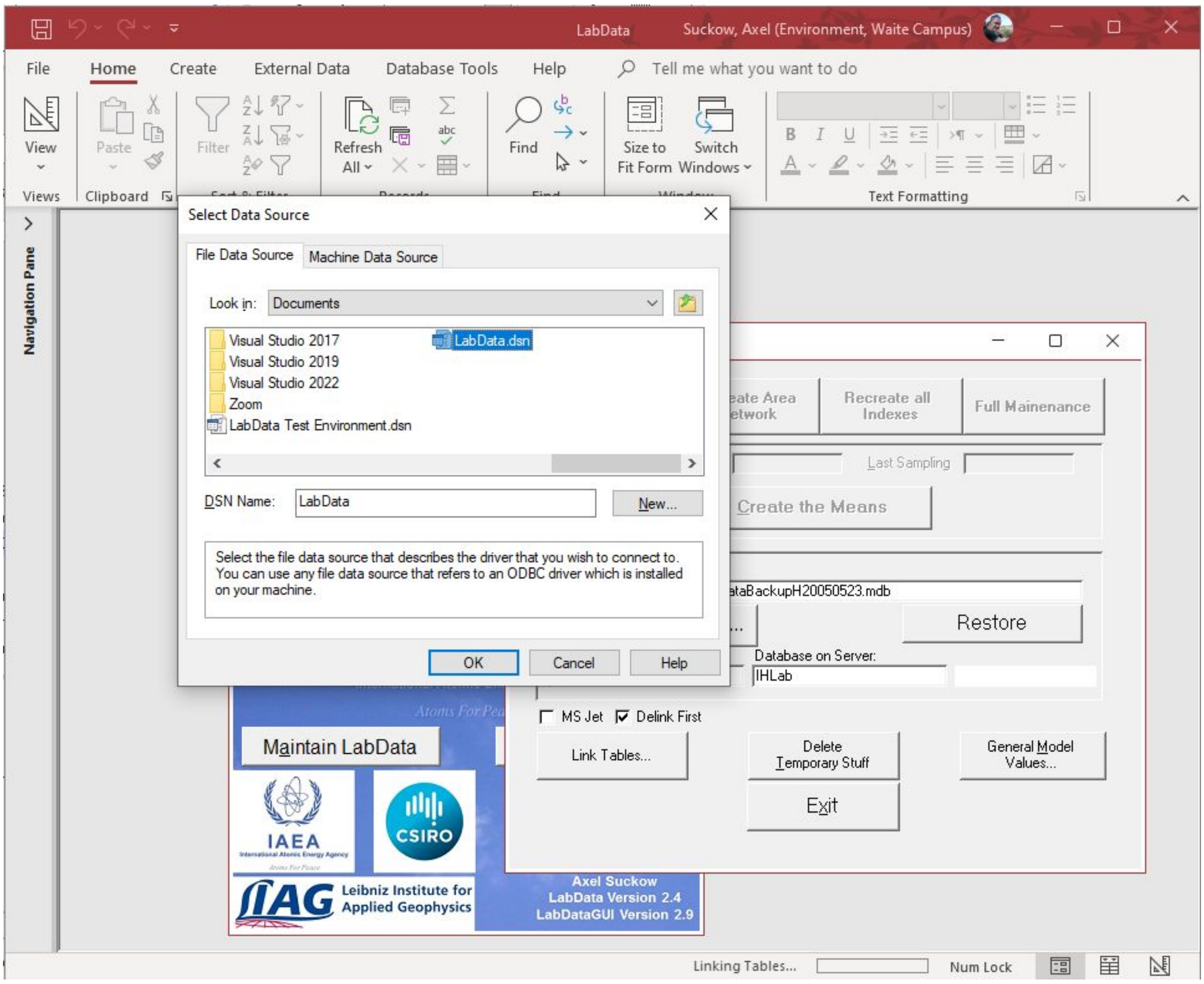
After this program finished successfully you must create some views. Views can be used the same way as tables, but they do not store any data, they just combine some of the content in the tables in a new way. The views you need for LabData can be created by running the file `LabData View Creation.sql` in Query Analyzer as described above.

Connection of the Database to the Graphical User Interface

To understand what these two programs did, it is useful to have a look at the tables again (switch to the Enterprise Manager, open your database and click at **Tables**). You will find several new tables that have been created, like *sectblSMeans* or *tblAreas*. Besides it created a lot of views, starting its name with vw like *vwAreaExposures* or *vwSubmitterSamplings*. Therefore, it is time to establish an ODBC connection and to connect these tables to the Graphical User Interface. So please open the GUI (graphical user interface) by a double click on the file *LabDataGUI.mdb*. You must accept the disclaimer, which shows up, because otherwise the graphical user interface will close again. After doing that you will see a picture similar like the following:



Seeing this first window select the command **Maintain LabData** and in the next Dialogue the button **Link Tables**. A dialogue shows up, that allows you to connect with any kind of external data sources. Here **ODBC Databases ()** must be selected.



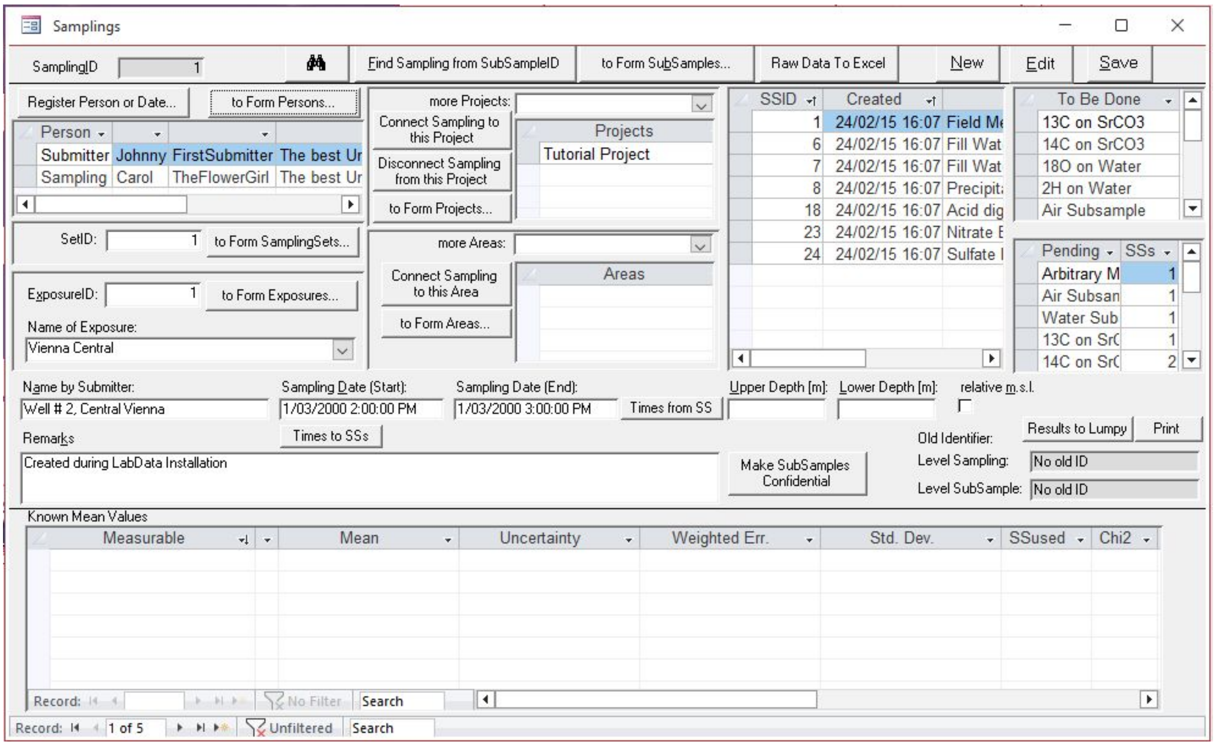
In the following dialogue, you select the ODBC Data Source you had created following pages 5 to 9 and click OK. Of course, this step would not work if you did not create the tables and views as described on page 9. Close **Maintain LabData**.
After the connection to the tables on the database server is established now, it is time to see how we can put data into the system.

Input of Data

Before the input of measurement results is possible, even before SubSamples can be created, you must define the methods how the SubSamples are created and the procedures how the results are obtained. In the LabData Manual you will find detailed information on how this very systematic and time bearing approach can be followed. Here in the Tutorial, we will start with a predefined dataset to work with. To have at least this minimum of data to work with, we will write them to the database using an SQL file. So, switch to Query Analyzer, open the SQL file `LabData Tables Fill for Tutorial.sql` and execute it.
The tutorial will now proceed with data navigation and editing and creation of new data. Don't worry if anything goes wrong or if some of the data do not look like they do on the screenshots in the tutorial: they only do if you follow the tutorial very literally. In any case we will destroy and rebuilt the whole table structure later – and so all you edited during these chapters – to give you a "juvenile" database for your own purposes.

Navigation in Samplings, SubSamples, Values

Switch back to the LabData GUI. On the entry Screen of the LabData GUI select the button **Edit Data** and on the Data Editing and Input dialog select **Samplings**. You will see something like this:



Samplings

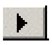
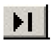
This form *Samplings*[1] is the most central one in LabData and if the software will be completely installed and "all users do their duties", it displays a lot of information about any given Sampling. Those who already read through the Suckow & Dumke (2001) paper know why. We will come back to this form frequently, every time looking at different details.

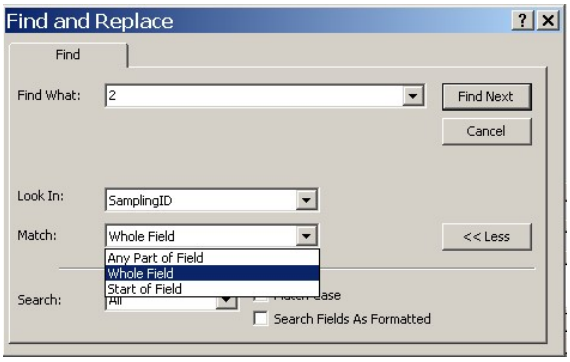
Many forms in LabData are designed like a window looking to a specific table. This window here looks to the table *tblSamplings*. Every entry in *tblSamplings* can be thought of as one *act of taking (Sub)Samples* and this act is called "Sampling". Please note: The entries into this table do not describe the "sample". To be precise, the term "sample" does not even exist in LabData! Instead, the term "SubSample" is used, which will be described in detail in the next paragraph (see page 15). For the moment the intuitive understanding of a SubSample is sufficient. The entries in *tblSamplings* describe how SubSamples were taken: where, when, by whom, how many SubSamples, what should be done with them and so on.

So, what does this form *Samplings* tell us now about the first sampling in the Database? Starting from the top left it tells us that we have sampling number one here (*SamplingID*). Again, a basic principle in LabData: You will find all unique identifiers for a certain table in the upper left of the form. The sampling came to us through a *Submitter* who is somebody called `Johnny FirstSubmitter`, and it was taken by `Carol TheFlowerGirl`. This sampling belongs to the first *Set,* and it took place at an *Exposure* with the name `Vienna`

[1] Just to avoid that anybody more wants to tell me that the word "Samplings" does not exist in the Anglo-Saxon languages: I am fully aware of this fact. The point is that a database needs abstract concepts. The concept behind table "Samplings" is NOT a table of samples. It is not a table of physical things but a table of actions. A table of several entities describing what we call "The action of taking samples". The second point is that nobody really takes samples – and that is why this word "sample" in this tutorial only exists on this page. While sampling everybody takes SubSamples. Just be patient, let the concepts enter your mind and follow the text.

Central. The term *Exposure* for groundwater samples codes for the well, but the concept behind is more general and comprises also things such as drill cores and will be described in more detail later (see section Exposures on page 29). Also, *Areas* and *Projects* are organizational structures to which we come later, but they are self-explaining. They will both have an own Chapter in the manual, and *Areas* even will be explained in this Tutorial beginning at page 31 so I would ask you for some patience. Just note that this Sampling belongs to the Tutorial Project, since we will use that later. The same holds true for button **Register Person or Date…** which will be explained on page 26. The larger field right of *Projects* and *Areas* then contains a whole small table with superscripts *SSID* and *Kind*. This sub-table tells you that during this sampling some field measurements were performed, that at least two bottles of water have been filled and carbonates have been precipitated. These are the *SubSamples*, and we will come to them in detail in the next chapter. Further on the right you will recognize that the analyses *To Be Done* on these SubSamples include ^{18}O, D, Tritium and Radiocarbon, and it seems that none of them was performed yet since all are *Still Pending*.
So, let's go on with the *Name by Submitter*. This field tells you how the sampling was called, by the one who had performed it. Normally this is the name of the well the water comes from or a number within a campaign. Right of that field you see the *Sampling Date* with a resolution of seconds, which will help you sometimes to distinguish different Samplings and their SubSamples if the one who has taken them was very busy that day. You can also see how long the whole Sampling was, because an end date (and time) can be provided. *Upper Depth* and *Lower Depth* in case of a drill core could be the depth interval that was sampled. In our case the two are empty, since this is a groundwater sample, and the upper and lower depth of the screen for an Exposure serves the same purpose. The field *relative m.s.l.* indicates if these depths are measured relative to the mean sea level or the ground surface. In case of SubSamples taken from a sediment core this would normally be the depth interval the SubSample comes from relative to the top of the core.
Let's see what else we have in *tblSamplings*: go to the next dataset by pressing the **PgDn** Key. In the very low left end of this window you will see that you are now in Record 2 of 5:


These record counters tell you where you are in a certain table and how many rows the whole table has, that you are looking at. Again, this feature will be found on all forms being a window to a table. You can move within the table also clicking with the mouse on the small buttons aside of this number two:  will make a step forward (same as pressing the **PgDn** key),  will make a step back (same as pressing the **PgUp** key),  will go to the last sampling (same as pressing the **Control&End** key) and  will move to the first sampling (same as pressing the **Control&Home** key). But the user can also directly navigate e.g. to the fourth sampling by entering a 4 into the number field of the record counter. But careful: this fourth record can be changing depending on how the entries in tblSamplings are ordered. To test this fact, please move the mouse pointer into the field *SamplingID*. Then execute the menu command **Records>>Sort>>Sort Descending**. Now the sampling with SamplingID 5 is record 1 of 5, and record number 4 would be the one with SamplingID 2.
So, let's move to the Sampling with SamplingID 2 now. To do so and to reach this record independent of any actual sorting, you can move the mouse pointer into the field SamplingID and press the button with the binoculars (). In the following Dialogue *Find and Replace* you can define in which field the search is done and if the search should be performed on the whole content, the beginning, or any part of it (which of course means differences in search speed):

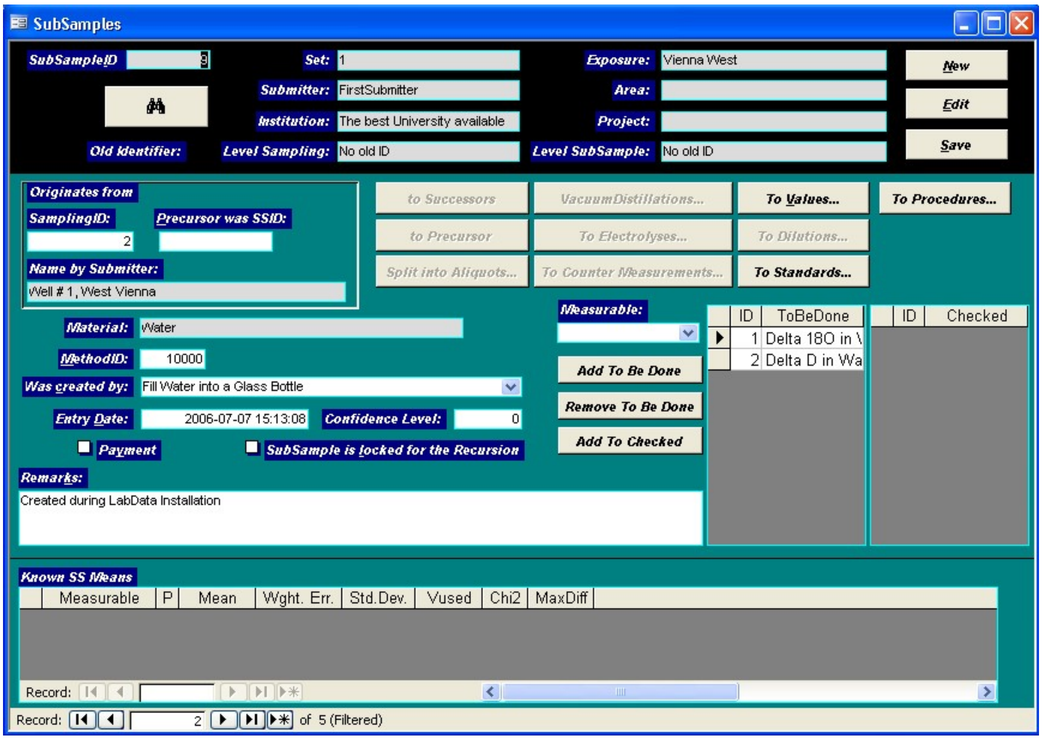


This dialogue works on most forms of LabData and is a way you can search in nearly every field of the forms any content you need. For instance, using `Name by Submitter` as the place where to search (*Look In*) you could search for `ourths` and find `FourthSampling` if necessary.

SubSamples

If you are in the record with *SamplingID* = 2 you see that this Sampling should have five different SubSamples ("SubSamples" will be abbreviated sometimes as "SS" in the following text). One SubSample is for field measurements, two were prepared by filling water into a bottle, one prepared by precipitating carbonates from water, one prepared by acid digestion of carbonates. This leads to the first intuitive understanding what a SubSample is: A SubSample is a well-defined entity of material belonging to a Sampling. For example, in isotope hydrology or groundwater chemistry we go out to the well and normally take several SubSamples at a time: these are for instance a small bottle for stable isotope measurement ($D/^{18}O$), a 1l bottle for tritium measurement, perhaps we create a precipitate of carbonates for ^{13}C or ^{14}C analyses, take a bottle of acidified water and of filtrated water for different chemical analyses and so on. Each of these entities is a SubSample and deserves to receive an own unique ID, the SubSampleID or shorter SSID. In this special case each SubSample is a bottle, but this could be also a bag of sediment, a piece of rock etc.

Let's have a more detailed look into these SubSamples by pressing the button named **to form SubSamples…** on the upper right of the form. You will end up in a new form looking like this:

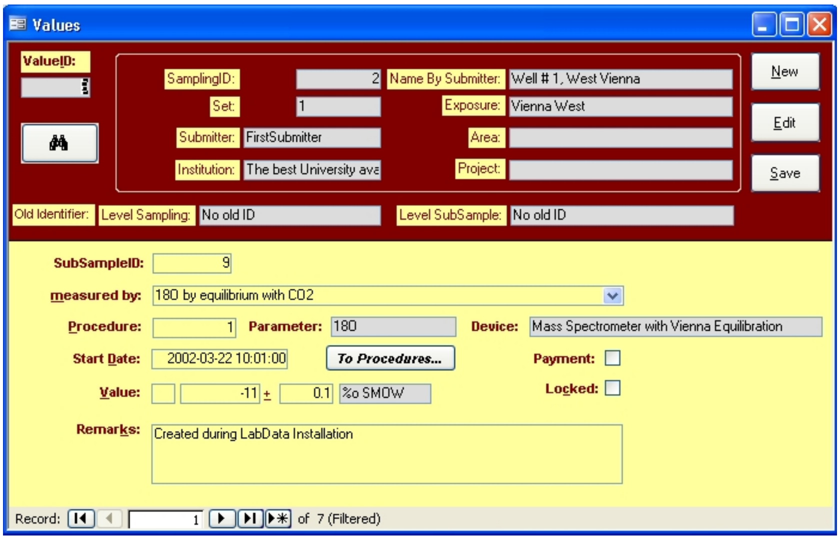


This form tells you some of the details about the SubSamples: what material they are, how they have been prepared and when, and so on. Again, the unique identifier for SubSamples (the *SubSampleID* or shorter *SSID*) can be found in the upper left of the form. For the internal work in a laboratory this SSID is probable the most important identifier in LabData. The whole GUI was designed in a way that every search, input, output and editing problem should be solvable if the SSID is known.
Form SubSamples has a larger header in which information about the origin of this SubSample is displayed. Here again you find who sent the SubSample, where it came from, and what the organisational context of this SubSample is (Set, Project…). But here this information is read-only and just gives you the navigation feedback, that you are in the right SubSample while editing.
Some things deserve your special attention concerning SubSamples:
First, proceed to SubSample number 19 (on page 13 you learned how to navigate through records) and you will see that this SubSample has a Precursor, namely SubSample 11. This means that the CO_2 of SS 19 was produced from SubSample 11 (the precipitated carbonate from groundwater) by using the technique of acid digestion. This example illustrates that LabData is capable of dealing with SubSamples of hierarchical origin (SubSamples 2, 9, 10 and 11 were prepared in the field, SS 19 was prepared from one of them in the laboratory). And it illustrates the principle, that the unique SSID characterizes not the "Sample", which does not even exist in LabData, but the *SubSample in a certain state in the laboratory*. In real live the carbonate precipitate is very different from the CO_2 gas evolved from it (in look, in feel, in the possibility of spilling it, concerning the container…), and the different and unique SSIDs in LabData reflect this fact. The hierarchy of SubSamples could have any depth and there could be any number of SubSamples on any hierarchical level, restricted only by storage space (which is really, really huge nowadays). So it is very useful to think of SubSamples as a piece of material having a defined state within the lab, and to create a new one if something important changes. Any important preparation step would be an example for such a change: in the case of a tritium laboratory the electrolytic enrichment of tritium changes the interpretation of the subsequent radiometric measurement and therefore one should create a new SubSample. Or a change of the chemical state as in the present case from precipitated

$SrCO_2$ to CO_2 would be a reason for me to define a new SubSample. Although the ^{14}C value for precursor and successor should be the same here, using two SubSamples helps you to keep track on what happened.
Second, if you look at the record counter of the form you will see the word "Filtered", which means that you do not see the whole content of tblSubSamples but only a certain part of it. To be more specific, you see all those SubSamples that belong to Sampling 2, since you came to this form via form Samplings. Although this should not be done now since we want to proceed to form Values, the filter can be removed pressing the toolbar button with the funnel (). Coming from samplings to SubSamples filtering is useful, because the SubSamples table contains typically several ten thousand entries whereas when you are dealing with those for one Sampling you typically deal with a dozen or less.

Values

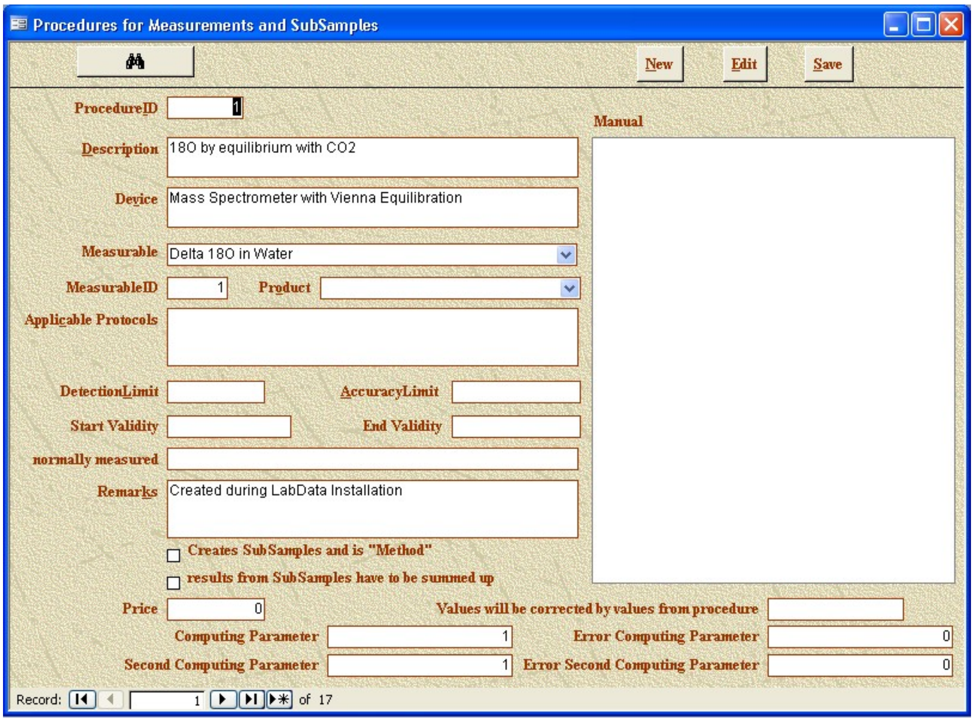
Pressing button **To Values** for SubSample with SSID 9 opens another form displaying measured values for this SubSample:



Again the records are filtered and only those values belonging to SSID 9 are displayed. Here filtering is even more useful since the table tblValues typically contains more than 100 000 rows. Again in the header you will find the information describing the origin of the sampling. Moving through the values for this SubSample you will be able to see 7 values, which are 3 and 4 values for $\delta^{18}O$ and δD respectively. Scroll through them to see what they are. You will notice that every value has a *Start Date*. You might also notice that there is a small checkbox called *Locked*, which indicates that you can exclude every value from further statistical treatment. Maybe you even had noticed before that this kind of checkbox existed for SubSamples as well.

Procedures

If you want to see some more details on how the values were created, press the button indicating **To Procedures…** Another form will open:



The details of this form again will be described in the LabData Manual. For the purpose here only a few things are important:

1. The procedure defines the parameter to be measured (Tritium, ^{18}O, Deuterium…) as well as the units of the measurement results (like TU or ‰ SMOW) and the material of the SubSample on which any procedure can be applied. The combination of Parameter, Unit and Educt Material is called a *Measurable*. Although this information is shown in the Values form, in reality it is stored in an own table called tblMeasurable, which is linked to tblProcedures.
2. The *ProcedureID* is one of the very rare unique IDs in LabData that is defined by the user. So you have to type in a number here, and LabData will make sure that you are not able to define duplicates.
3. Recipes how to create a SubSample are stored in tblProcedures as well. Just for distinction to the procedures that can only create measurement results and no SubSamples, we will call them *Methods* in this tutorial and the manual. The only difference between a method and a procedure is that the selection **Creates SubSample and is "Method"** is marked for a method. Of course any procedure can be a recipe to produce values and SubSamples at the same time: for instance sieving of sediment would create a new SubSample (for instance the fraction <1mm) as well as a value (the weight percentage of the total for the fraction <1mm). In fact this is the reason why the user is able to define the unique ID of a procedure. In the Hannover and Vienna Laboratories the convention exists that methods have large numbers. In these labs every procedure with an ID larger than 9999 can create SubSamples.
4. Things like the detection limit and the limit of accuracy of a certain method, possible laboratory protocols within LabData and even a whole manual describing the procedure can be stored and accessed with this form.

The best for you to do now would be to eliminate the filter that you still have coming from form Values and to have a look at the different procedures that exist in LabData at the moment. This will give you an idea on how procedures are used.

Editing and Creating Data

Imagine now, the information given was wrong (never trust a Database!!!), and the Sampling 1 should be labelled correctly “Vienna Central #11” with a sampling date not several years ago but yesterday. So close the forms for procedures, values and SubSamples by pressing **Control&F4** in these forms and go back to form *Samplings* and try to change the field *Name by Submitter*. LabData obeys the general Microsoft Windows convention that letters that are underlined can be reached with the keyboard combination **ALT&Letter**. So if you type in **ALT&a** you will be in the appropriate field.

If you try to edit this field, it will probably not work: for security reasons every form that directly displays a table is locked on opening. This is one of several means that try to make the user think about what she/he is doing. Press the button **Edit** in the upper right of the form and try again. This time a change of the *Name by Submitter* should work. Proceed to the field *Sampling Date* and put the date of yesterday in. These changes now are not yet accepted in the database table. They are only displayed in the form, up to now. Storage in the database will occur only if you press the *Save* button in the upper right corner of any form (**ALT&S**) move to another record (pressing e.g. the **PgUp** or **PgDn** keys) or if you close the form. Until then, pressing **Esc** will bring the data in the form back to the stage before editing, and keep your data in the underlying table unchanged.

To create a new record either use the *New* button, the menu command **Records>>Data Entry** or press the button  which is situated on the toolbar or on the record counter of many forms. You will be given an empty form that you can fill with the new data. This method works on any form representing a table. So create a new sampling now and input `TestSampling` as *Name by Submitter*. Table *tblSamplings* accepts entries that are nearly empty, so store this record by pressing **PgUp**.

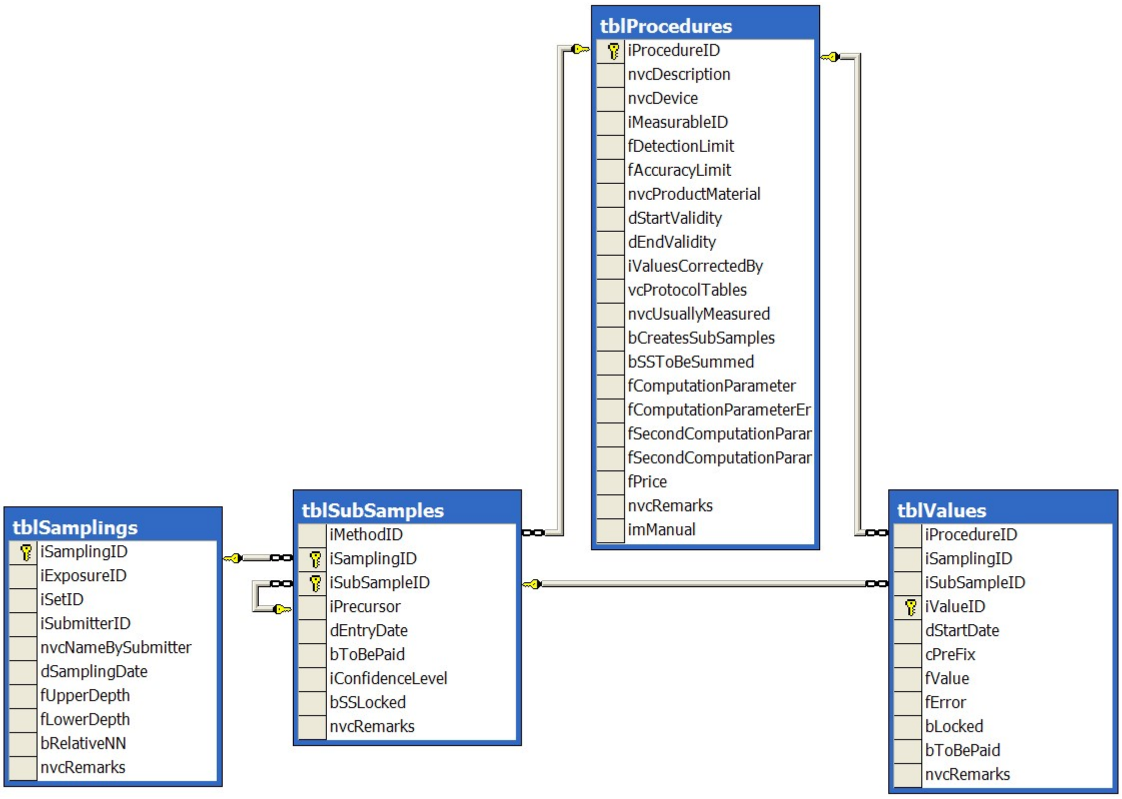
Try out the security measures of LabData: there are some ways that LabData gives you feedback on your inputs, and there are many things that are not allowed during input of data. For instance you cannot connect a sampling to a submitter, set or exposure that does not exist (The concepts behind Submitters, Sets and Exposures are explained in more detail on pages **Error! Bookmark not defined.**, 24 and 29 respectively). Go to field *ExposureID* and type in `4`, proceed with the mouse to the field *Name of Submitter* and select `Dr. Johnny FirstSubmitter`. You will recognize, that on leaving the field *ExposureID* the name `Vienna South` showed up in field *Name of Exposure*, and that after selecting `Johnny FirstSubmitter` his ID (`1`) showed up in field *SubmitterID*. Both these connections of the fields give you feedback on what you are doing and enable you only to select those possibilities that you are allowed to. Now try what happens if you input `10` into field *SetID* and want to submit this to the table by pressing *Save* or **PgUp**. You will see an error message that this entry was not accepted. The reason is that Set 10 does not exist. For a Set not existing you must not enter Samplings. Since there exists only one Set up to now and its number is 1, the only allowed value for Sets up to now is 1. A similar error message would be displayed, if you would try to input values >5 for the *ExposureID* or >1 for the *SubmitterID*. Just try it and you will notice that also the feedback giving you the name of the Exposure and Submitter will fail.

After we created a Sampling, let’s create a SubSample. Press the button **to form SubSamples…**. Form SubSamples shows up and a dialogue that tells you that there is no SubSample existing yet, and asks you if you want to create a new one. In case you press **Cancel**, both windows will vanish again (and no SubSample will be created). In case you press **OK** you will find the SamplingID from which you started already in the appropriate field.

All other fields in the form are still empty and you will not be able to save a SubSample in this state. This is due to the fact, that LabData does not know yet *how* this SubSample was

created. So select the method `Fill water into a bottle` into field *Was created by*. Also here the field *MethodID* will be filled with the appropriate number and the field *Material* will be filled with `Water`. Only now this SubSample is ready to be sent to the table (*Save* or **PgUp**). Note that also here it is impossible to create SubSamples for Samplings that do not exist (for instance try `25` for the *SamplingID*), SubSamples with Precursors that do not exist (for instance try `20` for a *Precursor*), SubSamples with Methods that do not exist (for instance try `100` as a *Method*). And you will note that the actual timestamp was attributed to the SubSample as creation date.
After we learned how to create Samplings and SubSamples, let's learn how to create Values. Navigate back to the SubSample that was just created and press **To Values…**. LabData will inform you that no value exists yet for this SubSample and prompt if you want to create a new value. Selecting **OK** you will find yourself on form Values, with the SubSampleID and SamplingID already filled with the correct IDs. In *measured by:* select `18O by equilibrium with CO2`. Moving to the next field the correct unit and parameter will be indicated. The only fields you have to fill now are the *Start Date* and a *Value*. Also here **PgUp** or **PgDn** will send the new value to LabData. And if you try to input a procedure that does not exist into field *Procedure* (for instance try `100` here) or if you try to input an entry for which no value is given (create a new value and leave *Value* empty), LabData will not accept the entry.

Since we learned now how to use the forms for Samplings, SubSamples and Values it is useful to have a look at the theoretical framework behind, what I like to call "the topology of the tables". The official name is a diagram of the relations or a "Bachmann Diagram":



What you see in this picture is the column headers of the tables named *tblSamplings*, *tblSubSamples*, *tblValues* and *tblProcedures*. The lines running between the table headers are called "relations" (that's why this kind of databases is called a "relational database") and they indicate that the same content is in the appropriate columns the line connects. The small key

in these lines () represents the primary key and the unique part of the relation, the small lemniscates () represents the foreign key. So for instance the *iProcedureID* is a unique value in *tblProcedures*, but in table *tblValues* many entries can have this *iProcedureID*: all those values that are created using this procedure.
Just to keep in mind: The first table (*tblSamplings*) contains the data about the *sampling action* in the field, not about the "samples" – an entity that even does not exist in LabData - nor about the SubSamples (the bag of sediment, the bottle of water…). The data on any of these SubSamples are stored in *tblSubSamples*. The method used for preparing a SubSample is given in *tblProcedures*. SubSamples can be produced during the sampling action in the field (e.g., bottles of water for the tritium, stable isotope and radiocarbon analyses and the bottles of filtered and acidified water for geochemical analysis) or during laboratory processing (e.g., taking aliquots, sieving, preparation of a counting gas or of AMS targets).
The measured values are stored in *tblValues*. The procedures used for the measurements are stored in *tblProcedures*. The values table is mainly for storing only the numerical values and will have the largest number of rows of all the tables in the data model (typical 1e5 to 1e6). It is therefore necessary to keep the number of columns ("fields") in this table small and to avoid strings and redundant information wherever possible. Therefore, the names of the *parameters* (e.g., "Tritium", "18O", "14C") and *units* (e.g., ppm, TU, pMC, ‰) are stored in other tables. In fact there exist several more tables not shown in the figure from which you can select the parameters, units, materials and measurables. Also detailed information about the procedure itself can be stored in the table *tblProcedures*: the field *imManual* can contain a whole text file (e.g. MS Word or pdf) with the full description of a certain laboratory procedure. This way *tblProcedures* can contain the whole information concerning laboratory and field methods you need for certification according to ISO 9001!
Note that a 1:n relation exists between the procedures in *tblProcedures* and the SubSamples in *tblSubSamples* (many SubSamples can be produced by the same method), between procedures and values (many measurement values can be produced using the same procedure) and between SubSamples and values (methods like gamma spectrometry and ICP-MS result in many values for the same SubSample, and measurements can be repeated).
It is possible for recursive relationships to exist in the SubSample table: Many SubSamples can be produced from one SubSample (the *Precursor*) using standard laboratory methods. The most straightforward example is the production of aliquots. Another example in geochemical laboratories is the sieving of sediment to obtain grain-size fractions for later chemical analysis. An example in an isotope hydrology laboratory is electrolytic tritium enrichment, resulting in a SubSample with higher tritium content than the precursor. In a radiocarbon laboratory with proportional counters, dilution with "dead" counting gas if the amount of gas was too small for the counter is another example.
This data model has the following advantages: (1) Scalability. Since the number of rows in the table for procedures is limited only by the available storage space, the system can handle a virtually infinite number of methods to produce SubSamples and data. (2) Effective use of storage space. Empty cells in the table of values are avoided: If a laboratory procedure generates only one value for a SubSample or 100, only the amount of storage space necessary is used. (3) Flexibility. The data model can be adapted to many kinds of laboratories and methods. A new procedure in the laboratory is represented in LabData within minutes by just adding a new row in *tblProcedures*. The system also is flexible with respect to the amount of details the user wants to put into the system: Whether all of the steps are to be documented – from the initial sampling to the final isotopic or chemical value (e.g., for ^{14}C: drying, acid etching, conversion to CO_2, conversion to acetylene, conversion to benzene, LSC counting, etc.) – or if only the final numbers (^{14}C for SubSample No. 7 is 35%) are to be stored, the same data model can be used with just different procedures. (4) Unique identifiers. Last, but nevertheless important, the data model provides two identifiers that are unique: One

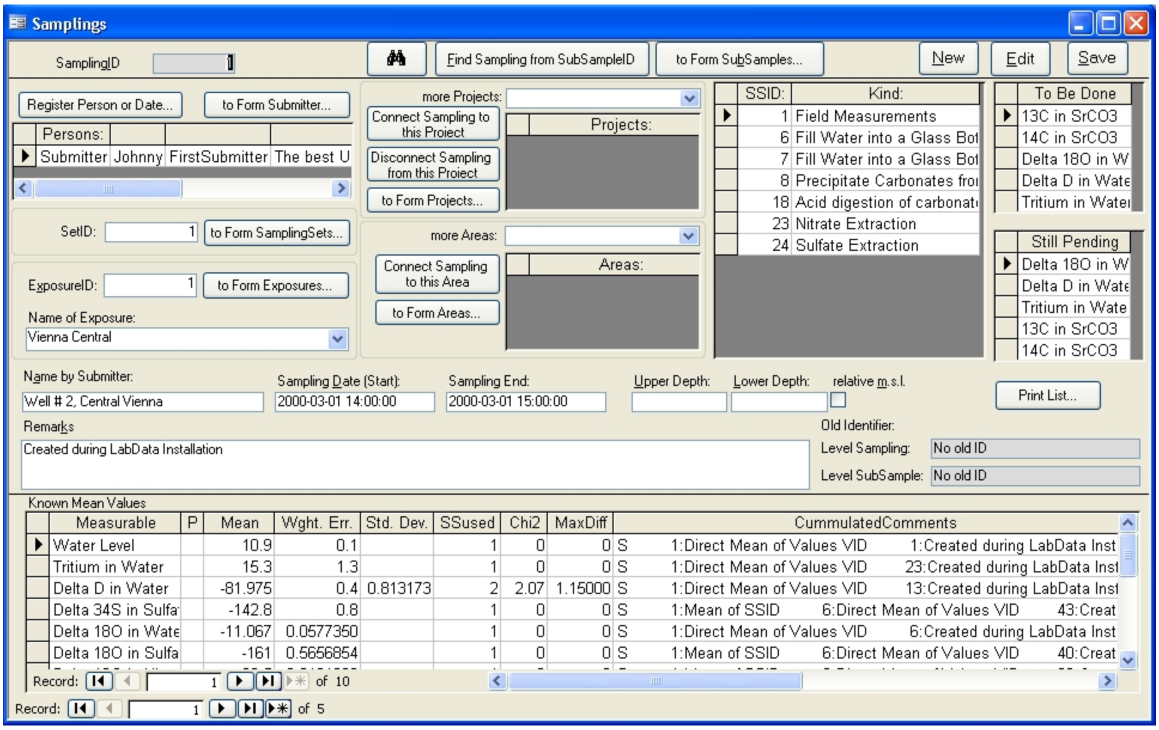
(*iSamplingID*) is unique for the problem as seen from outside the laboratory, which is the point of view of the submitter. The second (*iSubSampleID*) is unique as seen from inside the laboratory, and identifies the SubSample in a specific step of the laboratory process. The GUI in LabData is designed such that any problem should be solvable if you can give at least one SubSampleID relevant to the problem.

Numerical Procedures on the Server

OK, we learned what the meaning of a sampling and a SubSample is in the context of LabData. We learned how to find a certain sampling and how to interpret some of the information displayed. We learned how to navigate from a sampling to its SubSamples and to its values. And we learned how to create new samplings, SubSamples, Procedures and Values and how to edit the data already present. It is time now to learn about some of the automatic benefits of LabData. For this purpose we should recreate the juvenile database system.
To do this, close the GUI of LabData. Open the SQL Server Query Analyzer. Here open and start the following SQL files one after another in the order given below:

1. `LabData Tables Creation.sql`
2. `LabData View Creation.sql`
3. `LabData StoredProcedures.sql`
4. `LabData Triggers.sql`
5. `LabData Tables Fill for Tutorial.sql`

Restart the GUI. Open form Samplings, navigate to the one with SamplingID=1 and try to find a difference between what you see and the picture on page 13:



Did you notice the difference? Now in the lower part of the form a table is filled which is called *Known Mean Values*. This difference occurred since this time in building up the table structure of LabData we executed also two SQL files that created “stored procedures” and “triggers” on the server side of LabData. Doing this we created some kind of intelligence on the server side: every time we add a new value for a certain parameter like e.g. ^{18}O, the server side knows that some statistics have to be calculated for this new value. What does this mean in detail?

Whenever a value is created newly, changed or deleted in LabData, the SQL server routines search all the values for a certain SubSample, that give results for this measurable (combination of material, parameter and unit) and that are not locked. From these values the stored SQL server procedures calculate the weighted mean, the standard deviation, Chi squared and the maximum difference between the values. If uncertainties for the values are given, also the weighted error of the mean is calculated. Note that all values will be taken for these statistical values that have the same *measurable* (e.g. ^{18}O [‰VSMOW] in Water or Tritium [TU] in Water). Not only those having the same *procedure* will be averaged. In most cases in one laboratory they will probably have the same procedure, but sometimes you will want to see statistically if different procedures give the same result (for example δD on water using a procedure equilibrating a H_2 headspace isotopically with water and another procedure reducing the water with Zinc or uranium to H_2). If the SubSample has only one value, then of course the mean value is equal to this value and no standard deviation can be given.
If the SubSample has a precursor, then LabData searches all mean values for SubSamples that have this same precursor and creates the same statistical values for this precursor – this time not directly from the values but from the SubSample means. This is repeated as long as any precursors exist. After all mean values for all SubSamples along this precursor-successor hierarchy were calculated, the same statistical values are calculated for the Sampling from those SubSample mean values where the SubSamples have no precursor. And that is what you see in the table at the bottom of the form above. So from this form you can see at a glance that deuterium values have to exist for two different SubSamples, and that from the statistical point of view the results for these two SubSamples give indiscernible results (chi-squared test).
This process of creating statistical values is triggered whenever a value is created, changed or deleted, without any influence of the user. This process is even started, if the user does not use the GUI to edit the values, but the SQL server tables directly. If you want to see the mean values for the SubSamples, press the button **to form SubSamples…** and have a look. The first SubSample (SSID 1) is the one created for the field values and has only one value: the water level in the well. The uncertainty of this mean value is directly the uncertainty of the entry in *tblValues*. The second SubSample (SSID 6) in contrast has three values for ^{18}O and two values for D. From the statistics (Chi squared>8) you will recognize that only with a probability of less than 2% the three values for ^{18}O belong to the same statistical entity: the numbers for the values are –11, -10.9 and -11.3 and the given uncertainties are ±0.1 for each value. So for statistical reasons alone, the value with –11.3 most probably is an "outlier". Or, equivalently: the uncertainties for the values are too small. LabData helps you to recognize these statistical correlations, but of course you as the user have to find the reason…
To demonstrate this automatic statistical process, create a new sampling and SubSample and create new values for this SubSample. For example if you put values for `18O by equilibrium with CO2` of -5±0.1, -5.1±0.2, -5.2± 0.1 and -6±0.2 to LabData, then you should obtain a mean value of -5.19±0.06 (std. dev 0.46) with a Chi-Square of >20. Now lock the value for –6 for any further evaluation. To do so, go to form Values and activate the checkmark *Locked*. Chi-Square should diminish to 2 and the mean value should be -5.1±0.07 with a standard deviation of 0.1. Try it!

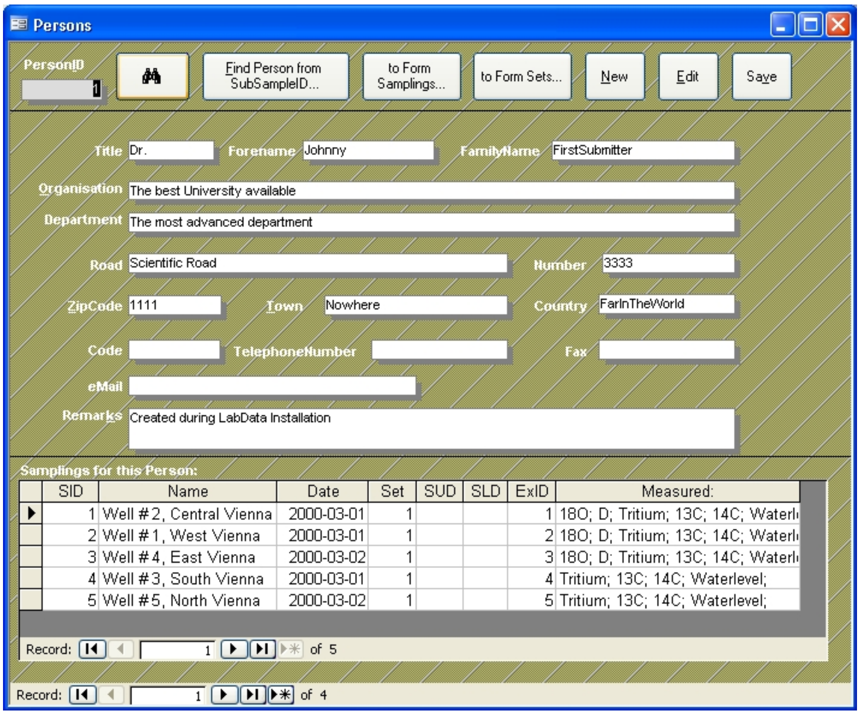
The organizational context: Persons and Sets

What you learned about LabData if you finished the tutorial up to here was mainly concerning the core of the data model, namely the tables tblSamplings, tblSubSamples, tblValues and tblProcedures together with the related forms. Up to now we do not know where to store the information about who performed the samplings and which SubSamples belong together.

Also we do not know how to create a common invoice or report. That's what this paragraph is about.

Persons

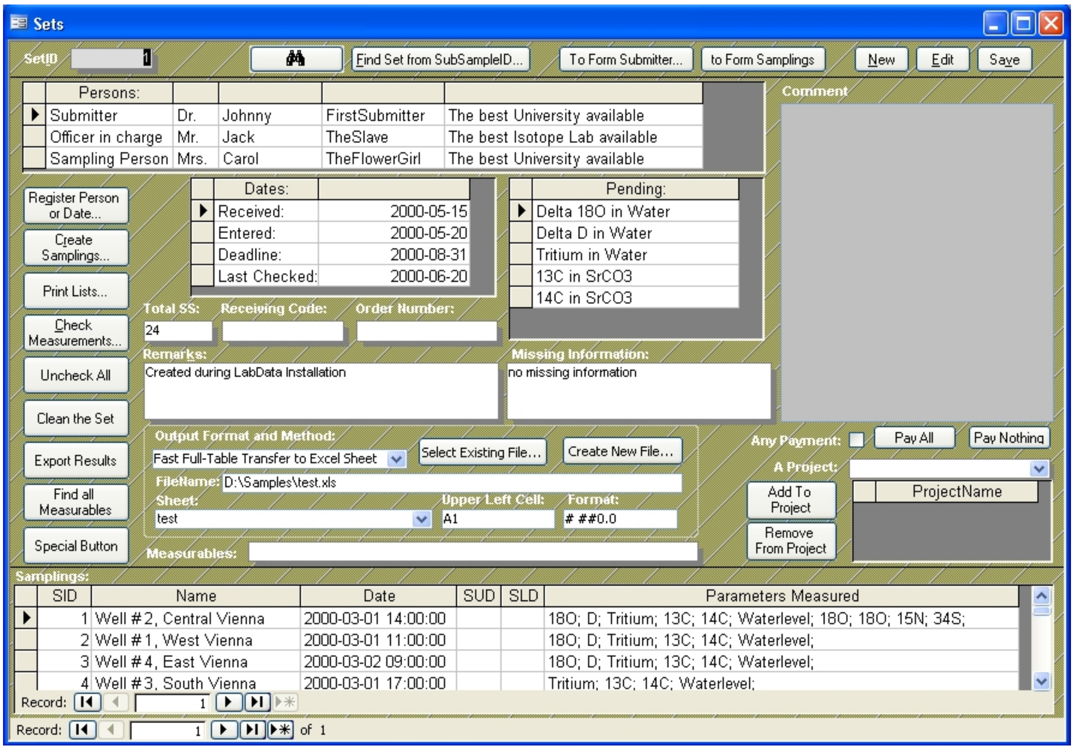
As nearly any laboratory database system, LabData has a table to store the information about persons, namely the submitter or customer. Only with this storage of "personal" information it is possible to write automatic invoices or reports. To see the information, start LabData and press the buttons **Edit Data >> Persons**. The following form will show up:



Obviously the form stores all the necessary fields for written and oral communication that describes name, title, address, phone number and so on. Besides that, this form at a glance displays how many Samplings this person did send to the laboratory and what was measured on these. If you want to see details you can press button **to Form Samplings…** and you will arrive at form *Samplings*, filtered to those samplings that are from this submitter.
In case you know the *SamplingID* or the *SubSampleID* and want to find the Submitter who belongs to this in a fast manner, try the button **Find Submitter from SubSampleID…** A dialogue will be displayed asking for the *SubSampleID*. You can input `SS###` or `S###` where ### is the SamplingID or SubSampleID, and you will arrive at the submitter – as long as a submitter is attributed as foreign key to this special sampling.
If you press button **to Form Sets…** you will arrive at the form that is described in the next paragraph.

Sampling Sets

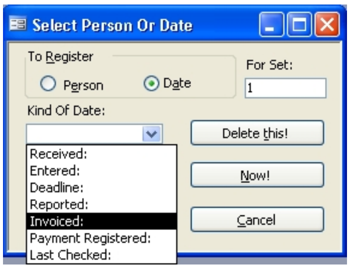
Sampling Sets or shorter *Sets* form entities that bundle several Samplings for organisational reasons: a Set comes from a person called submitter, normally all SubSamples arrive within one submission, the submitter wants one invoice for the set, and one report. So the set is one of the main handles during workflow through the laboratory. Typically in the physical world a Set is represented by a parcel with water bottles, bags of sediment... delivered to your lab. To see the according form, start LabData and press the buttons **Edit Data >> Sampling Sets**. The following form will show up:



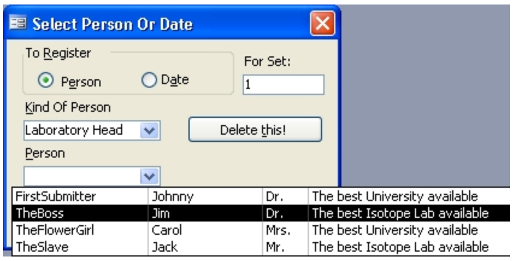
You will recognize that a set normally has a *Submitter* (here: Johnny FirstSubmitter), because otherwise you will face difficulties in reporting and billing your work. There can be more persons connected with a set like the *Officer in charge* who accepts the SubSamples in the lab (here Jack TheSlave), and the *Sampling Person* who did the field work (here Carol TheFlowerGirl), a fact that will be discussed below in detail. There are also several dates connected to a set: set 1 for instance was *Received* on march, 15 but the parcel was hanging around for five days in the lab until it was finally *Entered* into LabData. There also seems to be a latest date when the results have to be reported, called *Deadline*.
In field *Total SS* you can see that this set has a total number of 24 SubSamples and a short glance further down to the sub-form *Samplings:* will tell you that these originate from five Samplings only. Some institutions have administrational issues to be done after the SubSamples reached the institute but before they come to the lab. These institutions typically have some kind of entry code, for instance written in one of these good old paper books, and this alpha-numeric entry code can be stored in the field *Receiving Code*. Similarly you might want to store the *Order Number* that either your institution or the submitter gave to this set of SubSamples.
Since often the submitter does not really provide us with all the information needed (e.g. the sampling date for tritium…) there is a field *Missing Information* where this can be taken care of. If the work for the set is completed there will be a bunch of administrative paper. It is therefore possible and good practice to store this information in electronic form - like a pdf file or similar of the letters, tables with results, reports and so on - in the field *Comment*. An important issue for a lab is if there are any payments to be expected for the work to be done, which will be indicated by a ticked checkmark *Any Payment*. Payment is defined on the level of SubSamples and Values and will be discussed in detail in the LabData Manual.
As mentioned above, both Persons and Dates are not directly related to the sets. Instead, as can also be seen from the table topology for attribute data later on page 29, there is a n:m relation between tblSamplingSets and tblSubmitters with tblPersonsForSets in between and a similar relation with tblDatesForSets. The advantage of this construction is that you can have as many persons and dates connected to a set as you want: a submitter, somebody to send the results to, a third person doing the work in the lab, a fourth who did the field work and so on.

For dates this may be the entry date, the deadline, the date when the results were reported, the invoice was sent or when the payment arrives on the bank account. Button **Register Person or Date** allows you to attribute a date or a certain person to a set. Pressing it you will be confronted with the following form:

Let's assume you have done all the work that is needed for this set and that you want to report and invoice the results. To register that you did this just now, press the button **Now!** and select `Invoiced:` as the *Kind Of Date:* as seen below:



With this selection and the necessary acceptance pressing the **OK** button you will register the actual system date as the moment you invoiced the set. Similarly you can register a person: to define the laboratory head you should try the following: invoke this dialogue again, select ***Person*** instead of date, select `Laboratory Head` as the *Kind of Person* and `Dr. Jim TheBoss` as the lab head. This looks like the following screenshot:

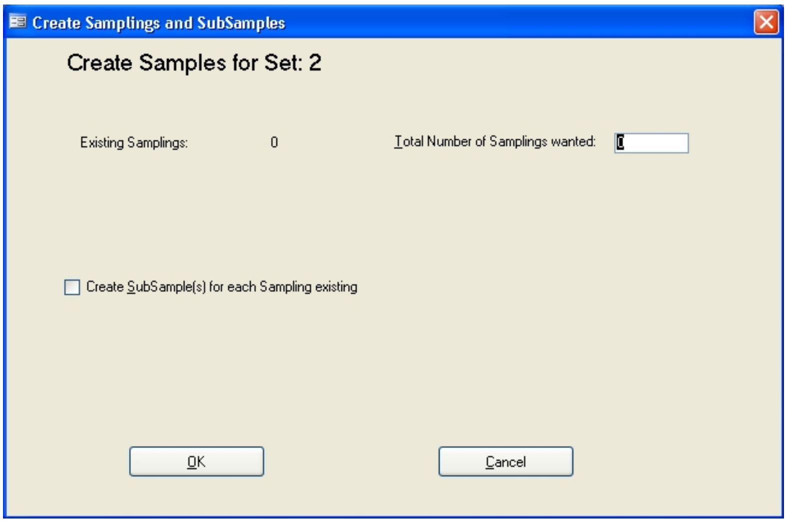


Please note that the same mechanism works for persons on form Samplings if you want to register a person for a certain Sampling.
You should also try the following four buttons: **Print Labels…** creates an Access report that can be used with a laser printer to produce stickers e.g. to label the SubSample bottles. **Print SubSample List…** similarly provides paper lists, sorted by the method with which SubSamples were created. **Print Submitter Information…** and **Print Bill…** are Access reports adapted to the work in the Leibniz Institute for Applied Geophysics (LIAG, Hannover, Laboratory for Geochronology and Isotope Hydrology). You surely will want to adapt them to the work in your laboratory. That needs some knowledge on the details of your lab that I am not able to provide (corporate identity prescriptions like a logo, address and so on), so to use this functionality you have to study the manual for MS Access and store and connect your

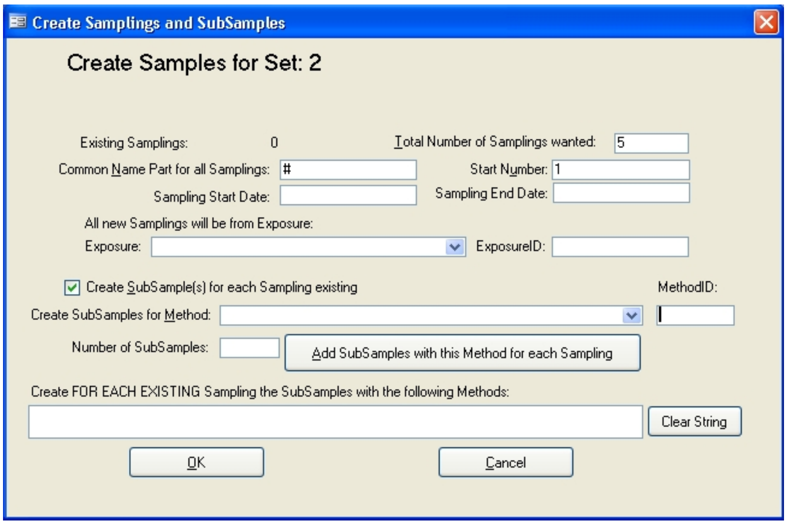
reports. The LabData Manual will explain what the constraints are for these reports and how to connect them to the functionality of these buttons.
The details of the sub-table *Pending* and of the button **Check Measurements…** are explained in the manual, since they describe functionalities beyond the scope of this tutorial. Also for the structure behind the table of *Projects* and their relation to Samplings I refer the reader to the manual. The buttons **Find all Parameters** and **Export Results** as well as all the stuff within the rectangle containing *Output Format and Method* will be of interest when we have results for a set, later in this Tutorial. But the next paragraph will describe comfortable functionalities that make life easier when SubSamples arrive in the lab.

Back from Field Work: Creating Samplings and SubSamples

Imagine you received a new set of isotope hydrological SubSamples. It consists of stable isotope, tritium and ^{14}C SubSamples for five wells. Samplings are numbered # 1 until # 5. At each well a litre of water was taken in a plastic bottle for tritium, two brown glass bottles were taken for stable isotopes, and the ^{14}C SubSample was prepared precipitating the carbonates as $SrCO_3$ and the precipitate was decanted into 1l bottles. The steps to put this information into LabData consist of creating a new Set, creating five Samplings for that Set, and creating a total of 20 SubSamples. In principle you already know how to do that with each of the forms, the *New* and *Save* buttons and so on. But working with the forms alone, this would be quite a boring job. The button **Create Samplings…** pretty much helps you to shorten that down. Create a new Set, perhaps give it a valid submitter, entry date or other related information you want to enter, and click **Create Samplings…** A dialogue opens:

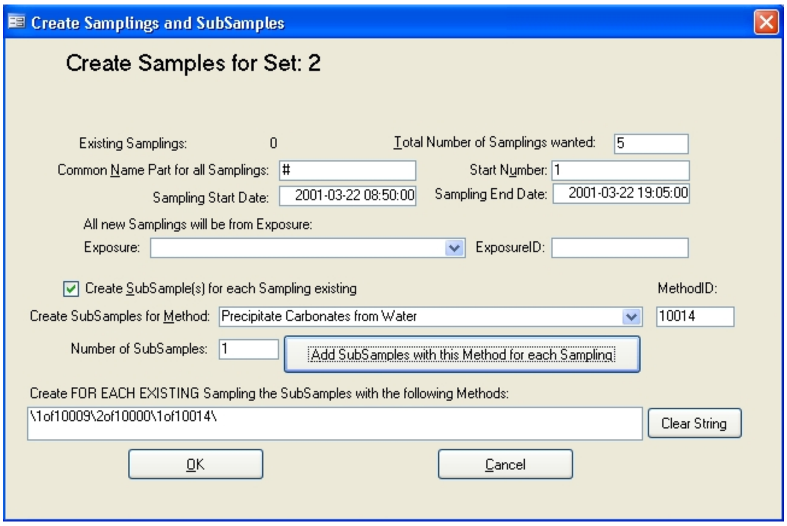


Since we have five Samplings the input of a 5 in the field *Total Number of Samplings wanted* seems appropriate. Proceed and check ***Create SubSample(s) for each Sampling existing***. You will notice that the dialogue changes and that the Common Name Part for all Samplings is "#", which actually is what we wanted. As you checked that you want to create SubSamples you received some new options:



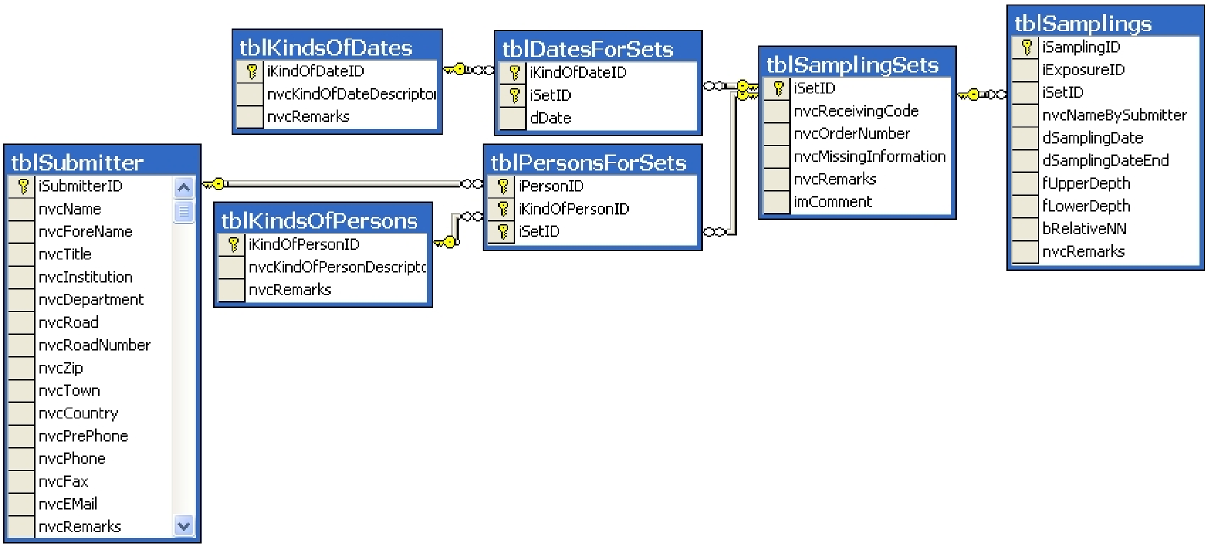
You can choose the method now for which you want to create the SubSamples for each sampling. Since we want to create a SubSample of water in a plastic bottle for each sampling, we select the method `Fill water into a Plastic Bottle` in the field *Create SubSample(s) for Method*. Method number `10009` will be shown as feedback to this selection in the field *MethodID*. Since only one such bottle exists for each sampling we write `1` in field *Number of SubSamples*. Press the button **Add SubSamples with this Method for Each Sampling** or alternatively press the **Alt&A** key. You will notice that the string `\1of10009\` shows up in the field *Create FOR EACH EXISTING Sampling the SubSamples with the following Methods*. Perhaps the principle is clear already. The next steps are to put 10000 into MethodID and 2 into Number of SubSamples and press **Alt&A**, followed by the selection of the method `Precipitate Carbonates from Water` with `1` as ***Number of SubSamples*** and again **Alt&A**.

If all this was done correctly then the final dialogue should look like the following screenshot:



Of course, knowing what to do and knowing the MethodID by heart you could have edited the string in field *Create FOR EACH EXISTING Sampling the SubSamples with the following Methods* also manually. The final step is now to press the OK button, and after a short confirmative security check a progress bar on the lower left of the main Access window will indicate how the SubSamples are created. The dialogues close and you can see the newly created Samplings in the sub form.

In the last figure I have also included two values for the *Sampling Start Date* and *Sampling End Date*. All Samplings created will have these same values. This is most probably not correct and you will want to move **to Form Samplings** and edit the Samplings created. But indicating the boundaries (the first and the last timestamp in the Set) saves a lot of editing work later in form Samplings. You might also wish to change other things for the Samplings like upper and lower depth, that can not be given in the best way with this automated input. In case the Samplings all come from the same geographical location (a time series of water from a spring for instance) and if this location was already defined as an Exposure in LabData, it is possible to indicate this exposure in the dialogue already. Exposures are discussed starting on page 29. You will also learn more about a more flexible way to create Samplings and SubSamples in the chapter *Import Excel table to Samplings* on page 37.
Since I assume that your memory lasts a bit longer than some tutorial pages, you should still be familiar with the Bachmann Diagrams as seen on page 20. The following picture shows the diagram for the relations between Sets and Samplings and the relations of tblSamplingSets to Persons and Dates:



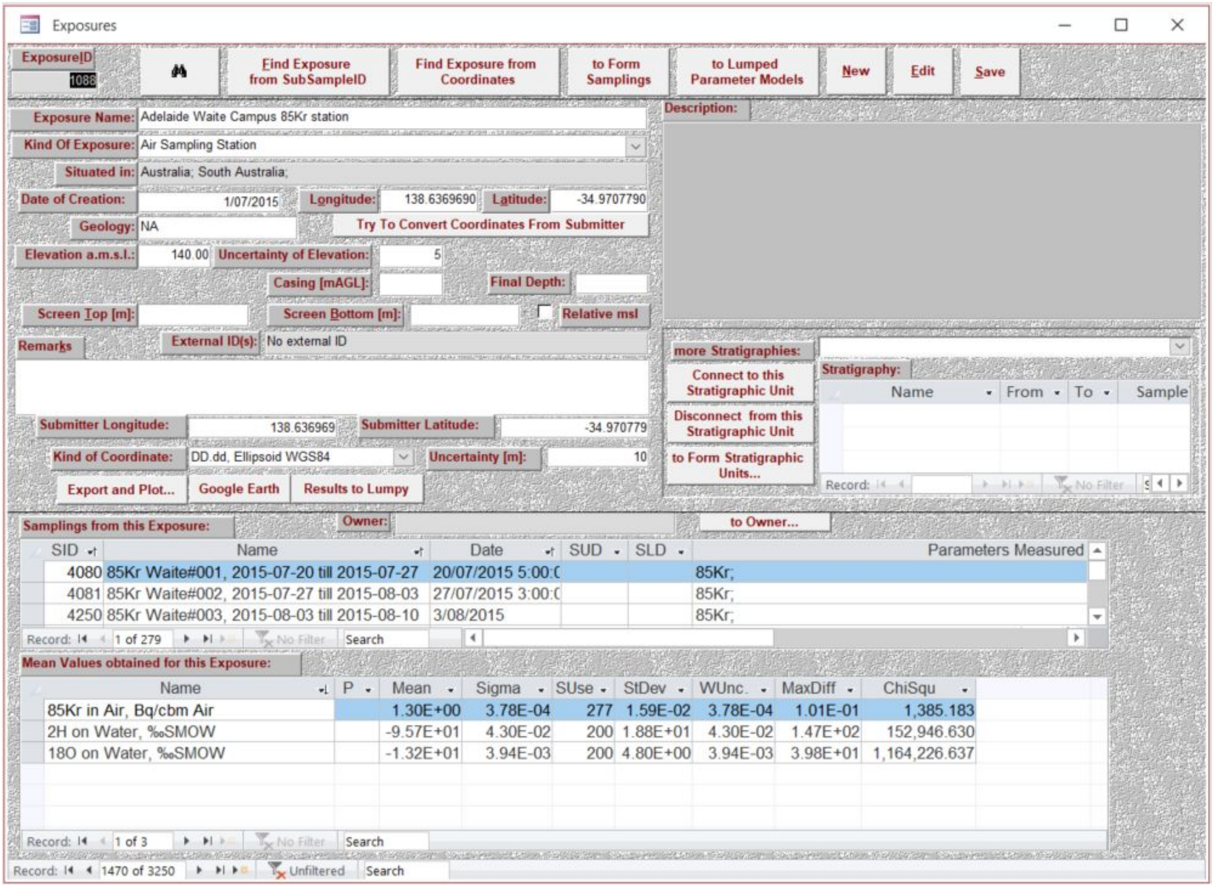
Regional Geographic Information: how to use Exposures, Areas and Stratigraphies

Besides the ease in use and the better organisation which a database can achieve for a laboratory, a database also has a high value as an archive that keeps old measurements accessible even if the operator or laboratory head was retired decades ago. For this functionality the questions asked normally are not “Give me all the results belonging to Mr. XY” or “Give me the results of all the Sets that arrived in the lab at date XY” but instead questions are of the type “Give me all tritium results from Germany between 1990 and 2000 together with their exact coordinates”. So, a strong geographical or geological component is necessary describing the place where the Sampling took place – where is it situated, if possible coordinates or even a borehole, which stratigraphic unit is it from etc. The way LabData organizes this kind of information, is what this chapter is about.

Exposures

Exposures are an abstract entity to describe the geographical place at which a Sampling happened. Any Sampling where the geographical coordinates are known should have an attributed exposure. Exposure is an abstract concept, so it is not really important if this place

with coordinates is a sediment core or a well or a borehole or an archaeological site or just a spot on planet earth where somebody passed by with a GPS system. But in order to draw the benefits of the concept for an Exposure within LabData, it is important to follow some rules concerning the coordinates.
To see the information in tblExposures, start LabData and press the buttons **Edit Data >> Exposures**. The following form will show up:

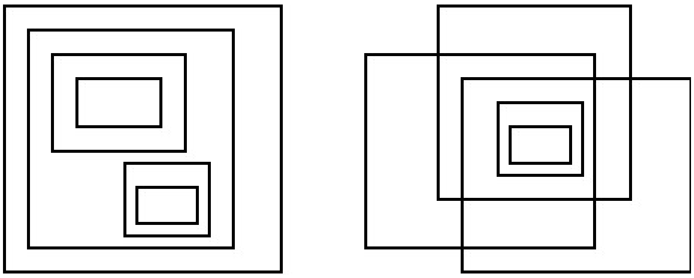


The more trivial parts of this form are the fields *Exposure Name* (an identifier that not necessarily is unique), the *Kind Of Exposure* (observation well, sediment core…), the *Date of Creation* (date of drilling the core or the completion of the well). This form allows the user to store a file describing details in field *Description*, like a photo of the core or well or a technical drawing how the well is constructed. The form also displays at a glance all the existing samplings for this exposure in a sub-form, with their Sampling Date and the measured parameters. And for all measurables it also displays the mean value over all times it was measured.
The most important fields for the geographical relations are the fields *Longitude* and *Latitude*. These are stored as floating-point numbers and are what LabData uses to identify geographical relations. To do that, LabData needs to perform comparisons like ">" and "<". Therefore, the very common UTM and Gauss-Krueger coordinate systems are not suitable for input in *Longitude* and *Latitude*: They are easily understandable rectangular systems, but because the reference longitude differs from case to case, they are not necessarily unique, and locations cannot be easily compared numerically. As long as they are related to a unique geographical datum, one can use degrees in decimal form for *Longitude* and *Latitude*. I decided to use this approach in all the LabData installations I have to supervise, and to relate them to "WGS 1984". This is readable by humans, easily converted to a point on a map,

comparisons by “>” and “<” are possible and it is globally unique if the southern and western hemispheres are given negative numbers.
Just to emphasize, it is important that all coordinates stored in *Longitude* and *Latitude* are based on the same coordinate system and the same geographical datum, because already slight differences like e.g. a change in the ellipsoid can cause shifts of the same coordinate of several hundred meters. So if for one project two different ellipsoids are used, then a map displaying the exposures can show exposure A east of exposure B when in the real world the opposite is true.
Since the original coordinates the submitter sent to the lab should be preserved for the submitter (e.g. for the final report), these are also stored. The fields for these coordinates on form Exposures are called *Submitter Longitude* and *Submitter Latitude*. Again these two fields are numeric floating-point values, but they are obviously no unique coordinate system within LabData and are not used for internal computations. Field *Kind of Coordinate* describes the submitter coordinates by their name. Entries might be for instance “UTM, Ellipsoid WGS84”or “Gauss-Krueger”. LabData is not a tool to perform coordinate transformations between the internal unique and the different submitter transformation systems! If such a tool is needed, a search on the web might be helpful: my search performed on 2004-12-29 resulted in two different free (demo) software packages that will do the job if a small number of coordinate transformations is concerned. Try for instance
http://www.eosgis.com/ESP/MAIN_FRAME.htm and
http://www.allsat.de/en/software_alltrans.htm.
Similar to Sets, the relation between Exposures and Samplings is realized by directly putting the *ExposureID* visible on the upper left of the form Exposures into the field *ExposureID* on form Samplings. Alternatively the Exposure can be selected in field *Name of Exposure* in form Samplings. Also here form Samplings provides some feedback: it displays the *ExposureID* and *Name of Exposure* after the change.
Isotope hydrologists will love the functionality that is hidden behind the button **to Lumped Parameter Models**, but this is covered in a separate manual because it is enough content for a full publication and surely beyond the scope of this tutorial.
The field *Situated in* is a field that the user cannot change, and it shows all the Areas this Exposure is situated in, so let’s have a look how Areas are defined.

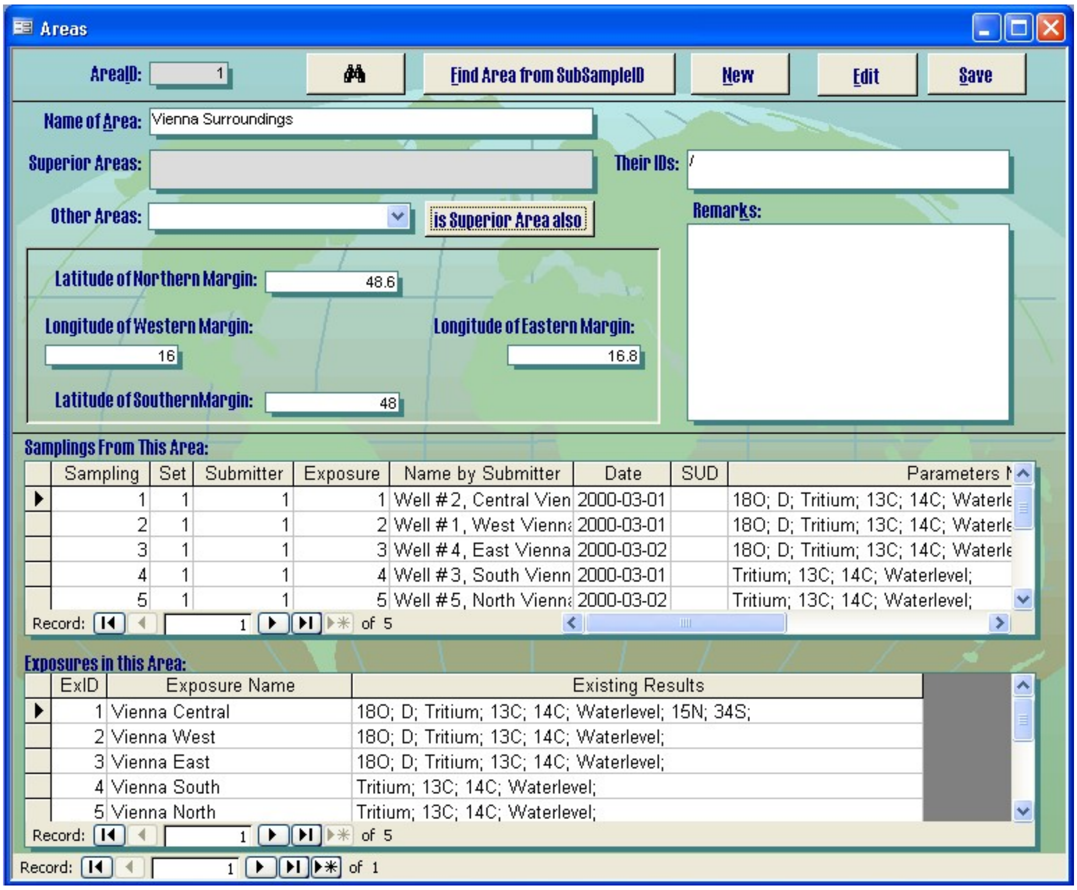
Areas

Sampling Sets are often submitted without exact geographical information about the sampling sites. Nevertheless, “fuzzy” geographical information can exist that might be of future interest (e.g., “SubSamples from the island Mallorca”). On the other hand, when results have to be drawn from the database on the basis of geographical information, in many cases the query consists of something like “give me all results for exposures situated on this map”. Both demands lead to the definition of areas as a rectangle defined by the coordinates of the southwest and northeast corner in which all the sampling points in question are situated.
Following this thought, a network of areas is obtained with relationships that can be numerically defined: There are areas that completely overlap each other, building a tree-like linkage structure (like Chicago – Illinois – USA), and there are areas that are situated completely in several super ordinate areas that do not overlap completely (for example, Green River – Wyoming – Utah):



LabData would find those areas that are completely within greater areas and automatically via SQL triggers and procedures. LabData then maintains the m:n relationships between samplings linked manually to the areas, connecting them to all super ordinate areas. This is necessary to find all data in a query for a super ordinate area: “samplings from Mallorca” could otherwise not be found in a search for “samplings from Europe”.
Things are of course more straightforward for Samplings being related to an Exposure: in this case point coordinate information exists and the triggers and procedures in LabData will automatically connect this sampling to the area via “<” and “>” coordinate relations.

To display the form to create and edit areas in LabData, start LabData and press the buttons **Edit Data** >> **Areas**. The following form will show up:



The fields that define an area are the *AreaID* (which as usual is defined by LabData whenever you create a new area), the *Name of Area*, the *Remarks* and the corner (or boundary) coordinates *Latitude of Northern Margin*, *Longitude of Eastern Margin*, *Latitude of Southern Margin* and *Longitude of Western Margin*. All these fields are straightforward in their meaning.

Define a new area by pressing the **New** button and typing in the name and values as in the figure above. As soon as you press the **Save** button (or use the **PgUp>>PgDn** combination) to submit the new data, LabData will show you that the trigger functionality on the server side found five samplings in this area. This is because the Exposures have coordinates and because the SQL procedures find the exposures situated in an area. They then establish the m:n relationships between areas and exposures and afterwards the n:m relationships between areas and samplings. The reason why LabData works with the relations instead of directly with the coordinates is that this saves computing time for a query: no further coordinate computations are necessary but only a determination of links between tables, for which the database software is optimized.

To test the functionality without exposures try the following: Create a new Area called `Schoenbrunn` with the coordinates *Latitude of Northern Margin*=`48.2`, *Longitude of Eastern Margin*=`16.35`, *Latitude of Southern Margin*=`48.18` and *Longitude of Western Margin*=`16.33`. Create another Area called `Austria`, with coordinates *Latitude of Northern Margin*=`49.1`, *Longitude of Eastern Margin*=`17.2`, *Latitude of Southern Margin*=`46.3` and *Longitude of Western Margin*=`9.4`. Save both areas. You will notice that `Schoenbrunn` is situated now both in `Austria` and in `Vienna Surroundings` – as it should be.

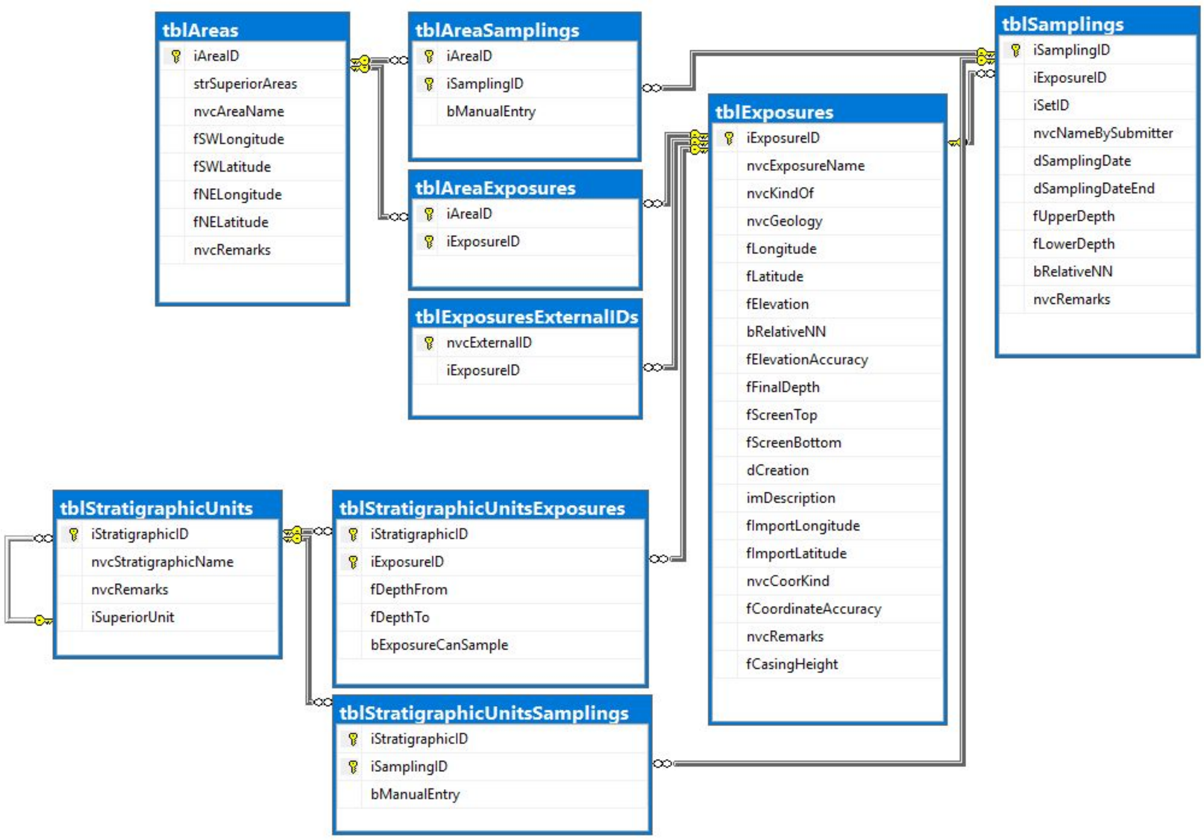
Now open the form Samplings, create a Sampling with the Name `A Sampling from Schoenbrunn` and relate this sampling to the area Schoenbrunn. To do this you select (still on form Samplings) `Schoenbrunn` from the field *more Areas*, and press the button **Connect Sampling to this Area**. After pressing **OK** on a short MessageBox you will notice that this sampling now is attributed to `Schoenbrunn`, but also to `Vienna Surroundings` and to `Austria`, since LabData noticed that the latter two fully overlap the area `Schoenbrunn`. Reopening form Areas will tell you that your new sampling is shown on the sub form *Samplings From This Area* for `Schoenbrunn` (which is the easy part) but it is also shown in `Vienna Surroundings` and in `Austria` although you never gave coordinates to this sampling and although you even do not know the exact origin. Isn't that nice?

Stratigraphic Units

Especially in groundwater, but also in sediment studies it is often useful to classify data by the stratigraphic unit they come from. The problem is similar to the relations between Areas, Exposures and Samplings. An Exposure can create samples from several stratigraphic units – just think of a sediment core that cuts through several layers. And of course, a stratigraphic unit is an extended object through which many sediment cores can be drilled. Or a stratigraphic unit could correspond to an aquifer in a multi-layer aquifer system, and there are many wells in this aquifer. The similarity to Areas extends in that Stratigraphic units can be hierarchical: The Paleogene can be subdivided into Paleocene, Eocene, Oligocene, Miocene and Pliocene. And perhaps we do not know from which of these layers come and just attribute it to the Tertiary, or we may know that it comes from the Oligocene and then we want to receive the results also if we start a query asking for all samples from the Tertiary. So, one stratigraphic unit can have many sub-units (but one sub-unit has only one superior).

To make these relations usable, LabData has three tables in complete analogy to the Area-Exposure-Sampling Trio. And it also has the necessary triggers and procedures to maintain these relationships. Table tblStratigraphicUnits provides the names and IDs for any stratigraphic unit. Table tblStratigraphicUnitsExposures on one hand creates the m:n relationship between exposures and stratigraphic units. On the other hand it is also like a stratigraphy table for every exposure. You can define for each exposure which stratigraphic

unit is found here from which upper to which lower depth. If the exposure is a drill core or groundwater well, then this table gives the geologic context. If it is a well that is screened, then only some of these stratigraphic units can be sampled for groundwater and this is indicated in a flag (bExposureCanSample). On form Exposures you have buttons to connect a stratigraphic unit to the exposure and if you press it, a dialogue will ask you from where to where this unit goes in the exposure and whether it can be sampled or not. A core could be sampled everywhere, a well with casing only in the screened interval. If you do this, triggers will take care that all existing samples from this well will accordingly be attributed to the stratigraphic unit. Similarly, if you get a set of samples from a core and the stratigraphic table for this core is existing already, then importing the samples with the correct depth (fUpperDepth and fLowerDepth) will sort them to the according stratigraphic units.
The following picture illustrates the relations between *tblAreas*, *tblExposures*, *tblStratigraphicUnits* and *tblSamplings* for those who want to understand some details how LabData works.



Data Import

The import from Excel tables into LabData follows some general rules. The import algorithm first searches a ***primary keyword*** in the Excel table, something like `ExposureName`, `NameBySubmitter` or `ValueForProcedure`. When the cell with this keyword is found, the row in which this cell is situated is defined as the header row for import. The algorithm then searches other ***keywords*** that represent the headers for other table content, like `ScreenDepth` or `SamplingDate` in this same row. After finding them, the algorithm will import all rows between the header row and including a last row the user indicates, and will try to import the cell content into the appropriate table (*tblExposures, tblSamplings,* *tblSubSamples*). As a rule of thump the keywords to search for in the Excel table are the column headers of the appropriate tables in LabData without the type identifiers, written as one word. So instead of *nvcNameBySubmitter* it is `NameBySubmitter`, instead of *dSamplingDate* it is `SamplingDate` and so on.

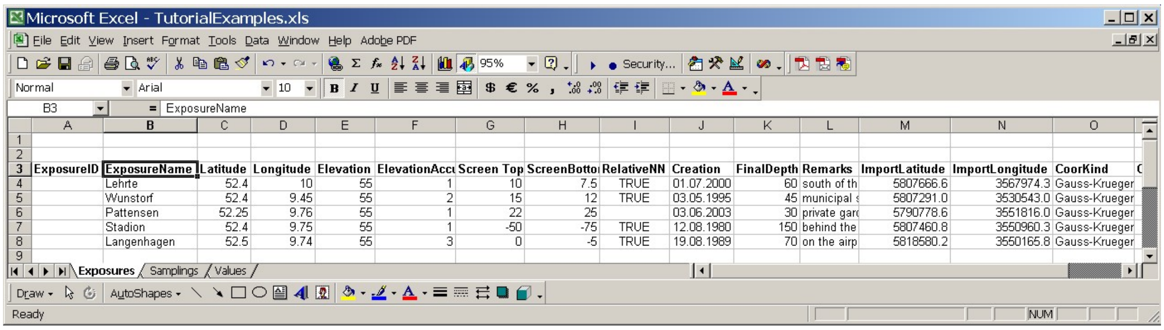
If the user wants so (which is recommended), the algorithm will write back the new ID created during this import (the *ExposureID*, *SamplingID* or *ValueID*) into a column of the Excel table that the user has to specify. For every row where the import fails, a message box will be displayed and the ID written back will remain empty. So import failures can be detected easily. In case the ID already exists, the import algorithm will perform an update of the information in the database. The following two examples show where this update mechanism is very convenient: 1) when a set of new wells is drilled and does not yet have coordinates, but already a first set of SubSamples exists, you will probably already start with the analyses in the lab. Whenever you get the table with the final well descriptions, just re-import it with the appropriate *ExposureID* and the correct coordinates will be placed in LabData (and no new exposures will be created). 2) When you have post-processing mechanisms before an import to LabData and have to repeat this post-processing with different parameters after import to LabData, then it is important to replace the former values. Many stable isotope labs have their own spreadsheets for linearity and drift correction and for renormalization to the SMOW/SLAP scale. Maybe you learn later that you have to use a different kind of drift correction for a certain measurement day. Or maybe you had a type mismatch for a SubSampleID in this spreadsheet. If the column ValueID exists for these imported values (and only then!) any re-import will not create duplicates, but make sure that the former values are corrected. In a more abstract formulation you create a foreign key relation between the LabData table and your spreadsheet table by copying the primary key from LabData.

Please note for all imports: in case the Excel workbook was closed or hidden during import, it will be hidden if opened next time after import. The Excel command **Windows>>Unhide** will change this state if you open the workbook the next time after import. The reason why the workbooks are hidden is unclear to me. My Visual Basic code just does the necessary communication using the COM (component object model) functionality that is distributed with Excel. Obviously COM notices that the workbook is not visible during import and keeps this attribute during saving of the workbook. Sorry for that and a glass of wine to the first who can explain me how I can change it…

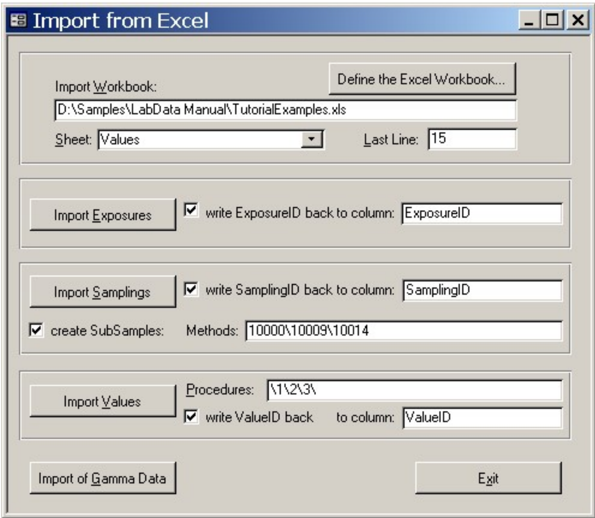
During the following steps we will import information on exposures, create samplings and SubSamples and import some measurement values. The information comes from the file `TutorialImportExamples.xls` that is distributed on the installation CD.

Import of an Excel table to Exposures

The keyword for the import of exposure information is `ExposureName`. The other keywords are the headers of *tblExposure* without the type specification. So instead of *nvcKindOf* it is `KindOf` ("nvc" stands for "new variable character" and describes unicode strings in a database), instead of *fLongitude* and *fLatitude* it is `Longitude` and `Latitude` ("f" stands for "float"). An example for a valid import table follows, which is taken from the sheet `Exposures` in file `TutorialImportExamples.xls`:



Please note that the writing of these headers is strict and case sensitive. So with `Screen Top` or `screentop` the information will not be found. But the order of the columns is not strict: if `Elevation` stands in the first, third or 100th column, it will be found as long as it is written in the same row as `ExposureName`. The import into the tables *tblExposures*, *tblSamplings* and *tblValues*, including the creation of SubSamples is done with the dialogue form *Import from Excel* that shows up if you press the button **Import Data** on the start screen:



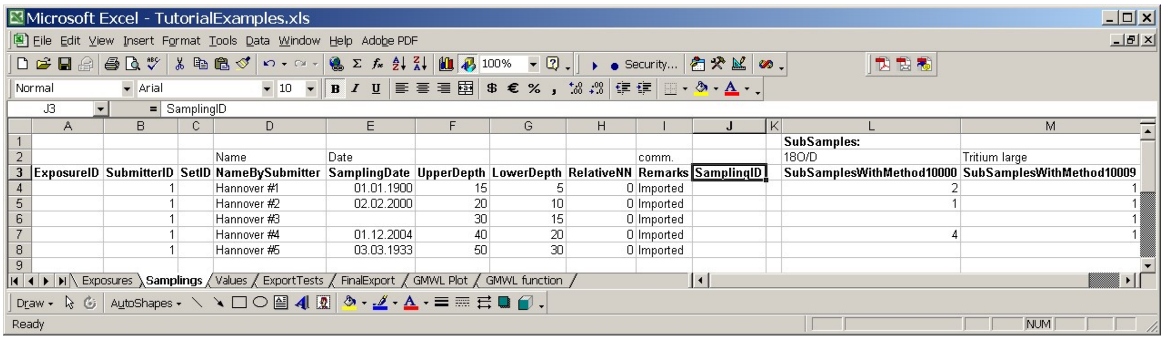
Open this dialogue, press **Define the Excel Workbook** and select the Excel workbook `TutorialImportExamples.xls` from whichever directory you copied it to. You can select then the appropriate sheet in this workbook: for import of exposure information in this tutorial and file `TutorialImportExamples.xls` it would be `Exposures`. For this example the field *Last Line* can be `10`. Since our example workbook has the appropriate column, make sure that *write ExposureID back* is selected and that the entry in *to column* is `ExposureID`. Then press the button **Import Exposures**. The progress bar at the bottom of the Access main window will indicate import progress. Look into the form Exposures and you will find five new wells in the vicinity of Hannover, Germany. Note that the fields *Screen Top [m]* are empty for all these new wells. This is because the Excel table had `Screen Top` in the header row instead of `ScreenTop`. Correct this type mismatch and if you decided to store the ID, you can repeat the import and the information will be found and imported correctly. Just try it…

The following table summarises the necessary header keywords for the import, the headers in *tblExposure* and the field in the form Exposures that belong together:

Header for Import	in tblExposures	On form Exposures	Comments
(user defined)	iExposureID	ExposureID	Will be defined by LabData during import, column where to write back is user defined
ExposureName	nvcExposureName	Exposure Name	Self-explaining
KindOf	nvcKindOf	Kind of Exposure	Well, Sediment core…
Longitude	fLongitude	Longitude	Geogr. Coordinates for Area search
Latitude	fLatitude	Latitude	As Longitude
Elevation	fElevation	Elevation a.m.s.l.	Measured elevation of ground surface
ElevationAccuracy	fElevationAccuracy	Uncertainty of Elevation	Uncertainty of elevation in m
FinalDepth	fFinalDepth	Final Depth	Depth to bottom of well or core
ScreenTop	fScreenTop	Screen Top [m]	Distance of ground surface to top of screen
ScreenBottom	fScreenBottom	Screen Bottom [m]	
RelativeNN	bRelativeNN	Relative m.s.l.	Valid only for Screen Top and Screen Bottom
Creation	dCreation	Date of Creation	Start/finish if drilling…
ExposureRemarks	nvcRemarks	Remarks	Anything important for this exposure…
	imPicture	Picture	A graphic file like jpg or pdf, can not be imported from Excel!
ImportLongitude	fImportLongitude	Submitter Longitude	The coordinates the submitter indicated,
ImportLatitude	fImportLatitude	Submitter Latitude	can be different from the LabData coordinate system.
CoorKind	nvcCoorKind	Kind of Coordinate	Describes the submitter coordinates
CoordinateAccuracy	fCoordinateAccuracy	Uncertainty [m]	Uncertainty of coordinates in m
Geology	nvcGeology	Geology	A string describing the geology (e.g. of the formation the well taps)

Import of an Excel table to Samplings

The primary keyword for the import of sampling information is `NameBySubmitter`. The other keywords are the headers of *tblSamplings,* as in case of Exposures without the type specification. An example for a valid import table is the sheet `Samplings` in file `TutorialImportExamples.xls`, of which a screenshot follows:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1												SubSamples:	
2				Name	Date				comm.			18O/D	Tritium large
3	ExposureID	SubmitterID	SetID	NameBySubmitter	SamplingDate	UpperDepth	LowerDepth	RelativeNN	Remarks	SamplingID		SubSamplesWithMethod10000	SubSamplesWithMethod10009
4		1		Hannover #1	01.01.1900	15	5	0	Imported			2	1
5		1		Hannover #2	02.02.2000	20	10	0	Imported			1	1
6		1		Hannover #3		30	15	0	Imported				1
7		1		Hannover #4	01.12.2004	40	20	0	Imported			4	1
8		1		Hannover #5	03.03.1933	50	30	0	Imported				
9													

The following table summarises the necessary header keywords for the import, the headers in *tblSamplings* and the field in the form Samplings that belong together:

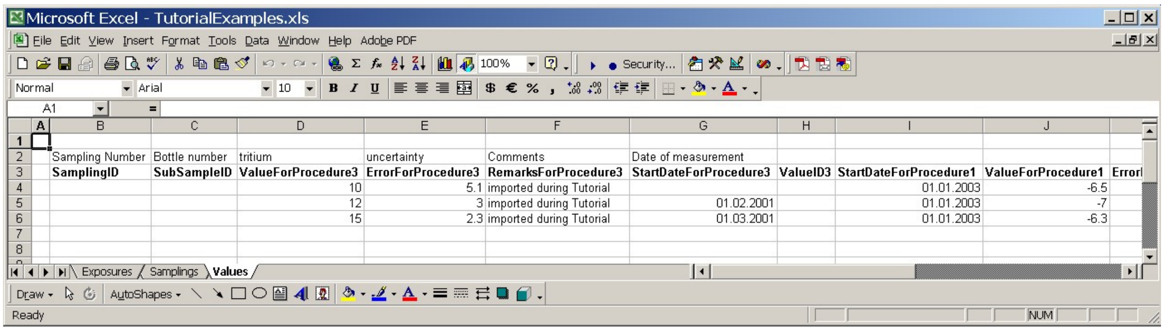
Header for Import	**in tblSamplings**	**On form Samplings**	**Comments**
(user defined)	iSamplingID	SamplingID	Will be defined by LabData during import, column where to write back is user defined
ExposureID	iExposureID	ExposureID	Foreign key, to be searched in tblExposures
SetID	iSetID	SetID	Foreign key, to be searched in tblSamplingSets
NameBySubmitter	nvcNameBySubmitter	Name by Submitter	The (hopefully unique) name the submitter gave to the sampling.
SamplingDate	dSamplingDate	Sampling Date (Start)	Date and time of sampling, with resolution of seconds
SamplingDateEnd	dSamplingDateEnd	Sampling End	Date and time for end of sampling
ElevationAccuracy	fElevationAccuracy	Uncertainty of Height	Uncertainty of elevation in m
UpperDepth	fUpperDepth	Upper Depth	Screen top for wells, distance from top of core for sediment cores, unit normally [m].
LowerDepth	fLowerDepth	Lower Depth	See above
RelativeNN	bRelativeNN	relative m.s.l.	States the base point of the two above
SamplingRemarks	nvcRemarks	Remarks	
SubSamplesWithMethod###			The content is interpreted as number and the according number of SubSamples with procedure ### is created
SubSamplesWithMethod###Remarks			The content is interpreted as text and added in the Remarks field of the SubSample
SubSamplesWithMethod###DateTime			The content is interpreted as date and used as entry date for the SS

SubSampleID###	If this column is given, the created SubSampleIDs are written to this field, as string where the Ids are separated by " ;"

During the import of Samplings, many SubSamples of any kind can be created automatically for any Sampling. To achieve this, make sure the header row contains a cell with the string `SubSamplesWithMethod###` where ### stands for the method with which the SubSample is to be created. In the rows below define the number of SubSamples to be created with this method. If the cell is empty or contains a non-numeric value, no SubSample will be created. If you repeat (or better "update") the import and the SubSamples exist already, no further SubSamples will be created, if the *SamplingID* was written back to the Excel sheet. In the screenshot above, four SubSamples for stable Isotopes would be created for the sampling called "Hannover #4", but none for sampling "Hannover #3". Besides these entries in the Excel table you have to make sure that on form *Import From Excel* the checkbox *create SubSamples* is marked. And you have to indicate the methods for which you want to create SubSamples in the text field *Methods*, as indicated on the screenshot on page 36. Note that the numbers of several methods are separated using the backslash (\).
In case you want to create SubSamples you can get a feedback on the SubSampleIDs created into the import Excel table. To use this feature, provide a column with the header `SubSampleID###` in the Excel table where `###` corresponds to the ID of a valid method for which you want to create SubSamples. After creating the SSIDs LabData will write them to this column. In case you want more than one SubSample to be created, LabData will concatenate the IDs into a string separated by " ;".
Of course the submitter can provide you a table where the information about the SubSamples is stored in. But she/he often does not know the according `SubmitterID` and obviously normally does not know the `SetID`. Therefore you create these IDs when you receive the table. The ID for Set, Exposure and Submitter have to exist in LabData prior to import, otherwise an error occurs. So create a new Set before you import the Samplings in this sheet, and write the number of this set into the column of the Excel sheet. You notice that a little effort in editing the table provided by the submitter is normally necessary. The following hint might help in some cases: If the necessary information about the exposure is given in the same excel table as the information about the samplings, and if no two samplings come from the same exposure, you can just add a column, give it a header `ExposureID` and first import the Exposures and then the samplings from the same table. LabData will write back the `ExposureID` in the first step and import it then correctly during the second step. Alternatively, for the screenshots above it is possible to copy and paste the ExposureID from sheet Exposures, which was written back to the Excel workbook in the tutorial step before. You should now easily be able to import the Samplings in sheet `Samplings` in file `TutorialImportExamples.xls` and create the SubSamples indicated in there. To do so, select `Samplings` in field *Sheet* and click **Import Samplings** on the form *Import From Excel* (figure page 36). In the even more convenient case that every Sampling has only one SubSample and that the submitter provides values to be imported also, you can use the SubSampleIDs created during import of the Samplings for the import of values. Just think about it and check it out!

Import Excel table to Values

The primary keyword for the import of values is `ValueForProcedure###` where "###" denotes a valid ProcedureID. Analogue to the import of Exposures and Samplings the other keywords are the headers of *tblValues* without the type specification. An example for a valid import table follows, which is a screenshot of the sheet `Values` in file `TutorialImportExamples.xls`:



	A	B	C	D	E	F	G	H	I	J	
1											
2		Sampling Number	Bottle number	tritium	uncertainty	Comments	Date of measurement				
3		SamplingID	SubSampleID	ValueForProcedure3	ErrorForProcedure3	RemarksForProcedure3	StartDateForProcedure3	ValueID3	StartDateForProcedure1	ValueForProcedure1	Error
4				10	5.1	imported during Tutorial			01.01.2003	-6.5	
5				12	3	imported during Tutorial	01.02.2001		01.01.2003	-7	
6				15	2.3	imported during Tutorial	01.03.2001		01.01.2003	-6.3	

You will notice in the screenshot that in contrast to the import routines above for samplings and exposures the headers for the values always contain the substring `ForProcedure###` where ### is a valid *ProcedureID*. The reason for this is that the import can be made much more flexible with this approach, because many values for a pair of *SamplingID* and *SubSampleID* can be imported at the same time this way. If you open the file `TutorialImportExamples.xls` you will notice that there are more columns than shown in the screenshot, and they correspond to other procedures (for the parameters D and ^{18}O). All these values will be imported in one run during the import routine. Whenever a combination of `SubSampleID` and `SamplingID` is not valid or whenever a cell under `ValueForProcedure###` is not numeric, this value cannot be imported and an error message occurs.

If you imported the samplings during the last step of the tutorial you should be able to find three valid SubSamples from this new set and write their SID/SSID into the excel sheet. Then you can import the values selecting `Values` in field *Sheet* and pressing the button **Import Values** on form *Import From Excel*. Note that for every procedure you want values to be imported you have to write the number of the procedure into field *Procedures* of form *Import from Excel*, separated by backslash (\). Also note that the column header to write back the ValueID is not really the full column header but only part of it: the algorithm will add the procedure number to the string (in case of the screenshot of page 36, the correct headers will be `ValueID1`, `ValueID2` and `ValueID3`).

The following table summarises the necessary header keywords for the import, the headers in *tblValues* and the field in the form Values that belong together:

Header for Import	in tblValues	On form Values	Comments
(user defined plus ###)	iValueID	ValueID	Will be defined by LabData during import, no column to write back the ID.
(given as substring ### in any imported header)	iProcedureID	Procedure	Foreign key, to be searched in tblProcedures. No own column exists.
SamplingID	iSamplingID	SamplingID	Foreign key, to be searched in tblSamplings
SubSampleID	iSubSampleID	SubSampleID	Foreign key, to be searched in tblSubSamples

StartDateForProcedure###	dStartDate	Start Date	Date and time of start of measurement, with resolution of seconds
PreFixForProcedure###	cPreFix		Indicates below detection limit (“<”) or out of range (“>”)
ValueForProcedure###	fValue	Value	The measured value
ErrorForProcedure###	fError	±	The measured uncertainty
LockedForProcedure###	bLocked	Locked	If not to be used for mean values
RemarksForProcedure###	nvcRemarks	Remarks	

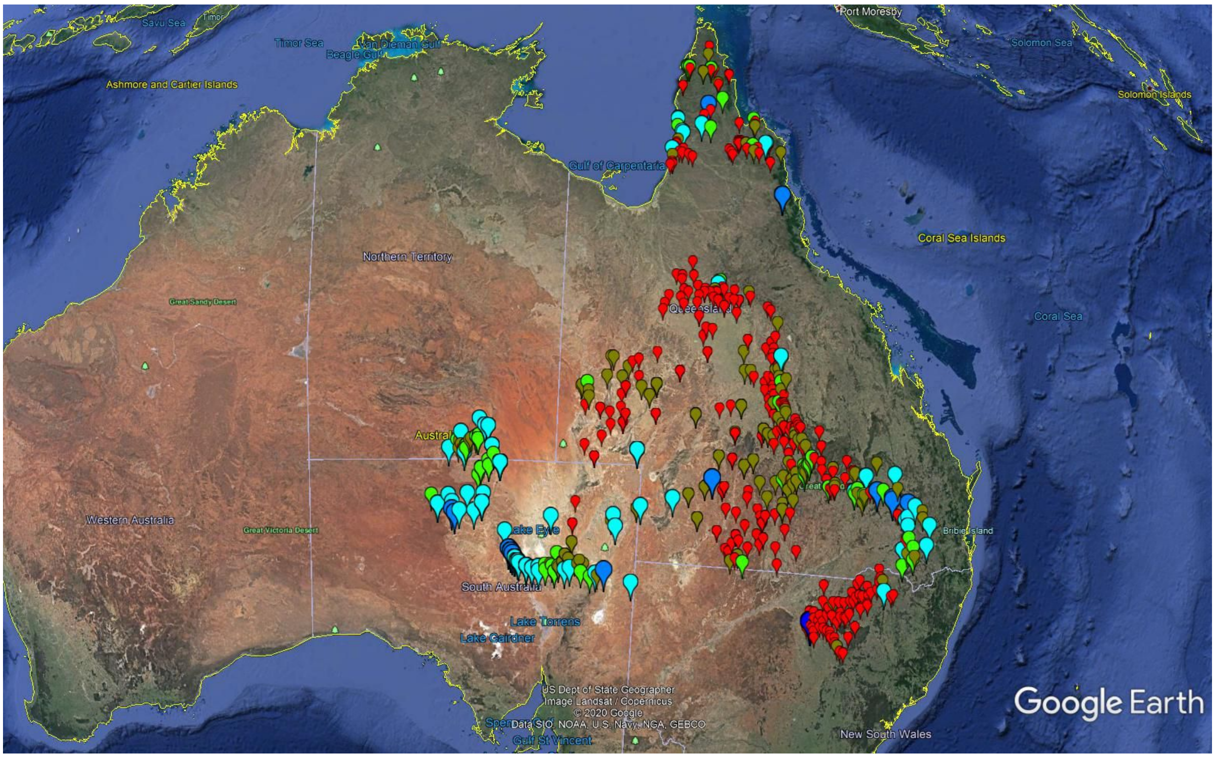
Data Export

A database would not be very useful, if you could only insert data and were not be able to get them out somehow. The GUI of LabData already gives you a lot of feedback on the state of your samplings and their organisation, if you are willing and able to use the functionality. Besides that, it is of course always possible to use the built-in functionality to create queries that is inherited from Microsoft Access itself. This is not very difficult and does not need an understanding of the SQL programming language since there is an easy-to use graphical query editor in Access. Try it using the `Create` ribbon, move to the `Queries` and use the query wizard. Many of the things you can do here is self explaining, however, according to Murphy’s law, the stuff you will need most urgently probably is not. This tutorial will not describe the details how to create a query, but you can find that in every textbook on Microsoft Access©. This functionality enables you to query really any information that is stored in LabData at all.

Existing Access tables and queries can also be used in Excel directly: to do so, you have to open Excel and use the menu command **Data>>Get external Data>>New Database Query**. This will show you all existing ODBC connections and allows also connecting to LabDataGUI selecting `Microsoft Access Database` and `LabDataGUI.mdb`. Here you can select any existing query, but you can also create your own, because you are in a query wizard similar to the one in Access itself.

Export to Google Earth

On each of the forms Areas, Exposures, Persons, Projects, Sets and Stratigraphic Units you will find a button **Google Earth**. If you click this button (and Google Earth is installed) the positions of all Samplings that have Exposures with coordinates will be transferred to Google Earth. The forms also allow you to select a Measurable and a Colour Map that codes the measured values into a colour and symbol size. If you do select these two fields as well, Google Earth will display the results as map. An example is below, which displays the chloride concentrations in groundwater in the Great Artesian Basin, the data is from Radke et al. (2000).

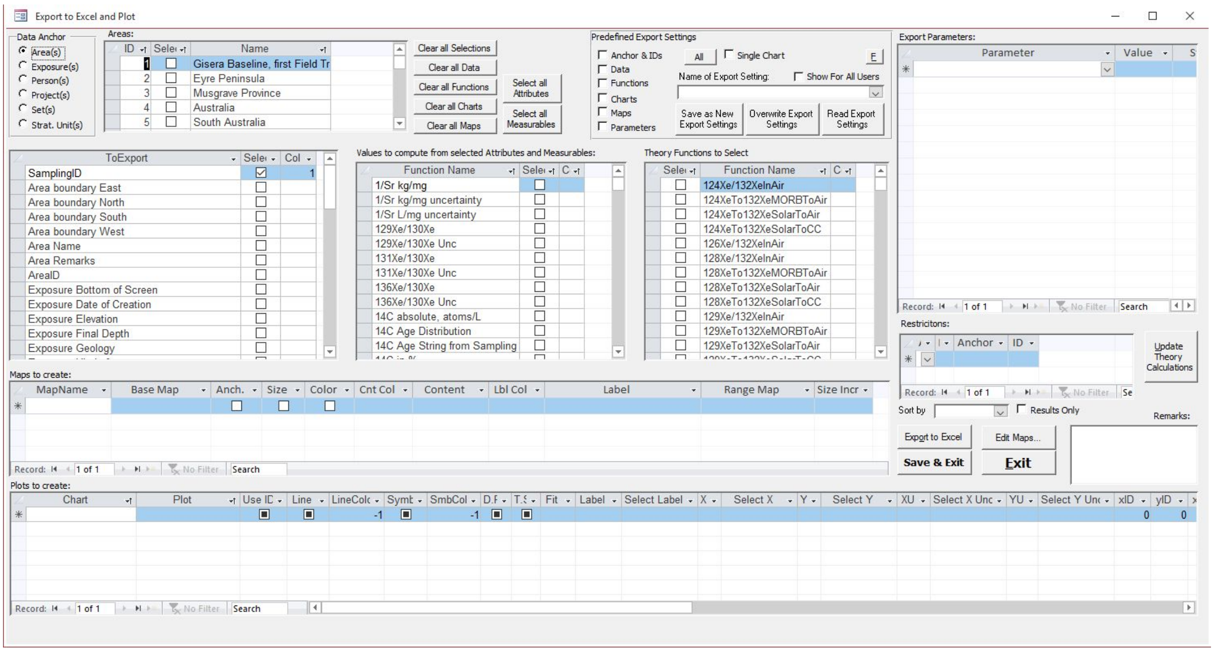


This is probably the quickest way to get data into a display. However, I have not found a way yet to create a legend for the values in Google Earth. Also, the degree of numerical analysis possible this way is rather limited. LabData therefore has much more powerful tools to display and analyse data.

Export to Spreadsheets

Most users do not need all kind and combination of information stored in LabData in their daily life. And many are not willing to learn how to create SQL queries – even not with the easy to use Access graphical interface. For those users who wish to stay in the pre-defined graphical user interface of LabData, a form was implemented to export final data to an Excel sheet. The problem creating such a form is that it is content dependent. The programmer knows all attributes of LabData and can query all the Measurables that a user stored in LabData. But it is never possible to create a graphical user interface for all possibilities of results to be computed from these attributes and measurables. Therefore, this export form for principal reasons must be incomplete concerning the ability to query any possible information.

Start LabData and press the button **Export Data**. A form will show up, headed *Export to Excel and Plot*:



This looks complex but don't feel overwhelmed, it is quite easy to use. As usual, we will work through and explain this form roughly from top left to bottom right.

In the upper left you see a block called *Data Anchor*, where you may select the top-level handle with which you want to grab the data. You can choose one of the six tables of attribute data discussed on pages 23-34. Since we started this form from the general menu, *Area(s)* is selected here, which is the default. Next to the right you see the list *Areas* where you can select those area you wish to export data from. Just mark the field *Select* if you wish to have data from an Area exported to Excel. If you change the data anchor, this table will change as well. So e.g. if you select *Person(s)* in the block *Data Anchor*, the table will change and be filled with the names and forenames of all available persons in LabData.

The form *Export to Excel and Plot* is reachable from the forms *Areas*, *Exposures*, *Persons*, *Projects, Sets* and *StratigraphicUnits* as well, clicking on the button **Export and Plot…** available on each of these forms. Whenever you navigate to the form from one of the other anchor points, this anchor will be selected, and the table be filled accordingly. Since LabData assumes that in this case you will wish to export data from this anchor, the according ID of this anchor will also be selected. Just try it by clicking button **Export and Plot…** from one of these forms.

The minimum information you will obtain as output during export with a selected anchor is the SamplingID. Try that using *Set* `1` as the only anchor. But, of course, you may wish to export more data. For instance directly after editing of importing Samplings and SubSamples for a new Set – and before anything was measured yet – you may wish to create a user defined Excel list with the date of sampling, the depths and all available Exposure information. These are called attribute data.

Attribute Data

Attribute data are those data that are not produced (measured) in your lab, but that describe the origin of a sampling or SubSample. Examples are the *Sampling Date*, the *Name by Submitter*, the name of the exposure and so on. So, let's imagine you want to have in your list the *Sampling Date*, *Upper Depth* and the *Name by Submitter*, together with the *Latitude* and *Longitude* of the Exposure. These are selected in the table directly below the anchor blocks by just ticking the appropriate *Select* box.

Select the following attributes using Set 1 as anchor: `Sampling Name by Submitter`, `Sampling Date`, `Sampling Upper Depth`, `Exposure Latitude` and `Exposure Longitude`. For some of these attributes you will have to scroll down in the table. You may notice that after each selection a number shows up in the field *Col* of this table. This number is the column in which the output is placed in the Excel table, so *Col*=`1` corresponds to column A in Excel, *Col*=`3` to column C in Excel and so on. Perhaps you also noticed that that the name of the attribute table always comes in front of the attribute: *Sampling Name by Submitter* is from table Samplings, *Exposure Latitude* is in table Exposures and so on. This makes the selections easier because you more easily find the attributes. Press button **Export to Excel** on the very right of the form and see what happens. LabData creates a new Excel workbook containing a sheet *Set1*. For each selected attribute an own column is created with the according header. This column will be at the position indicated by the number in Col, which can be easily changed by the user. In case there are gaps in this numbering, LabData creates empty columns with Column# as header and # the corresponding number. You may wish to verify this changing the number of *Sampling Upper Depth* (which should have number 4 according to the sequence above) to 8 and export again. Now LabData creates two empty columns, headed with **Column4** and **Column7**. Also note that all cells for which LabData has no information will be empty, for instance for our selection *Sampling Upper Depth* is empty.
In most cases in a laboratory, however, you will not only wish to export the attribute data that you ideally knew before starting your work, but you will export the measured data as well.

Measurement Results

The table below the data anchor block contains not only attribute data but also measured data. You may have noticed that this content of the table is renewed whenever you change the anchor selection and whenever you change the selection of anchor IDs. This tells you that this table is filled in an intelligent way: LabData shows you only those Measurables that were measured for your selection of anchor IDs. Also, LabData will give you the possibility to select the uncertainty for most measured values. Please remember: a measurable is the combination of Parameter (like ^{18}O, ^{2}H or Tritium), Unit (like VSMOW, VPDB or ppm) and Material (like Water, Nitrate, carbonate…). ^{18}O on Water therefore must be a different measurable than ^{18}O on NO_3 (sounds reasonable, isn't it?). For our case we want to select the results for tritium, the stable water isotopes D and ^{18}O, and the results for ^{13}C and ^{14}C. So please tic the *Select* for `Tritium on Water`, `18O on Water`, `2H on Water`, `13C on TDIC_H2O, %oVPDB`, `14C, %`. Submit the query clicking **Export to Excel** (or pressing **Alt&O**). As expected, you will find the measured results also exported in the Excel sheet. Keep this Excel sheet open for the next step.
As soon as you have measured data it makes sense to test what the effect of the tic mark *Results Only* on the lower right of the form might be. Tic it and repeat the export by clicking **Export to Excel** again. Comparing the two Excel sheets you will notice that this time you have less rows exported. LabData in this case exported those rows only were at least one of the selected Measurables was actually measured. In case no results exist (as for SID xy), the row is omitted.
Up to now we only demonstrated that LabData can do what everybody expects a database to do: to store data and to show and export data. But LabData also has post processing features and can compute (sometimes with quite complicated algorithms) secondary results from actually measured data.

Secondary Functions

In a laboratory for geochronology or isotope hydrology for instance you may have measured radiocarbon (^{14}C) on a sample with the purpose to determine the age of the sample. But up to this point we have no means yet to really get this age displayed. In the middle table of form *Export to Excel and Plot*, with the header *Values to compute from selected Attributes and Measurables*, you will find the names of functions that compute results from measured data. For instance, you will find functions here that compute the average depth from the upper and lower sampling depth, which is defined as *(Upper Depth + Lower Depth)/2*. For the Set just selected, there are several functions that compute radiocarbon ages. These are model ages for geochemical correction formulae which can be found in the appropriate literature. Another example for isotope hydrology would be the deuterium excess *d*, computed from ^{18}O and D according to the formula $d=D-8*^{18}O$. This way LabData can export any function that can be programmed in Visual Basic, and many useful functions of this kind are defined already. Some, but not all of them are documented in the LabData Manual. As an example, we will export the `Depth Screen Midpoint below Groundwater` and the `Depth Range Exposure` of the Exposure together with the `Deuterium Excess` and the ^{13}C corrected Closed System ^{14}C Age (Pearson & Hanshaw 1970), called `14C Age Closed System`.

Open *Export to Excel and Plot* and press **Clear All Selections** to create an empty form. Select `Set(s)` as *Data Anchor* and *Set* `1` as the only anchor ID. Select the functions discussed above. If you have a look at the attribute and measurable selection you will notice that LabData itself selected several attributes and measurables. LabData does this because it knows that these measurables and attributes are necessary to compute the secondary functions. Select `Tritium on Water` as an additional measurable please. Your form should look similar like the following picture:

Selection Form to be displayed.

Try the export and you will observe that the radiocarbon age of the selected data ranges between xy and xy. Already from the numbers you may notice that the radiocarbon age increases with depth in a systematic manner. However, it is much nicer to have a plot of these data readily created in Excel during export.

You might have noticed during selection of secondary functions, that this table contains many entries. It is far beyond the scope of this tutorial to explain in detail what these are useful for and how the numerical recipe behind works. If you are interested in these functions and how to create own user-defined functions in this field, you must consult the LabData Manual.

If you want to have data from more than one Set selected simply tic these other Sets as well in field *Select* in the upper left. LabData then will create one worksheet per selected anchor ID, in this case one worksheet per set. In case you selected any other data or in case your form is not empty in the other fields by any reason, the button **Clear All Selections** will help to achieve a virgin state of the form. If you want to keep for instance the selected anchors but delete only the selection of measured and attribute data, you use button **Clear All Data**. Similarly, and self-explaining, the buttons **Clear All Functions** and **Clear All Charts** work.

Creating Excel Charts from LabData

Getting the numbers and attribute information out of the database into Excel is nice, but how much nicer would it be to also generate a plot of the data! This is what the lowermost part *Plots to create* is good for.

With this output you created, it would be easy to see from these graphics that all stable isotope results lie on the meteoric water line. You can see from the depth profile of tritium that groundwater containing tritium is only found at depth shallower than 50m. So groundwater at greater depth might probably be older than 40 years. And you will notice that the ^{14}C groundwater model ages according to the closed system model increase with depth. So adapting the Excel Workbook used for output to your special needs can give you the opportunity to obtain the very first interpretations on any queried dataset by just pressing a few buttons.

Comparing results with Theoretical Functions

Charts are only useful if the compare data with a conceptualisation. This may be as simple as a straight line or even clustering of values in regions that are known already. This is what theoretical functions are good for. They allow you to display data in the context of models describing the data.
One of the simplest and most common plots in the world of isotopes is the comparison of measured values of $\delta^{18}O$ and $\delta^{2}H$ with the Global Meteoric Water Line (GMWL).

Parameters

Storing and retrieving Export Schemes

Now you've come so far exporting, displaying and modelling your data, I am sure you want to preserve this and not generate it again next time when the results for a few more samples trickled in. The way to preserve this export is to press the button **Save as New Export Setting**. You will be asked to provide a name all what you selected and as soon as you press the OK button you can retrieve this export again. You can test this by pressing the button **All** to select the Anchor & IDs, Data, Functions, Charts, Maps, Parameters, selecting xxx in the selection field **Name of Export Setting** and then pressing the button **Read Export Setting**. LabData will populate the whole form, including necessary measurables, secondary functions, theoretical functions, plots and the only thing you need to do is press the button **Export to Excel** to see the results. Note that what you store is a scheme. That means it is dynamic in the sense that the output changes if new data are available or if stored data change. If you remember the section "Numerical Procedures on the Server" any re-measurement of a SubSample will change the mean value for this measurable. Any new Sampling with new SubSamples and measurements will now be displayed as well.
Putting together a new export scheme can be quick and easy. If you have selected a Data Anchor, for instance a stratigraphic unit, and just want to quickly control a certain plot for this anchor, you can select this plot with the tick box **Single Chart** ticked and press **Read Export Settings**. LabData will select all the data necessary to display this plot. Of course you can successively add more plots and select more data.

If you've worked through the tutorial this far then you'll probably ask yourself what the sub-form Maps to create will help you to do. Well, it will do exactly this: create maps, in MS Excel. But this is something to be explained in the manual.

Additional Features

It is neither the scope of a tutorial nor did I try to describe the full functionality of LabData here. Probably this will never be done, since LabData evolves steadily. But nevertheless, the LabData Manual will evolve with a similar speed and already contains some additional information not described here. This information for instance includes details about the forms:

If you ever asked yourself what the *ToBeDone* in form *SubSamples* is good for, what the "O*ld Identifier*" are that exist on many forms, or what the meaning of a *Detection Limit*, an *Accuracy Limit* and the *Computation Parameter*s on form *Procedures* could be (page 17), you might find an answer there.
It will also contain information on special forms for special measurement techniques implemented in the Hannover, Vienna and Adelaide laboratories. Examples that are described already or that will be described soon are proportional gas counters for ^{14}C and tritium, electrolytic tritium enrichment, vacuum distillations and dilutions, Gas Chromatography and grain size measurements and the whole complex post-processing for noble gas measurements. Also the form(s) concerning Lumped Parameter Models are described in a separate manual.
Besides that for the export of data a list of secondary result functions can be found in the manual, that probably never will be complete but that is nevertheless quite useful. So, I would encourage the reader of this tutorial to have a look into the manual. If the desired feature is not yet described, it is at least worth to come in contact with me and ask for an update of the manual. Either you are lucky, and the feature is described meanwhile, or you can have the necessary short information via email.

Your Future

There are two ways for you to proceed now. One way is to keep LabData as it is and to start to hack your own data in. This has the disadvantage that there exist some Samplings, SubSamples and Values in your Database that are only worth existing for the tutorial. This is only a cosmetic disadvantage and should not be a large problem. At any time, you can decide to destroy the data of the tutorial: This is done using the menu command **Edit>>Delete Record** or with the toolbar button with the cross in red: . Of course, this is only possible if you have the permission to delete something on the SQL server level. Such permissions can be defined with a high level of detail in SQL server: you in principle can control which login is allowed to do what for any single table, but this of course would be quite a large administrative effort. In the laboratories where LabData is installed to date the permission to delete records for example is given to very few persons only, more persons have the right to edit data, often specific to some tables, and many persons have read-only access to the data.
Please note that deletion is only possible from the back end of the relations. So you first have to delete all values before you can delete the SubSample, you have to delete all SubSamples before you can delete the Sampling and so on. There are some helper functions in the GUI and some of these are described in the manual. The ID values for the deleted records will not be used any more, so you will have some gaps in your database.
The second way for you to proceed is to recreate the table structure, the Procedures and triggers as described on page 22, but either without any tables filled (leave out file `LabData Tables Fill for Tutorial.sql` completely) or with a version of initial data without procedures, Samplings, SubSamples, and results (use file `LabData Tables Fill.sql` instead). The former gives you the opportunity to define your parameters, kinds of coordinates, materials, units and so on from scratch, but I would not encourage you to do this: you will probably have to consult the manual even for the definition of a simple procedure and will probably be discouraged by many error messages. The latter is recommended because it provides you with a basic set of parameters, materials, units and measurables and you can directly start to define procedures.

Literature

Coplen, T.B. (1998): *A manual for a laboratory information management system (LIMS) for light stable isotopes*. Version 7.0. USGS open file report: 98–284, 124pp. Available at http://water.usgs.gov/software/lims.html

Pearson Jr., F.J., Hanshaw, B.B. (1970): *Sources of dissolved carbonate species in groundwater and their effects on carbon-14 dating.* Isotope Hydrology; IAEA-SM-129/18, Vienna.

Radke, B.M., Ferguson, J., Cresswell, R.G., Ransley, T.R. and Habermehl, M.A. (2000): *Hydrochemistry and implied hydrodynamics of the Cadna-owie-Hooray Aquifer Great Artesian Basin*. Bureau of Rural Sciences, Australia.

Suckow, Axel, Dumke, Ingolf (2001): *A database system for geochemical, isotope hydrological and geochronological laboratories*. Radiocarbon 43, No. 2, pp. 325-337.

And Thanks…

Several persons contributed in one way or the other to this software, some of them are probably not even aware how much.

Gudrun Drewes was the first person that really tried to apply the software and gave me the opportunity to erase the first severe errors in the GUI. She contributed a lot to the primary version of forms and functions concerning proportional counters in LabData.

Slavica Babinca was the first person that really demonstrated the value of the software since she was (besides me) first mapping a complete project in LabData and who used most of the post-processing functionality for her PhD. Especially *Lumpy* evolved by her definition of needs.

Robert van Geldern, my first successor in Hannover for the time I spent in Vienna, was the first "power user" who critically questioned some of the GUI functionality and who found several intolerable bugs that I had put into a longer queue during my own work with the system.

My special thanks go to *Manfred Gröning*, who decided to use LabData at the Isotope Hydrology Laboratory (IHL) of the International Atomic Energy Agency (IAEA) as the data management system of the laboratory. He also gave me the opportunity to further develop the system such that it can be applied in other laboratories around the world. Many fruitful discussions pushed the system forward and several important features only came into being discussing his ideas. The form "Set Status" described in the LabData manual is due to capabilities needed in the IHL to control the IAEA Data Reporting System (DRS).

Tyler Coplen reviewed the system during the 2007 IAEA symposium in his short and concise manner and suggested to include some capabilities to produce load lists for the machines and lists of the next SubSamples to be analyzed. This triggered the form "SubSample Status" that is described in the LabData manual.

Philipp Klaus was responsible for tritium measurement at IAEA-IHL, using the parts of electrolytic enrichment and LSC counting and provided many useful hints on making the system more user-friendly.

Finally I have to thank my family *Beate*, *Johannes* and *Christoph* for their patience during uncounted hours that I spent in front of the PC at home, programming…

No good deed goes unpunished, the feedback of the LabData users sometimes keeps me quite busy… But: may LabData be useful for as many laboratories as possible!