Hochschule
für nachhaltige Entwicklung
Eberswalde

Eberswalde University for Sustainable Development

Faculty of Forest and Environment

Work Report

For

Course: Forestry Data Structures and Spatial Data Models

Submitted to:

Dr. Evelyn Wallor

Submitted by:

Kazi Jahidur Rahaman

Date of Submission:

February 18, 2022

Winter Semester 2021/2022

1 Introduction

1.1 Background

The earth environment is evolving very rapidly and unique environmental problems are discovered every day. To solve this problems plenty of research works are also happening around the world, which requires a large amount of data which are produced by in-situ data collection, as well as remote sensing technologies. This huge load of data has been difficult to manage with antiquated data management practices. The concept of Database Management System (DBMS) came under focus to solve the complexity of data management in more efficient manner in 1960's which later has been improved by numerous inventions throughout the time and currently it's importance in every aspects of computing is beyond argument. Database is basically a collection of data and; DBMS is a set of those data and complex services to access and manage those data. (Silberschatz, Korth, & Sudarshan, 2010)

The concept of Database (DB), DBMS and Database System (DBS) are complex and inter-related. In plain words these can be generalized as, Database is data stored in organized and relational way which different programs can use, DBMS is the software program to manage these database which can also be a collection of different micro programs to perform individual operations (access control, data visualization, manipulation etc.) on database. And finally, the DBS is the combination of Database itself and the DBMS software.

In the environmental field, the data are mostly related to ecosystems and landscapes. From these sources abundant amount of data is collected with latest Geographic Information Systems (GIS) and Global Navigation Satellite System (GNSS) technologies, which can be quantitative as well as qualitative. (Wallor, 2022)This huge complexity is now a days tackled by the use of database systems. This is because the end users of these databases, who are non-computing professionals, can access and use these data on their own convenient way from anywhere through web based applications or even an Application Program Interface (API) without having to worry about the security, integrity, accidental loss of data or how the data is being stored.

1.2 Objective

The objective of this work is to conceptually design a soil measurement database which serves to provide easy access to specific types of data as the provided sample. The work also aims to construct the database in such that the consistency and integrity of the data is preserved during the update or deletion of existing data. Finally, it also expects to report on the performance on the developed database.

2 Material & Methods

2.1 Data

The data provided for the task consists of 2 excel sheets, among which sheet **'Datatable'** contains the data for the task and the sheet **'Metainformation'** contains the metadata for the explanation of the data. Metadata refers to a higher level description of data (Michener, 2006). This is particularly important for long-term studies where the database outlives the original investigator or where data are collected by scientists from many disciplines over a broad area, requiring considerable data integration and synthesis. (Michener, 2006)

In the sheet **'Datatable'** there are 22 columns and 46 rows of data. The columns are -

i. **Federal state** - Federal state of Germany where AOI belongs to
ii. **AOI** - Area of interest where soil sampling etc. has been conducted
iii. **ID C_{org} measurement** - ID of Corg measurement
iv. **Measurement** - Type of analysis / measurement (further analysis could be soil nitrogen, pH, etc.)
v. **C_{org} content** - Result of Corg measurement in [mg/kg soil]
vi. **y_coord** - Y coordinate of soil sampling point (Gauß-Krüger coordinate system)
vii. **x_coord** - X coordinate of soil sampling point (Gauß-Krüger coordinate system)
viii. **sample point no.** - Number of soil sampling point
ix. **campaign no.** - Number of sampling campaign
x. **sampling type** - Type of sampling (further types are: raster sampling; irregular sampling)
xi. **year of sampling** - Year of soil sampling
xii. **soil unit** - Soil unit to which sampling point belongs (soil map information)
xiii. **soil unit definition -** Definition of soil unit
xiv. **BD1** - General bulk density in [g/cm^3] of soil unit, first soil layer
xv. **BD2** - General bulk density in [g/cm^3] of soil unit, second soil layer
xvi. **BD3** - General bulk density in [g/cm^3] of soil unit, third soil layer
xvii. **FC1** - General water content at field capacity [vol%] of soil unit, first soil layer
xviii. **FC2** - General water content at field capacity [vol%] of soil unit, second soil layer
xix. **FC3** - General water content at field capacity [vol%] of soil unit, third soil layer
xx. **PWP1** - General water content at wilting point [vol%] of soil unit, first soil layer
xxi. **PWP2** - General water content at wilting point [vol%] of soil unit, second soil layer
xxii. **PWP3** - General water content at wilting point [vol%] of soil unit, third soil layer

According to the 'Metainformation' sheet, among the columns, the debatable holds different types of environmental data. In general, Environmental data can be defined as information closely related to the ecosystems and landscapes. Since environmental problems are often location dependent, they contain coordinates which allow it to be considered as spatial data too. (Wallor, 2022) Environmental data could also hold information about, Climate (temperature, precipitation, seasonality, solar radiation, relative humidity, vapor pressure), Rainfall, Cloud cover, Vegetation type, Per cent tree cover, Topography (elevation, slope, aspect, grid complexity), Hydrology (streams, drainage basins, flow accumulation, flow direction), Energy, Productivity, Land-cover, Land use, Soils (soil type, texture, soil water capacity and pH) etc. data. (Kenneth, Graham, & Wiens, 2008)

The data for this particular task contains location (x_coord, y_coord), Grid (sample point no, campaign no, sampling type), and Soil (Corg content, BD1, BD2, BD3, FC1, FC2, FC3, PWP1, PWP2, PWP3) data which categorizes the dataset as an environmental spatial dataset.

2.1.1 Data scales

The dataset contains both Qualitative and Quantitative scale data. The deeper categorization of the dataset according to the scales could be presented as the following table.

Table 1: Data scales and Datatypes of sample data

Column	Qualitative			Quantitative					Field datatype
	Binary	Nominal	Ordinal	Discrete	Continuo	Interval	Closed	Ratio	
Federal state		×							Character varying
AOI		×							Character varying
ID C_{org} measurement				×		×			Integer
Measurement (type)		×							Character varying
C_{org} content					×			×	Numeric
y_coord				×		×			Bigint
x_coord				×		×			Bigint
sample point no.				×		×			Integer
campaign no.				×		×		×	Integer
sampling type		×							Character varying
year of sampling				×		×			Integer
Soil unit		×							Character varying
soil unit definition		×							Character varying
BD1, BD2, BD3					×	×			Numeric
FC1, FC2, FC3					×	×			Numeric
PWP1, PWP2, PWP3					×	×			Numeric

2.2 Software

Environmental datasets often comes with additional data (i.e. business data, documentations etc.). Therefore, efficiency in managing complex data should be considered during the selection of software. (Pokorný, 2006) The provided dataset for this task along with measurement data also holds additional data such as Soil unit definition, Sampling Type, measurement type etc.

Although a real database is not asked to be developed rather a conceptual model is to be developed, we consider the future aspects of upgradation and select PostgreSQL as our language to develop our Database System (DBS). PostgreSQL comes with a 3 tier architecture in pgAdmin 4 software. The layers of PostgreSQL are-

a) **Presentation layer** - The Graphical User Interface
b) **Business logic** - Activities, rules, calculations controlled with Postgre Structured Query Language
c) **Database storage** - Stores data the server side

These layers are implemented as thick 2 –tier concept in pgAdmin 4 software.

Few diagrams of the task were constructed with the help of Draw.io web application, which is a free and open source cross-platform graph drawing software developed in HTML5 and JavaScript.

2.3 Approach

2.3.1 Data model

A data model in general could be expressed as a virtual representation of a real life data scenario with the help of database concepts. Data models can be conceptual as well as implementative which would offer ways to define, query, manipulate and administrate the stored data. According to the requirement of the task, this report only highlights the Conceptual Data Model for the sample dataset.

Conceptual data model illustrates the logical structure, relationships and dependencies within the data using graphical display or Entity-Relationship- Model. This should be constructed in such a way that it remains independent in respect of thematic and software dependencies. Because, the same data model often needs to be implemented in different software environment in professional fields.

Considering the requirement of Simplicity, we consider Entity Relationship Model (ERM) to construct the Conceptual data model. The ERM is the bridge between the end-user of the DBS and the DB developer. Among the plenty of methods to describe the ERM's, the Entity Relationship Diagram (ERD) method is chosen to implement the conceptual data modelling for the provided dataset. The ERD presents a graphical representation of the Entities, Attributes and Relationships of the database. The key elements of an ERD can be described as the following-

i. **Entity:** Individual element of the conceptual model of database. Each row of the dataset is considered as an entity of our database.
ii. **Attributes:** Each characteristic or property of an entity is called its attribute. The attributes has their unique value for each instance of a particular entity from the value which are summed up as domain.
iii. **Relationships:** The connection between 2 or more Entities of the data model is called Relationship. Based on the degree of connections with other entities, relationships can be 1:1 – relationship, 1:n – relationship, or m:n – relationship
iv. **Constraints:** The constraints are the restrictions that distinguish the entities from other entities by defining the relationship types, data types, data domain, cardinality, and dependency to other entities etc.

2.3.2 Working Steps

2.3.2.1 Field data types

Before starting to implement the data model we first need to understand the field data types. The overview of the data-table shows it contains both numbers and text data. The database system we use (PostgreSQL) has a rich set of data types to accommodate different types of real life data in the database. The complete list of data types available in PostgreSQL is available in (The PostgreSQL Global Development Group, n.d.) The datatypes applicable sample data of this work is summed up in the *Table 1*.

2.3.2.2 *Preliminary ERD*

The dataset we are provided has 22 columns. As a database developer we consider the provided data to be appropriate to design the database. So, we consider each of the columns as an entity type and design our ERD based on that. Following *Figure 1* represents the preliminary ERD of the database.

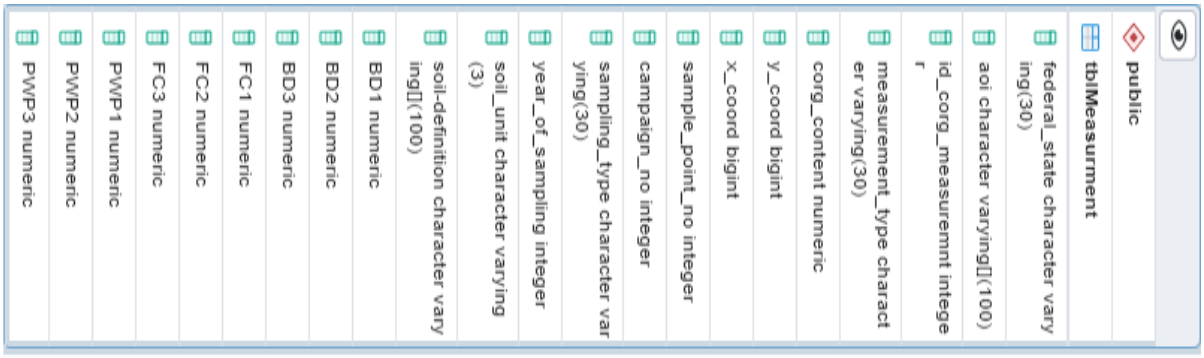

Figure 1: Preliminary Database ERD

After creating the database we can see that there are many redundent(multiple copies of same data) values for several columns (Federal_state, aoi, id_corg_measurement, measurement_type, sampling_type, year_of_sampling) and number of empty columns(soil_unit_description, BD2, BD3, FC1, FC2, FC3, PWP1, PWP2, PWP3).

	Federal_state character varying	aoi character varying	id_corg_measurement integer	measurement_type character varying	corg_content numeric	y_coord bigint	x_coord bigint	sample_point_no integer	campaign_no integer	sampling_type character varying	year_ intege
1	Brandenburg	study area Ziethen	7	Corg	4424.7	5867488	5424325	1007	1	permanent sampling	
2	Brandenburg	study area Ziethen	8	Corg	4436.0	5867940	5424663	1008	1	permanent sampling	
3	Brandenburg	study area Ziethen	21	Corg	6349.6	5869885	5427332	1021	1	permanent sampling	
4	Brandenburg	study area Ziethen	23	Corg	4841.6	5869787	5426839	1023	1	permanent sampling	
5	Brandenburg	study area Ziethen	50	Corg	8863.4	5871690	5427881	1051	1	permanent sampling	
6	Brandenburg	study area Ziethen	50	Corg	8863.4	5871690	5427881	1051	1	permanent sampling	
7	Brandenburg	study area Ziethen	98	Corg	5112.3	5867488	5424325	1007	2	permanent sampling	
8	Brandenburg	study area Ziethen	99	Corg	4505.3	5867940	5424663	1008	2	permanent sampling	
9	Brandenburg	study area Ziethen	112	Corg	6928.0	5869885	5427332	1021	2	permanent sampling	
10	Brandenburg	study area Ziethen	114	Corg	7618.0	5869787	5426839	1023	2	permanent sampling	
11	Brandenburg	study area Ziethen	141	Corg	10597.1	5871690	5427881	1051	2	permanent sampling	
12	Brandenburg	study area Ziethen	189	Corg	3875.5	5867488	5424325	1007	3	permanent sampling	
13	Brandenburg	study area Ziethen	190	Corg	6313.2	5867940	5424663	1008	3	permanent sampling	
14	Brandenburg	study area Ziethen	203	Corg	6598.8	5869885	5427332	1021	3	permanent sampling	
15	Brandenburg	study area Ziethen	205	Corg	7391.6	5869787	5426839	1023	3	permanent sampling	
16	Brandenburg	study area Ziethen	232	Corg	10622.2	5871690	5427881	1051	3	permanent sampling	

Figure 2: Preliminary Database (1)

soil_unit character varying	soil_unit_definition character varying	bd1 numeric	bd2 numeric	bd3 numeric	fc1 numeric	fc2 numeric	fc3 numeric	pwp1 numeric	pwp2 numeric	pwp3 numeric
D2a		1.55	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D2a		1.55	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5b		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5b		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5a		1.55	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5a		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D2a		1.55	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D2a		1.55	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5b		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5b		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5a		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D2a		1.55	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D2a		1.55	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5b		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5b		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
D5a		1.63	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]

Figure 3: Preliminary Databse (2)

Figure 2 and *Figure* 3 shows the condition of the database designed based on the preliminary ERD where we experience the following problems-

1. Failing to retrieve the target row because of multiple copies of same information.
2. Multiple copies of same data might lead to inconsistency
3. Unintentional data deletion. User might want to delete one instance of data but because of redundancy all instances might be deleted.
4. Same can happened for update case too.
5. More time and space complexity.

To overcome these drawbacks, we should consider Normalization of the database. Normalization is the process of organizing the data in such a way, so that the issues resulting from the redundancy of data in database might be solved. In normalization the universal data table is divided into smaller schemas which are later rejoined using the JOIN queries during the data retrieval. The idea behind normalization is the analysis of functional dependencies, that is, if value of a particular attribute A is dependent on another attribute B, we can easily extract the value of A from B. In this sense, normalization is also the process of removing the redundancy and improve integrity, efficiency, and scalability of the database.

2.3.2.3 Normalization

The goal of normalization is to create schemas which are appropriate normal form. For understanding weather data are in our expected normal form, we need additional information about the real world system we are designing the database for. We can also look for what is wrong with the current data formation (Silberschatz, Korth, & Sudarshan, 2010), for which we try to understand the provide data, and identify the problems with it.

1. **Understanding the data:** One of the main targets of a database is to ensure the easy access, integrity and security of the data. Before starting to construct the ERD we first try to understand the data. We look for the redundancies within the data. Some incident of repeating entries are listed below which shall influence our normalization process.

i. The value group (*Brandenburg, study area Ziethen, C_{org}*) for (*Federal state, AOI, Measurement*) is repeated several times. Also the *ID C_{org} measurement* column has repeated values too that can be seen in *Figure 4*.

1	Federal state	AOI	ID Corg measurement	Measurement
2	Brandenburg	study area Ziethen	7	Corg
3	Brandenburg	study area Ziethen	8	Corg
4	Brandenburg	study area Ziethen	21	Corg
5	Brandenburg	study area Ziethen	23	Corg
6	Brandenburg	study area Ziethen	50	Corg
7	Brandenburg	study area Ziethen	50	Corg

Figure 4: Repeating value group for (Federal state, AOI, Measurement)

ii. The attribute group *(campaign no., sampling type, year of sampling, soil unit)* has redundancy in their values. Also, the entries of (*y_coord, x_coord, sample point no.)* also has more than one entries which is shown in *Figure 5*.

y_coord	x_coord	sample point no.	campaign no.	sampling type	year of sampling	soil unit
5867488	5424325	1007	1	permanent sampling	1992	D2a
5867940	5424663	1008	1	permanent sampling	1992	D2a
5869885	5427332	1021	1	permanent sampling	1992	D5b
5869787	5426839	1023	1	permanent sampling	1992	D5b
5871690	5427881	1051	1	permanent sampling	1992	D5a
5871690	5427881	1051	1	permanent sampling	1992	D5a
5867488	5424325	1007	2	permanent sampling	1993	D2a
5867940	5424663	1008	2	permanent sampling	1993	D2a

Figure 5: Repeating value group for (campaign no, sampling type, year of sampling and soil unit)

iii. From *Figure 5* it is also seen that the columns (*soil unit definition, BD2, BD3, FC1, FC2, FC3, PWP1, PWP2, PWP3)* has null entries that means they add no new information for the current state of the data.

soil unit definition	BD1	BD2	BD3	FC1	FC2	FC3	PWP1	PWP2	PWP3
	1.55								
	1.55								
	1.63								
	1.63								
	1.55								
	1.63								
	1.55								
	1.55								

Figure 6: Entries with no new information

2. **Identifying Entity – Relationship - Attributes:** As mentioned earlier, during normalization process we consider dividing the large universal table in smaller schemas. In this case, we prefer grouping the attributes to be in the same schema which are related to each other either in real life or related by their values present in the provided datatable.

    i. **Entity and Entity type:** Each row of the datatable is an entity of our database and the type of the entities are called the entity type. For example, the preliminary datatable holds the data for a soil measurements. Each row of the measurement is an entity of that, whereas 'Soil Measurement' is an entity type.
    But, due to the drawbacks of single-entity type model, we consider dividing our one entity type in multiple smaller entities (schemas/tables). For this the normalization we choose the entity types, based on our understanding of data. The entity types are Measurement Type, Soil Parameter, Measurement, Campaign, Soil, Sample, Sampling Type, and Area presented graphically in *Figure 7*.

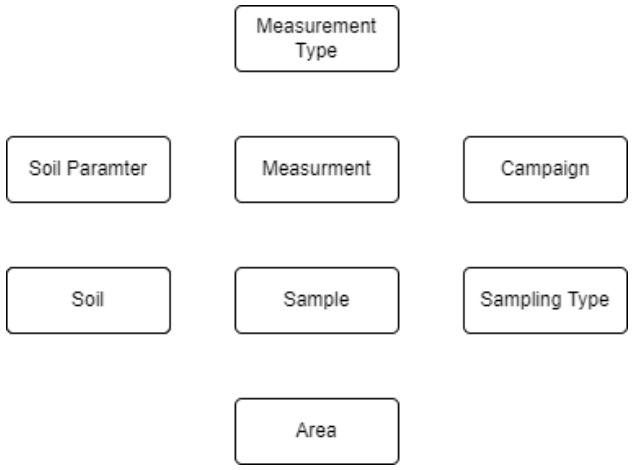

Figure 7: Proposed Entity Types

    ii. **Attributes:** The characteristics or properties of the entities are the attributes of the entity. The each entity of the entity type Soil Measurement has attributes like Federal state, ID C_{org} measurement, C_{org} content, y_coord, x_coord, sample point no, campaign no, sampling type, soil unit, soil unit definition, BD1, BD2, BD3, FC1, FC2, FC3, PWP1, PWP2, PWP3 etc.
    Since we are considering more than one smaller entity types instead of one large entity type to normalize our database, we consider our entity types mentioned above. During defining the entity types we encompass all the attributes of our universal table. For the convenience of access of the database, we introduce some new identifier attributes in some entity types, i.e. soil_parameter_id, area_id. Reason behind introducing the attribute was also to avoid the complexity of matching 'character varying' type value during the accessing (selection queries) the database contents. The entity types are shown on the following *Figure 8*, where each rectangle represents an entity type, the

header of the rectangle holds the entity name and the following list holds the attributes of the entity type.

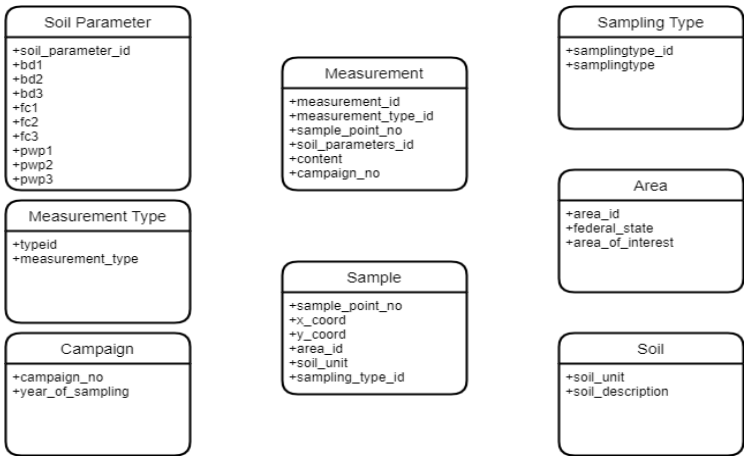

Figure 8: Attributes of the Entity Types

These attributes can have their own values for each of the entities, and the set of these values is called the domain.

iii. **Relationships:** The entity types are interrelated within themselves. This can be explained by, Each *Measurement* should belong to a *Sample*, while each *Sample* also should be from an *Area*.

The Entity-Relationship- Attributes could be exemplified with the following table-

Table 2: Entity, Entity Type and Attribute:

Entity Type	Sample	Sample
Entity	Sample Point No 1007	Sample Point No 1008
Attribute	y_coord	soil_unit
Attribute Value	5867488	D2a
Domain	[5867488, 5871690]	[D2a, D3a, D5a, D5b]
Relation Type	Sample has soil	Sample belongs to a sampling type

While the Entity-Attribute-Relation can be explained by the above table, the overall relationships of the database can be mapped by the following diagram in *Figure 9*.

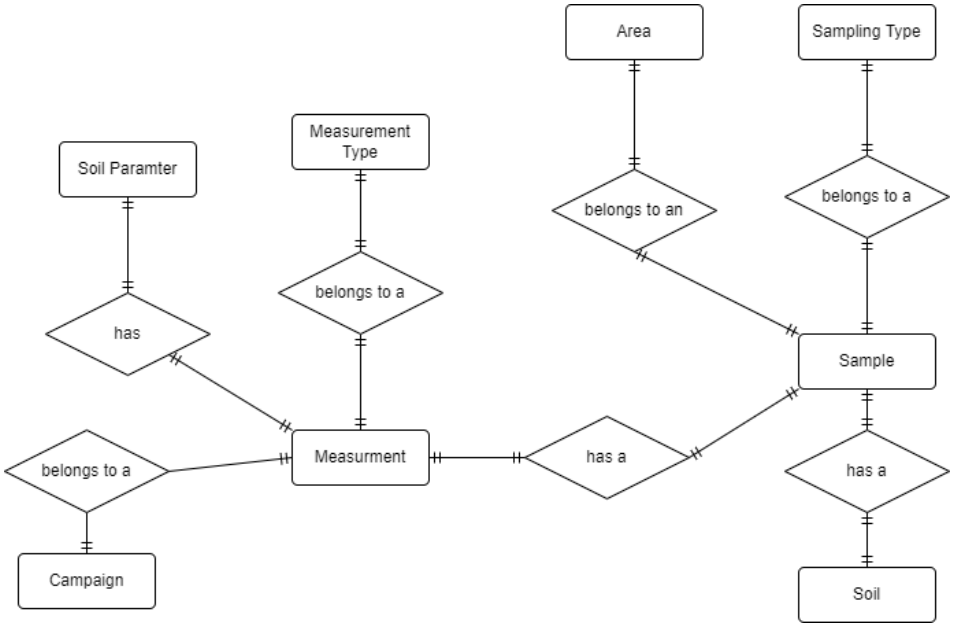

Figure 9: Relation in Proposed ERD

The proposed relations could be explained as following-

a) The database contains data of soil organic content measurements. Each measurement has a sample (1:1 relationship), a Soil Parameter entry (1:1 relationship). The measurement also belongs to a campaign and a measurement type (1:1 relationship)
b) Each sample can have multiple measurements (1: m relationship).
c) Each Sample belongs to an area (1:1 relationship) and a Soil (1:1 relationship). The sample also belongs to a Sampling type. On the other hand each soil unit can have multiple samples and each area can also have multiple samples. (1: m relationship).

3. **Identifying Keys:** By conception, each entry of an entity type (entity) is distinct. So, the database should not content two entity of same attribute values, this could also be said as the uniqueness of the entities and also applicable for ensuring eradication of redundancy of database. (Silberschatz, Korth, & Sudarshan, 2010) Thus, a key can be defined as a set of attributes with which an individual entity can be distinguished uniquely.

For a particular database keys can be of several types such as-

i. **Primary key:** Primary key is minimal an attribute or a set of attributes of an entity type which can uniquely identify an entity within the relation. While no minimal set is available to uniquely identify an entity, then surrogate ID is introduced as a Primary key.

ii. **Foreign Key:** Foreign keys are the keys in an entity type which has inherits value from another entity type (reference schema). In this case, the referencing key should be the primary key of the referenced table.

iii. **Composite key:** Set of more than one attribute which are used to identify an instance of a particular entity type. In particular cases, the composite keys can also act as the primary key of a particular entity type.

The following table summarizes the key distribution of our proposed database design.

Table 3: Keys for the proposed ERD

Entity Type	Primary Key	Foreign Key	Composite Key
Measurement Type	Type_id(surrogate id)	-	-
Soil Parameter	soil_parameter_id(surrogate id)	-	-
Measurement	measurement_id, measurement_type_id, soil_parameters_id	sample_point_no, campaign_no	measurement_id, measurement_type_id, soil_parameters_id
Campaign	campaign_no	-	-
Soil	soil_unit	-	-
Sample	sample_point_no	area_id, soil_unit, sampling_type_id	-
Sampling Type	samplingtype_id(surrogate id)	-	-
Area	area_id(surrogate id)	-	-

For the entity types Measurement Type, Soil Parameter, Area and Sampling Type we consider creating surrogate id for each of the entities and use them as their primary key while for Campaign, Sample, and Soil we consider campaign_no, sample_point_no, and soil-unit as primary key respectively. The primary key for the entity type Measurement is a composite key of measurement_id, measurement_type_id and soil_parameters_id. Because the entities of measurement type cannot be uniquely identified with a single attribute value.

The measurement table also uses the sample_point_no, campaign_no attributes of Sample and Campaign entity type respectively to inherit the value from corresponding tables. So, there are the foreign keys of the Measurement entity type. Similarly, the foreign keys for the table sample are the area_id, soil_unit, sampling_type_id for the Sample table using of these foreign keys help to avoid entering values multiple times, rather uses single instance of a value in multiple cases.

4. Created the Entity-Relationship Diagram with pgAdmin 4 software.

3 Results

After identifying all the building blocks of the Entity Relationship Diagram (ERD) the main diagram was constructed using the pgAdmin 4 software. The final output of the process provide us with the following ER Diagram in *Figure 10*. In the diagram, each rectangles represent each of the entity types while the connecting lines among the entity types represents the relationships among the entity types.

Figure 10: Proposed ER Diagram Developed with pgAdmin 4 Software

4 Discussion

The ER Diagram presented in *Figure 10* shows the entity types, attributes, field data types, and relations of the proposed normalized database. It is expected to solve the drawbacks which were present in our preliminary (Non-normalized) database.

The proposed model divides the universal large entity type in 8 smaller entity types. Each entities of a particular type can be uniquely identified by using the primary key of that particular table. For example-

1. *Select **
   FROM tblSample WHERE sample_point_no = '1007';

The above query should provide us with the sample info of the sample point which holds the sample point number 1007. Similarly, we can retrieve the information of individual *Area, Campaign and Soil* entity types by making select queries and using their respective primary key in the where condition. This is possible due to the introduction of primary keys in the entity types, which also ensures absence of repeating columns without new information. Thus the design achieves 1st Normal form.

On top of that, for avoiding the redundancy the proposed ER Diagram also suggested introduction of foreign keys. As a result the entity types which share some common information are not required to repeat the same information, rather all the information can be extracted with a single Join query.

2. *SELECT **
   FROM tblMeasurement AS Measurement
   INNER JOIN tblSample AS Sample ON Measurement . sample_point_no = Sample . sample_point_no
   INNER JOIN tblSoilParaMeters as SoilParamter ON Measurement . soil_paramters_id = SoilParameter . soil_parameters_id
   WHERE Measurement . sample_point_no = 1051
   ORDER BY Measurement . measurement_id ASC;

The query mentioned in 2 should provide with the Measurement as well as the Sample information for the sample point number 1051. The query first find the entities where the sample_point_no is 1051. This should have 2 entities of measurement selected from the tbMeasurement table since we have two entities where sample_point_no is 1051 and then it joins the sample information with the measurement information and soil information. This statement should also decrease the time complexity significantly since because of using inner join it significantly reduces the tuples which the query needs to search in.

However, the database design proposed in this work does not ensure 2nd Normal for which is removing the functional dependencies. Because, to work with complex environmental data, the database designer should have enough background and subject related knowledge which the author (also the database designer) of this report lack. As a result, no particular mathematical or dependencies could be retrieved among the measurements like Bulk Density, Water content in Field Capacity, Water Content in Permanent Wilting Point and C_{org} . But, as a result of introducing the key concepts the empty cells which were

prevalent for the Soil Unit definition and BD2, BD3, FC1, FC2, FC3, PWP1, PWP2, and PWP3 columns of the data-table, were tackled by allowing only the rows which hold information in the database.

The most complex aspect of the database was to accommodate the entities where all attributes except the BD1 values were same. To address the incident the soil parameter entries were accommodated and given a unique soil_paramter_id so that those can be correlated to the measurement and sample information. The introduction of composite primary key was also done to address the issue. Nevertheless, although the ER Diagram presented in this report suggests a solution which should fulfill most of the common requirements of an accessible, efficient database, there should be further effort to improve the design with more background knowledge of soil science.

5 Conclusion

The goal of this work was to develop a sophisticated database structure which can not only accommodate a large number of soil measurements data, but also ensure that the data can be easily accessed for using in less amount of time while also keeping the consistency of data. The proposed database design has easily been explained in this report. The normalization concept applied in the design to ensure less redundancy as well as less time complexity. The application of several key concepts also contributes to lessening the complexity also ensures the database is automatically updated while updating or deleting an existing data. Although the efficiency of the design is discussed, there is still aspects where improvement can be made with more background knowledge of soil science.

6 References

Kenneth, K. H., Graham, C. H., & Wiens, J. J. (2008, March). Integrating GIS-based environmental data into evolutionary biology. *Trends in Ecology & Evolution, 23*(3), 141-148. doi:https://doi.org/10.1016/j.tree.2008.02.001

Michener, K. W. (2006, January). Meta-information concepts for ecological data management. *Ecological Informatics, 1*(1), 3-7. doi:https://doi.org/10.1016/j.ecoinf.2005.08.004

Pokorný, J. (2006). Database architectures: Current trends and their relationships to environmental data management. *Environmental Modelling & Software, 21*(11), 579-1586. doi:https://doi.org/10.1016/j.envsoft.2006.05.004

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database System Concepts* (6 ed.). McGraw-Hill.

The PostgreSQL Global Development Group. (n.d.). *Documentation PostgreSQL 9.2*. Retrieved from PostgreSQL : https://www.postgresql.org/docs/9.2/datatype.html

Wallor, E. (2022, January). Environmental data. *Forestry data structures & spatial data models*. Eberswalde, Germany: Eberswalde University for Sustainable Development · Faculty of Forest and Environment. Retrieved from https://lms.hnee.de/pluginfile.php/9572/mod_resource/content/1/Lecture_1.pdf