Towards an ISRIC WOrld Soil Information Service (WOSIS Version 1.0)

ISRIC World Soil Information

ISRIC Report 2013/02

Piet Tempel, Daniël van Kraalingen, Jorge Mendes de Jesus and Hannes I. Reuter

Towards an ISRIC WOrld Soil Information Service (WOSIS Version 1.0)

Piet Tempel, Daniël van Kraalingen, Jorge Mendes de Jesus and Hannes I. Reuter

ISRIC Report 2013/02

Wageningen, May 2013

Director, ISRIC – World Soil Information
PO BOX 353
6700 AJ Wageningen
The Netherlands
E-mail: soil.isric@wur.nl

Additional information on ISRIC – World Soil Information can be accessed through http://www.isric.org

Citation
Tempel, P., D. van Kraalingen, Jorge Mendes de Jesus and Hannes I. Reuter, 2013. *Towards an ISRIC World Soil Information Service (WOSIS Version 1.0)*. Report 2013/02, ISRIC-World Soil Information, Wageningen. 188 pp.

ISRIC Report 2013/02

Contents

List of figures

Preface

ISRIC - World Soil Information, which hosts the World Data Centre for Soils, has a mandate to serve the international community as custodian of global soil information and to increase awareness and understanding of soils in major global issues. It aims to strengthen its position as a leading institute in collecting, storing, processing and disseminating global soil and terrain information for research and development of sustainable land use. Developments in soil science, such as low cost techniques to measure soil data, reduced handling and storage costs of data, and web-based data access have progressed rapidly during the last decade. To better manage the current demand for soil information, and being in the frontline of information services, ISRIC is in the process of updating its enterprise data management system, including revised procedures for registering newly acquired data, quality assessment/control and versioning. The centralized enterprise database will contain validated and authorized data with defined and registered accuracy and quality (e.g., known data lineage). Further, ISRIC is in the process of developing a series of web-based services, known as Global Soil Information Facilities (GSIF), which will be made available gradually to the wider user community. Where appropriate, algorithms and model states will be documented and preserved. Recommended procedures are being developed and tested to align with the frameworks of on-going, large international projects such as the FAO led Global Soil Partnership (GSP), GlobalSoilMap.net and e-SOTER and other international programmes. The Global Soil Partnership wants to improve governance of the limited soil resources of the planet in order to guarantee healthy and productive soils for a food secure world using a wide agenda. The GlobalSoilMap.net project is an initiative of a global consortium of scientific institutes that collaborate to make a new digital soil map of the world using state-of-the-art and emerging technologies. The e-SOTER project delivered a web-based regional pilot platform with data, methodologies, and applications, using remote sensing data to validate, augment and extend existing terrain and soil maps. These products will also serve the global soil observing system as part of the Global Earth Observing System of Systems (GEOSS).

This document describes the rationale and structure for version 1.0 of the World Soil Information Service (WOSIS). The system has been developed at ISRIC, as a collaborative effort with partners from Alterra, Wageningen University & Research centre, and inputs/advice from a range of external experts.

Dr. ir. Prem Bindraban
Director, ISRIC World Soil Information

Summary

Over the years, ISRIC – World Soil Information has developed and maintained a number of (soil information) databases that are freely available to the scientific community and other non-commercial groups. In the past decade, however, dissemination opportunities have changed drastically as a result of new (web) technologies that permit faster and new forms of information delivery. Strategies adjusted to these opportunities are being pursued for improved information retrieval and dissemination.
In this context, ISRIC has implemented a centralized and user-focused database ('world soil database') containing only validated and authorized data - preferably with a known and registered accuracy and quality. As a result, all data managed or maintained by ISRIC will be made available on-line from one central database environment in accordance with the conditions for data sharing defined by the various data providers. The entire system will allow access, processing and visualization through one set of tools. The database will support geographic objects, since most of ISRIC's datasets are at least partly spatially defined. Finally, the world soil database can be queried using standardized interfaces based on known XML/GML exchange models.

The database design as discussed in this report consists of 70 interrelated tables. The database has a number of schemas, which in turn contain the data tables (and other database objects). An important reason to use schemas is to organize database tables in logical groups to make them more manageable. Another important use for schemas in the world soil database is to mimic a federated database system. This is a type of meta-database management system which transparently integrates multiple autonomous database systems into a single virtual database. Each schema is described in detail the report.

Future expansion and testing of the world soil database is seen as an iterative process that is largely governed by the addition of new soil information datasets (including spectral data). Instrumental to enhanced usability and accessibility of the world soil database will be the harmonization of soil property values (i.e. their domain), as well as standardization of analytical procedure descriptions. Use of these interfaces will allow for the fulfilment of future demands for global soil information, and enable further incorporation of soil data held by third parties.

NASA's Directory Interchange Format (DIF) is the metadata standard used for describing ISRIC datasets. GeoServer is used to serve geospatial data from the database - profile locations, SOTER maps, GlobalSoilMap.net products, etc. A stack of open RESTFull web services will be specified in the next version of this document, supporting (exchange) current and future data models.

1 Introduction

ISRIC's strategic plan of June 2009 states: "The ISRIC mandate implies that it will be active in collecting and handling soil and terrain data, encourage the good use of that data and, in advocacy of the role and importance of soil in global development issues"

This mandate is also clearly reflected in the institute's registered name: International Soil Reference and Information Centre.

To meet its goal, ISRIC has developed, and currently maintains, a number of databases that are freely available to the scientific community and other non-commercial groups. As stated in the strategic plan, dissemination opportunities have changed drastically in the past decade as a result of new (web) technologies which permit faster and new forms of information delivery. Strategies adjusted to these opportunities should be pursued for improved information retrieval and dissemination.

At the time, challenges to implementing such a strategy included:

- Limited facilities to browse data online. For example 'Show me, what data ISRIC has for Cuba, then I'll pick what I need'.
- Lack of a consistent interface to query the website for specific data. It was not possible to get an answer from the ISRIC website to a simple question like 'Give me a pre-defined set of data from ISRIC's data holdings for all soil profiles in Cuba with a pH smaller than 5.8'.
- Metadata for the data sets are query-able through NASA's Global Change Metadata Directory (GCMD), which hosts a portal for the ICSU World Data Centre System[1]. Yet, these 'descriptive metadata' could not be queried at the ISRIC site.
- Data sets were in project-specific formats, and there was no functionality to modify formats in response to user needs. Data were organized as a fragmented set of individual datasets that have evolved in the past years from projects and personal initiatives. These datasets have been made available in a form (or format) geared towards their originating projects' goals; as such, they may not necessarily fulfil the goals or needs of a wider user community.
- Support all kind of data storage, standardization, harmonisation and export for the Global Soil Information Facilities (see Batjes et al., 2013).

To address these challenges and enable the information strategy outlined in the strategic plan for 2009-2012 (see ISRIC, 2010), ISRIC has developed:
'A centralized and user-focused database containing only validated and authorized data with a known and registered accuracy and quality.' The resulting system, known as 'ISRIC World Soil Information Service' (WOSIS), is described in this report.

[1] ISRIC - World Soil Information, the World Data Centre for Soils since 1989, is a regular member of the ICSU World Data System (ICSU-WDS).

1.1 Approach

As a first step, an inventory was made of all data sets present at ISRIC that might be eligible for inclusion in the new world soil database. ISRIC staff was invited to submit descriptions of these data sets. The next step was to define selection criteria for data to be included in the first approximation of a world soil database. This was a crucial step since it defines the information content (i.e. the kind of questions and queries the database should be able to provide answers to) as well as its initial complexity. The single most important selection criterion the Steering Group formulated was that only soil, site, and terrain data would be handled in the emerging world soil database. Application of these criteria greatly reduced the number of relevant data sets. However, there is a lot of overlap between these datasets; the same information is included in more than one database, merely organized in a different way.

Based on this first analysis, the following objectives for a world soil database were formulated. It should:
- be able to handle and harmonize soil profile and terrain data, such as held in WISE (Batjes, 2009), ISIS (2006), SOTER (Van Engelen and Dijkshoorn, 2012) and GlobalSoilMap.net (Sanchez et al., 2009)[2], including functional properties, soil sample analysis results, horizon analysis results, terrain, terrain components, soil classifications, as well as spatial data (polygon and raster format);
- meet quality aspects: availability, continuity, consistency and robustness;
- be powerful enough to become the reference database for future, permanent storage;
- serve the international community with quality controlled soil data in a most effective way.

Inclusion of a spatial component is an important requirement since most of ISRIC's datasets are at least partly spatially defined. Therefore, the new world soil database should support the storage, spatial indexing and processing of spatial data. Further, the enterprise Relational Database Management System (RDBMS) should include a spatial extension.

The system should also support of standards-based metadata registration at dataset level as well as at the level of individual data objects and attributes[3].

Finally, the world soil database should be query-able using standardized interfaces based on XML/GML exchange models – notably those defined by the Open GeoSpatial Consortium (OGC). These interfaces must fulfil future demands for global soil information, and enable the inclusion of soil data of third parties.

Based on the analysis of the 'current' (sensu 2009) ISRIC datasets, developments towards new datasets, and the desired functional features of the world soil database, the system should:
- Support current ISRIC datasets (ISIS, WISE, SOTER, soil degradation datasets), that is: capable to incorporate (any) data from aforementioned datasets.
- Allow for new future datasets, so extensibility and flexibility are important.
- Provide flexibly to extend attribute sets for data objects.
- Be able to accommodate metadata for the object and dataset level.
- Provide support for spatial objects, both vector (point and polygon) and raster based.
- Conform to current and future (exchange) data models (e.g., e-SOTER SoterML (Pourabdollah et al., 2012), SoilML (Montanarella et al., 2010) and GeoSciML (CGI 2013)).
- Support for GML and KML for easier and faster data dissemination.

Henceforth, '**WOrld Soil Information Service**' and its acronym '**WOSIS**' will be used for naming purposes in this document.

---

2 http://www.isric.org/projects

3 At present, NASA's Directory Interchange Format (DIF) is the metadata standard used for describing ISRIC datasets, in accordance with the ICSU-WDC system requirements. Some 50 datasets have been described using DIF at NASA's Global Change Master Directory, see under 'WDC for Soils' as data center.

2 Database design

2.1 General Concept

The database design consists of 75 interrelated tables following the standard relational model. These tables have a number of common characteristics. Each table has a single-field, with a unique primary key of type integer (long) and a conventional (descriptive) name. Except for the four tables in the System schema, all tables share the following five fields (in this order, and shown at the end of a table's fields list):

- **Created**: A DateTime field that defaults to the current date and time when a new record is inserted.
- **Modified**: A DateTime field recording the latest modification of a record; defaults to the current date and time when inserting a new record, or updating a record.
- **Editor**: A text field recording the login role that created or last modified the record; defaults to the current login role when inserting a new record, or updating a record.
- **RevisionId**: Reference to the last revision (i.e. update) of the record in the audit trail.
- **Version**: Sequential record version number - starting with 1 for the initial entry.

Users cannot interact with any of the above fields directly. The fields are initialized or updated by the database system to allow for record change tracking and audit.

Tables holding measured property values contain the following fields which are explained in detail below:

- **Date**: date of observation or measurement.
- **Trust**: level of trust in the given value: 'A' as entered, no validation, 'B' - harmonized and standardized, 'C' (semi)-automatic error checking and 'D', the highest level, for data validated by a soil scientist.
- **LoD**: Level of Determination - accuracy of an observation or measurement precision.
- **Quality**: Quality indicator - 0 is the lowest and 255 the highest quality level.

WOSIS consists of a group of schemata, which in turn contain tables (and other database objects), as shown in Figure 1. Schemata are used for organization purposes, to group the database tables (and other database objects) into logical groups making them more manageable. Schemata are analogous to directories in an operating system level, except that schemas cannot be nested. The same table name can be used in different schemas without conflict. For example, both **ThisSchema** and **ThatSchema** may contain tables named **MyTable.** The **MyTable** is normally described using the schema name e.g.: **ThisSchema.MyTable** or **ThatSchema.MyTable.**

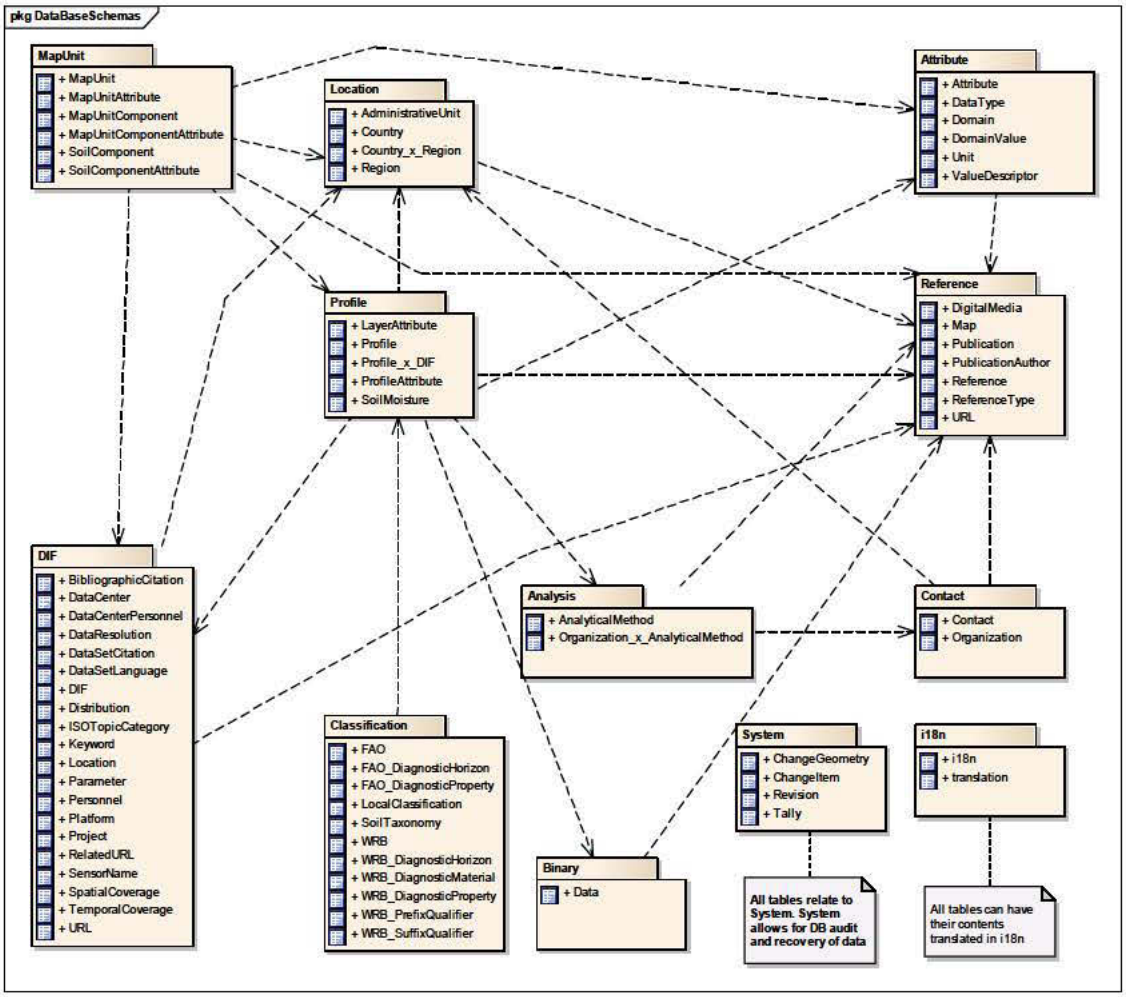

Figure 1
Database schemas.

The Version 1.0 of the world soil database model makes use of the following (10) schemas (number of tables in each schema in parentheses):

- Analysis (2)
- Attribute (6)
- Binary (1)
- Classification (11)
- Contact (2)
- DIF (21)
- Location (4)
- MapUnit (7)
- Profile (6)
- References (7)
- System (4)
- i18n (Internationalisation) (2)

In the following sections, each schema will be described in greater detail. The names of database objects - schemas, tables, field names - are shown in bold.

2.2 Quality and accuracy of data

Quality of data can be evaluated against a range of parameters, for example positional accuracy, attribute accuracy, logical consistency, completeness and lineage. Underlying these parameters are always the two central themes in data quality assessment: the concepts of accuracy and precision. For application to environmental data, these can be defined as follows:

– Accuracy - the degree of correctness with which a measurement reflects the true value of the parameter being assessed.
– Precision - the degree of variation in repeated measurements of the same quantity of a parameter.

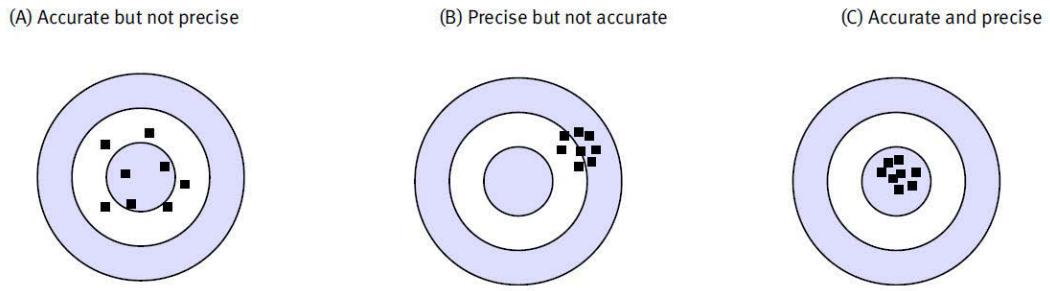

Figure 2
Depiction of the accuracy and precision of measurements.

For example, if eight measurements for a given parameter are taken at the same time at the same location using the same method, the accuracy would be indicated by how well the average of the eight measurements reflects the actual parameter value, and the precision would be indicated by the variation in the results of the eight measurements.

Using the classic example of marksmanship, Figure 2[4] distinguishes the concepts of precision and accuracy.

A high degree of precision and accuracy do not necessarily occur simultaneously in a process, as illustrated in the previous figure. Measurements may have a high degree of precision, while not being very accurate. Conversely, a set of data may have high accuracy but lack precision. When results are both precise and accurate, confidence in data quality is maximized.

In order to address these issues that may affect confidence in the WOSIS data, quality indicators are applied throughout the database: Level of Trust (subjective measure based on soil expert knowledge), Data Quality Index (objective measure based on some reproducible algorithms) and Level of Determination (Laboratory/Field/Location related uncertainty). These were developed to provide guidelines that allow investigators to recognize factors that may compromise the data quality. Use of all three components ensures that objective methods are applied for evaluating data in the database, while at the same time it enables soil expert knowledge to override these assessments.

[4] The precision and accuracy of environmental measurements for the building assessment survey and evaluation program. U.S. Environmental Protection Agency, March 31, 1999. http://http://www.epa.gov/iaq/base/pdfs/precisionandaccuracy.pdf

2.2.1 Level of trust

Different datasets/parameters held in the database need to be characterised in terms of trust. The lowest level 'A' is for data entered 'AS IS'. Subsequently, such 'A level' data can be standardized/harmonized to reference methods (Level B), (semi) - automatically error-checked for inconsistencies (Level C); the data owner is informed about the outcome of this procedure. Flagging data at level A to C can be performed automatically without any human interaction, while the final step to level D requires certain skills and experience. Level D data are those that have been approved by an expert (e.g., a soil scientist) who has performed an in-depth check, considering the value in relation to the full soil profile, the given natural conditions and the surrounding profiles and found no anomalies.

2.2.2 Data quality index

The data quality index for a given record characterises the data quality in numerical space. Any new value entered in the database initially receives a Zero value, which is the lowest rank or index value. Next, this value may increase based on whether it passes specific tests. At present, the 0-100 range is considered sufficient to characterise the data quality in numerical space. The remaining 154 values, between 101-255, are used to store specific error codes associated to specific error routines (as Domain Values in the Attribute Schema).

2.2.3 Level of determination

Any given measurement has a specific measurement error, which can be determined with a variety of methods.

The level of determination (LOD) for values derived in a laboratory can be characterised using blind samples or based on repeated measurements from reference materials. Any laboratory should be able to provide these parameters according to good laboratory practice (OECD, 1998; Van Reeuwijk, 1998).

For measurements that use other devices (e.g., GPS, soil moisture sensors, maps) the LOD can be determined by extracting information from manufacturers, repeated measurements, literature and even expert knowledge. An example for the latter is the LOD for GPS locations, which depends on several factors. Before 2000, a normal, commercial GPS had an LOD greater than 100 m; nowadays the LOD can be less than 10 m. Factors like time of day, position of satellites, land use (e.g., forest cover leads to loss of signal) will also influence the LOD for the location (e.g., Tuček and Ligoš, 2002). Lorenz et al. (2008), for example, describe factors that need to be considered to assess the LOD for field measurements, such as repeated measurements of the same property at specific, preselected random locations.

2.3 Analysis schema

The **Analysis** schema contains two tables dealing with data describing processes that have been used to obtain observation results for soil or terrain properties - essentially laboratory procedures for soil analysis.

Analytical methods are characterized by a descriptive name, and a broad description with a reference to an authoritative source (e.g., ISO standard) describing the analytical method, for instance a publication, laboratory guidelines or a web site.

The schema deploys a link table to link laboratory procedures to laboratories (in effect, their organizations).

The **Analysis** schema links to the **Reference** schema and to the **Contact** schema (i.e., it refers to data in a table in both schemas, using relational keys).

2.4 Attribute schema

Despite the broad definition of the domain of knowledge that underlies the database, every effort was made to be as accurate as possible in the definition of the entities of interest as well as their characteristics. If available, reference is made to an authoritative source for definitions and descriptions. The schema's table storing these definitions may be conceived as a soils and terrain glossary. A glossary is an (often alphabetical) list of terms in a particular domain of knowledge with the definitions for those terms.
Besides the definition of terms, this list is meant to:

- exclude ambiguity in the meaning of concepts and terms,
- promote consistency in the use of terms within the database,
- promote efficiency and precision.

To enable within-domain communication between scientists, a common language with a sufficiently shared vocabulary is needed. Such vocabularies often take the form of a controlled vocabulary, which make it easier to query a database. Since there are many different ways of describing concepts, drawing all of these terms together under a single word or phrase in a database makes searching the database more efficient as it eliminates guess work. The data in the **Attribute** schema may thus be seen as a controlled vocabulary.

In data management and database analysis, a data domain refers to all unique values which a data element may contain. The rule for determining the domain boundary may be as simple as a data type with an enumerated list of values. For example, a database table that has information about people, with one record per person, might have a 'gender' column. This gender column might be declared as a string data type, and allowed to have one of two known code values: 'M' for male, 'F' for female, and NULL for records where gender is unknown or not applicable (or arguably 'U' for unknown as a sentinel value). The data domain for the gender column is: 'M', 'F'. Other data sets with information about people, however, may employ other code values - for example, '0' for male, '1' for female, and '-1' when gender is unknown - for the same 'gender' phenomenon. Since the database should allow users to enter data in their primary form - that is, users should not be burdened with conversion issues upon entering (their) data - a mechanism to link a phenomenon to more than one data domain is required. This mechanism is the value descriptor. A value descriptor essentially links a phenomenon to a data domain. Our 'gender' example would require two value descriptor records (denoted by a value descriptor identifier) to link 'gender' to its available data domains. Conversely, a data domain may be used by more than one characteristic. For example, in the FAO Guidelines for Soil Description (FAO, 2006a), several surface characteristics use the same surface coverage classes, hence the same data domain.

In a normalized data model, the referenced data domain is typically specified in a reference table. Following the previous example, the first Gender reference table would have exactly two records, one per allowed value – 'M' and 'F', excluding NULL. The second Gender reference table would have three records, one per allowed value – '-1', '0', and '1'. In the database all potential reference tables have been merged into a single domain values table.

Reference tables are formally related to other tables in a database by the use of foreign keys. Thus, a domain value would refer to a value descriptor that links a phenomenon to the specific data domain this domain value belongs to. However, since a data domain may be referenced by more than one characteristic, the relationship between the domain values table and the value descriptor table would be of a many-to-many nature. To

circumvent this many-to-many relationship from the database, a domain table was added between the table with value descriptors and the domain values table (Figure 3).

AttributeId	AttributeName	AttributeDescription
253	Gender	a set of characteristics distinguishing between male and female, particularly in the cases of men and women (Wikipedia).

ValueDescriptorId	AttributeId	DomainId
12	253	34
13	253	65

DomainId	DomainName
34	Gender values domain according to Guidelines X
65	Gender values domain defined by organization Y

DomainValueId	DomainId	Value	Meaning
45	34	F	Female
46	34	M	Male
...			
434	65	-1	Unknown
435	65	0	Male
436	65	1	Female

Figure 3
Relationship between attribute, value descriptor, domain and domain value.

Less simple domain boundary rules, if database-enforced, may be implemented through a check constraint or, in more complex cases, in a database trigger. For example, a column requiring positive numeric values may have a check (i.e. validation) constraint declaring that the values must be greater than zero. Clearly, these attributes do not need explicitly a domain associated with them. The value descriptor may be extended with a unit and a data type for the values of its associated attribute, thus enhancing the flexibility of the Attribute schema.

The **Attribute** schema only links to the **Reference** schema.

2.5 Binary schema

Databases must store several data types like text, numbers and dates as well as binary data (e.g. columns that store binary data). The binary schema can describe various items, such as images from the soil profile or

the surrounding landscape as well as a multispectral scan. Binary data are associated to a specific mimeType (data format/characteristics).

WOSIS uses a mixed approach, storing binary data on specific local file storage and keeping a Unique Universal Identifier (UUID) identifier linking the files to the Database records. The binary table also contains a smaller version of the original binary dataset as a thumbnail.

The mixed approach allows for:
- Smaller database size, since binary data are stored in a local file system.
- Web sites will benefit from small thumbnails stored in the DB (up to 100kb per image).
- If necessary the binary data can be stored in a public accessible location.

Negative aspects:
- DB backups require that the specific local file storage is also backed-up.

The **Binary** schema links to the **Reference** and **Profile** schema.

2.6 Classification schema

Soil classification involves the systematic categorization of soils based on distinguishing characteristics as well as criteria that dictate choices in use. Soil classification is probably one of the most controversial soil science subjects. Unlike plant taxonomy, there is no truly universally accepted classification system for soil, and the principles of soil systematics have been varied and polemical (Strzeminsky, 1975). Many countries have developed their own classification systems; international correlation of the various systems is being addressed by the World Reference Base for Soil Resources (IUSS Working Group WRB 2006) and earlier through the FAO-Unesco Soil Map of the World (FAO-Unesco, 1974; FAO, 1988).

The classification schema in WOSIS supports three widely used soil classification systems:
- FAO Soil Map of the World: originally intended as legend for the Soil Map of the World, 1:5M, but in the course of time it has been used increasingly as a classification system (FAO-Unesco, 1974; FAO, 1988) now subsumed into the WRB.
- World Reference Base for Soil Resources: the international standard taxonomic soil classification system endorsed by the International Union of Soil Sciences (IUSS Working Group WRB, 2006).
- USDA Soil Taxonomy (Soil Survey Staff, 2010).

Further, the Classification schema includes a table for the storage of national or local classifications other than the global systems mentioned above.

The same soil profiles may have been described in multiple initial ISRIC data sets (e.g., in ISIS, WISE and SOTER). However, a given profile may have been classified differently in each data set, using the same soil classification system, due to differences in perceptions (see Kauffman, 1987). Therefore, the world soil database uses a link table (part of the Profile schema) to assign profiles to specific (source) data sets. All classifications refer to an entry in this link table (that is, a profile in a particular data set), thus enabling more than one classification per profile, per dataset - and even within the same data set. Figure 4 clarifies the relationship between profile, data set, link table, and classification.

At present, the classification schema permits characterization and classification of individual soil profiles. Legends for soil map units, however, are not supported yet as they are considered regular attributes of map units.

Profile

ProfileId	Country	Location
734	Uganda	POINT(33.5,1.8)

Data set (DIF)

DIF_Id	Data set name
12	WISE version 3
13	SOTER for Central Africa

Link table Profile – Data set

ProfileDIF_Id	ProfileId	DIF_Id	ProfileName
89	734	12	UG007
90	734	13	UGBun17-6

FAO Classification

FAOId	ProfileDIF_Id	SoilUnit	PublicationYear
456	89	Eutric Leptosol	1988
457	90	Lithic Leptosol	1988

Figure 4
Relationship between profile, data set, link table and classification.

Soil Taxonomy coding is inconsistent between editions as different standard notations have been used in successive versions (e.g. Soil Survey Staff, 1975, 1992, 2010). Alternatively, the original and revised Legend to the FAO Soil Map of the World (FAO-Unesco, 1974; FAO, 1988) uses a well-established coding scheme. Conversely, there is no coding scheme in WRB nor in the associated WRB Legend (IUSS Working Group, 2010), as WRB is not a hierarchical system. Therefore, to avoid any ambiguity in soil classification names, for any soil classification system full descriptive names are stored in the database, together with the edition of the system.
The **Classification** schema links to the **Profile** schema.

2.7 Contact schema

The **Contact** schema describes organizations and/or persons that have been instrumental in obtaining the observation results (either descriptive or measured) that are stored in the database. It is the single entry point to authoritative names and contact information in the overall database. This is to prevent the use of different names or spellings for the same organization or individual in various parts of the database (e.g., KIT, Tropen-Instituut, Royal Tropical Institute, Koninklijk Instituut voor de Tropen).

The main database object in this schema is the organization. There are a variety of legal types of organizations, including: corporations, governments, non-governmental organizations, international organizations, armed forces, charities, not-for-profit corporations, partnerships, cooperatives and universities. The organization object may also store organization components like departments, regional centres and laboratories.
The second database object in the **Contact** schema is a contact. A contact stores contact information for a real person. Currently, a contact can be linked to only one organization - in the sense of a 'works with' or 'is

employed by' relationship.
Access to contact information - either organizational or personal - may be controlled by way of a numerical Access Code.

The **Contact** schema links to the **Location** schema (that is, it refers to data in a table in the **Location** schema) and to the Analysis schema (used in the laboratory contact information).

2.8 DIF schema

The Directory Interchange Format (DIF)[5] is a metadata standard used to describe Earth science datasets (Global Change Master Directory Staff, 2008). The DIF standard is used to create records for the NASA Global Change Master Directory (GCMD), an on-line system with information about Earth science datasets, and is inter-operable with standards used by the ICSU World Data System (see: www.icsu-wds.org/services/data-portal), of which ISRIC is a Regular member. The DIF standard has a total of 36 elements, including eight mandatory elements. Some of the fields are text fields, while others require the use of controlled keywords. The DIF allows data users to understand the contents of a data set and contains those fields which are necessary for users to decide whether a particular data set would be useful for their needs. The DIF standard is compatible with both the ISO 19115 and CSDGM standards[6].

The soils and terrain metadata is a database implementation of the Directory Interchange Format. The DIF table is a pivotal (crucial) table in the metadata schema. It contains the eight fields that are required in the DIF. The other fields expand upon and clarify the information, but are not repetitive. Repeatable DIF fields are stored in separate, but related tables. That is, nearly all other tables are linked to the DIF table, and represent (groups of) repetitive data elements, e.g. keywords associated with a dataset. Some minor elements that are repeatable according to the DIF standard (e.g., multimedia_sample), are not repeatable in the metadata database (to keep the schema somewhat simpler).

The **DIF** schema links to the **Location** schema, the **Contact** schema, and the **Reference** schema (that is, it refers to data in a table in all of these schemas).

2.9 Location schema

Essentially, the Location schema contains authoritative names for country groups, countries, and first level administrative units within countries. As with the **Contact** schema it is the only entry point to authoritative geographic names (notably of countries) in the core database system. The reason for this is to prevent the use of different names or spellings for the same geographic location in various parts of the database where this information is required (e.g., Great Britain, United Kingdom, Royaume Uni). Names comply with ISO-3166, a standard published by the International Organization for Standardization (ISO)[7] that defines codes for the names of countries, dependent territories, and special areas of geographical interest, and their principal subdivisions (e.g., provinces or states). The official name of this standard is Codes for the representation of names of countries and their subdivisions.

---

5 http://gcmd.gsfc.nasa.gov/User/difguide/difman.html

6 The ISO 19115/TC211 geospatial metadata standard was adopted in June 2004. Required elements and appropriate modifications were approved by the CEOS IDN Interoperability group and incorporated into the DIF to achieve full ISO compatibility.

7 http://www.iso.org/iso/country_codes

Country groups enable ad-hoc grouping of countries, such as 'Former Yugoslavia' and 'ASEAN'. Country groups may be used to query the database for these groupings. A Country group may refer to an authoritative source defining the list of countries that are part of the region. In the context of the database, Country groups exceed the geographic sense of the term: it may refer to any grouping of countries, for example NATO or SADEC.

Future access methods to the data in the core database that are location-based may use web services like GeoNames[8] to enable the use of aliases, synonyms, etc. The Country table will be updated in future versions to contain polygon information (spatial data), allowing for spatial queries involving countries and profiles.
The **Location** schema only links to the **Reference** schema (i.e., it refers to data in a table in the **Reference** schema).

2.10 MapUnit schema

A map unit is a collection of areas defined and named the same in terms of their differentiating criteria, or miscellaneous areas[9]. The SOTER methodology[10], for example, identifies areas of land with a distinctive, often repetitive, pattern of landform, lithology, surface form, slope, parent material, and soil. Tracts of land distinguished in this manner are named SOTER (map) units. Each SOTER unit thus represents one unique combination of terrain and soil characteristics (Van Engelen and Dijkshoorn, 2013).
Each map unit differs in some respect from all others in the area of interest and is uniquely identified on a map. Each individual area, or polygon, on the map is a delineation of a particular map unit. Map units are represented by cartographic units with a single colour or pattern.
The database will store each map unit as a single point set union of their polygon geometries (i.e., borders), often referred to as a multi-polygon. All polygon maps, and therefore their mapping units, are stored in a single table. A map Id identifies the individual maps within the table. Every map unit must refer to a data set (defined in the **DIF** schema). For SOTER maps, a unique code (up to four digits) is assigned to every SOTER map unit that has been distinguished at the country level. Finally, a map unit may reference a source from which the data were derived for the compilation of the mapping unit.
Reference datum for any point on the Earth's surface that is referenced in the world soil database is WGS84[11].

Applying differentiating criteria in a step-by-step manner may eventually lead to overly complex and/or fragmented map units that cannot be mapped at the scale under consideration. The corresponding information is then stored as a 'non-mappable' map unit component without a geometry. Thus, in SOTER a map unit may consist of one or more map unit components, each covering a certain proportion of the map unit. Mapping unit components are sequentially numbered; largest mapping unit component first, followed by the second in size, and so on until 100% coverage is obtained.

Soil map units typically contain a number of so-called impurities, conventionally assumed to be at least 15% dependent on the scale of mapping (Landon, 1991; Soil Survey Staff, 1993). Pragmatically, a soil component is defined as a constituent of a mapping unit component that represents a single soil type (covering a certain

8 http://www.geonames.org/

9 Miscellaneous areas are areas that fall outside the differentiating criteria, e.g. water bodies under terrestrial criteria.

10 SOTER aims to establish a World Soils and Terrain Database, at scale 1:5 000 000, containing digitized map units and their attribute data in standardized format. The program was implemented by FAO, UNEP and ISRIC, under the aegis of the IUSS, in collaboration with a wide range of national soil institutes (1986 - present).

11 http://en.wikipedia.org/wiki/World_Geodetic_System

percentage of the map unit). Since a soil component is also a non-mappable area, it might be conceived as a second level map unit component, or a 'component of a map unit component'. However, introducing various component levels ('component X is a component of component Y') would make data retrieval and management overly complex, and besides, would rarely be needed. Therefore it is not implemented in the current version.

Every soil component should be described by at least one, but preferably more, fully described and analysed regionally representative profiles. One of these profiles is designated as the representative profile for the soil component, while the remaining profiles are associated with the soil component through a link table. This link table thus shows the potential range in soil variability within a given soil component.

Map unit attributes and their values are stored in a separate map unit attribute table. Attributes link to a map unit through its map unit Id. The same applies to the attributes of map unit components and soil components. For details, we refer to the **Profile** schema paragraph.

Map Unit

MapUnitId	MapId	DIF_Id	CountryId	SOTERUnitID	Location
3050	12	6	44	2446	MULTIPOLYGON(...)
3051	12	6	44	3204	MULTIPOLYGON(...)

Map Unit Component

MapUnitComponentId	MapUnitId	Number	Proportion (%)
7843	3050	1	80
7844	3050	2	20
7845	3051	1	100

Soil Component

SoilComponentId	MapUnitComponentId	Number	Proportion (%)	ProfileId
5284	7843	1	80	2974
5285	7844	1	10	2947
5286	7844	2	10	2956
2445	7845	1	100	2720

Link table Soil Component - Profile

SoilComponentProfileId	SoilComponentId	ProfileId
101356	5284	2976
101357	5284	3010
101358	5284	3011
101359	5285	2846
101360	5285	2835
101361	5286	2960
101362	5286	2969
101363	2445	2611
101364	2445	2734

Figure 5

Relation between Map unit, Map unit Component and Soil Component.

The example in Figure 5 shows two map units, with MapUnitId 3050 and 3051, respectively. Both map units are part of a SOTER map with a MapId of 12, and belong to the same data set with a DIF_Id of 6.
Map unit number 3050 consists of two map unit components:
– Map unit component 7843, covering 80% of the map unit, and
– Map unit component 7844, covering 20% of the map unit

Map unit component 7843 consists of a single soil component with Id 5284. The Id of the representative profile for this soil component is 2974. The second map unit component consists of two soil components, both covering 10% of the map unit.
Likewise, the second map unit (3051) consists of a single map unit component (7845) covering 100% of the map unit. This map unit component consists of a single soil component with Id 2445, also covering 100% of the map unit. Both map unit component and soil component are mappable, since they coincide with the map unit. Map unit components covering less than 100% of the map unit they collectively compose are not mappable since a component then only accounts for part of the map unit spatial entity.

The link table lists the Id's of other profiles that are associated with the soil components in the example.

The **MapUnit** schema links to the **Location** schema, the **Profile** schema, the **Attribute** schema, the **DIF** schema, and the **Reference** schema (that is, it refers to data in a table in all of these schemas).

2.11 Profile schema

The tables in the **Profile** schema describe two basic entities from the domain of discourse underlying the database: a profile ('pedon') and its properties (land use, position in the terrain, signs of erosion, etc.), and its constituent layer attributes (structure, colour, horizon designation, texture, pH, etc.) (Soil Survey Staff, 1975). The schema stores soil profiles, either synthetic or real-world, along with their location. To associate synthetic profiles, or profiles with an unknown location and estimated properties, with a minimal geo-reference, a country reference has been included. A 'Synthetic' identifier denotes whether a profile is synthetic (not observed in the field, 'derived') or real. Each profile also refers to the data source - publication, map, web site, etc. - from which the corresponding data were derived.
The location of a profile is stored in decimal degrees (geographic latitude and longitude, using the WGS84 datum). The accuracy of the location is stored in column positionLOD. Further, values in a column indicates if the profile can be shared with the general public; conditions for this are specified by the data providers in accordance with the ISRIC Data Policy (http://www.isric.org/data/data-policy).

Values for profile characteristics (associated with the profile's site as well as its identification) are stored in a separate attribute values table. A value descriptor links each profile attribute value to a characteristic in the **Attribute** table in the **Attribute** schema, and - if applicable - to a values domain through the Domain table (and a unit in which the value is expressed). An example may illustrate this (Figure 6); domain values are described in the **DomainValue** table in the **Attribute** schema.

For a profile with ProfileId 788, the database stores ten attribute values. The last two of these attribute values have ValueDescriptorId 281 and 282, respectively. Note that there are apparently attributes that allow for more than one value - e.g. the attributes with ValueDescriptorId's 32 and 33. ValueDescriptorId 281 refers to a property with AttributeId 18 and a values domain with DomainId 145. This property is dimensionless: there is no UnitId. Looking at the **Attribute** table, AttributeId 18 refers to the slope form of the profile site. Its value is "S". The meaning of this value can be looked up in the domain that is associated with this property: a domain with DomainId 145.The profile's attribute value eventually evaluates to a slope form of the profile site that is 'straight' (according to the 'Guidelines for profile descriptions', FAO 2006). The **Domain** table is required to prevent a many-to-many relation between the **ValueDescriptor** table and the **DomainValue** table.

Profile

ProfileId	CountryId	ReferenceId	Synthetic	Location
788	28		FALSE	POINT(26.1 -20.0)

ProfileAttribute

ProfileAttributeId	ProfileId	ValueDescriptorId	Value	ValueGroup
673	788	6	911	
1104	788	5	Central Province	
4357	788	32	WT	711
6013	788	33	1	711
6056	788	32	WG	710
6078	788	33	2	710
6155	788	260	BSh	
6617	788	19	0	
7144	788	281	S	
7488	788	282	L	

Value Descriptor

ValueDescriptorId	AttributeId	UnitId	DomainId
281	18		145
282	9		146

Attribute

AttributeId	ReferenceId	Name	Description
9	245	Position	An indication of the relative position of the site within the land.
...	...	...	...
18	245	Slope form	The slope form refers to the general shape of the slope in both the vertical and horizontal directions.

Domain

DomainId	ReferenceId	Name	Description
145	245	Position values	Each value denotes a particular position of the profile's site within the land.
146	245	Slope form values	Each value denotes a particular shape of the profile site's slope .

DomainValue

DomainValueId	DomainId	Value	Description
1850	145	V	Convex
1851	145	C	Concave
1852	145	S	Straight
...	...	...	...
1858	146	L	Lower slope
1859	146	S	Slope unspecified
1860	146	F	Flat
1861	146	V	Open depression

Figure 6

Relationships between Profile, ProfileAttribute, ValueDescriptor, Attribute, Domain and DomainValue.

Likewise, the second attribute value, the one with ValueDescriptorId 282, evaluates to a relative position of 'lower slope' of the profile within the terrain (according to the 'Guidelines for profile descriptions', FAO, 2006).

Figure 6 also illustrates the use of ValueGroups. A Valuegroup is a numeric that groups associated property values for a profile, e.g. degree and extent for two or more types of erosion at the profile site. In the example, ValueDescriptorId 32 points to 'main type of erosion at the profile site'. There are two values for erosion type: 'WT' (evaluating to "Tunnel erosion"), and 'WG' (evaluating to 'Gully erosion'). Both erosion types are associated with their matching extent (pointed to by ValueDescriptorId 33) through a common ValueGroup number. Thus, gully erosion is associated with extent '2' (evaluating to '10-25% of the area'), and tunnel erosion is associated with extent '1' (evaluating to '5-10% of the area'). By convention, the more dominant manifestation of a property has a lower ValueGroup value. Thus, in this example, gully erosion is considered dominant over tunnel erosion.

Another field, not shown in the example, may link each attribute value to a combination of a laboratory procedure and a laboratory (a link table in the Analysis schema). Either the laboratory or the procedure may be unknown. If both are unknown, the field is left blank.

All tables named '**ObjectAttribute**' use the schema **Attribute** to link attributes and their values to '**Object**', e.g. table **MapUnitAttribute** links attributes and their values to map units.

The **LayerAttribute** table describes soil profile attributes (from either synthetic or real profiles) in terms of their relative and / or absolute position within a soil profile.
The table also specifies the depth to the upper and lower boundary (in centimetres) for each layer, measured from the surface (including organic and mineral cover) of the soil downwards (FAO, 2006; Schoeneberger et al., 2012). The layerattribute table can store parameters on a Horizon basis as well as a fixed depth basis. Note also that prior to 1993, the zero datum was at the top of the mineral surface, except for thick organic layers, such as peat or muck; organic horizons were recorded as above and mineral horizons recorded as below, relative to the mineral surface (see Schoeneberger et al., 2012, pp. 2-6).

Profile descriptions may be included in more than one data set, using different identifiers (and subsets of properties). Using a link table in the **Profile** schema, data sets can be reconstructed, using the original identifiers.
Since available soil moisture plays a prevalent role in many applications, for instance crop production, it has been made a database object in its own right. Otherwise, soil moisture tensions would have to be e derived from their analytical methods, requiring arbitrary string parsing in queries.

Profiles have a specific numerical ID but also an UUID (universally unique identifier), for examples profileID 788 has an UUID of 'f0f6ae7a-13b1-11e2-8634-001bc6405f0b'. This UUID is automatically generated when a record is inserted. Using UUIDs allow for easy profile identification in diverse computer systems like harvesting environment, web services or broadcasting structures (Twitter, Facebook etc.).

The **Profile** schema links to the **Location** schema, the **Analysis** schema, the **Attribute** schema, the **DIF** schema, and the **Reference** schema (i.e., refers to data in a table in all of these schemas).

2.12 Reference schema

Data, definitions, and descriptions may be drawn from a variety of data and information sources. Potential sources are: publications, maps, web sites (URL's), organizations, and digital media. These sources vary widely in nature, and in the way they are described. The **Reference** schema enables a harmonized way to refer to these heterogeneous sources.

The **Reference** schema allows for the description of the following types of information sources:
- Publication
- Web site (URL)
- Map
- Organization
- Digital media (CD-ROM, DVD, etc.)
- Other

Except for 'Other', every reference type is represented by a separate table, either in the current schema, or in the **Contact** schema (Organization as an information source).

A reference consists of a ReferenceTypeId and a TypeReferenceId. A ReferenceTypeId refers to a reference type (publication, web site, map, etc.) in the **ReferenceType** table, and a TypeReferenceId refers to an information source in the corresponding reference type table. E.g., a ReferenceTypeId of 4 means the source of information is an organization, and the corresponding TypeReferenceId value refers to a particular organization in the **Organization** table in the **Contact** schema.

Most other schemas somehow rely on this schema. Schema **Reference** only refers to the **Organization** table in the **Contact** schema.

2.13 Internationalization schema

WOSIS is a federated system and therefore we decoupled the translation without affecting the data structure. The general assumption is that each database has a default language (e.g., in ISRIC's case English) and that the Schema named i18n (Internationalization) will contain tables linking to other schemas (e.g., profiles, attribute) that contain the translations according to language table. As a standard, we choose the W3C recommendation (RFC5646) which allows over 7000 language tags, while enabling ISO639 3 letter language codes as well as 2 letter language codes. Examples are: zh (Chinese), hi (Hindi), es (Spanish), en (English), bal (Balochi), apk (Kiowa Apache) and tpi (Tok Pisin).

The schema consists of the tables i18n.i18n which contains the generic internationalisation structure (e.g.: rfc5646 code) and schema/table/primarykey location of the translation, while the i18n.translation table stores the actual translation.

Table translation is linked to table i18n; therefore the metadata concerning translation is decoupled from the translation.

2.14 System schema

Schema **System** contains all database specific tables that support or enable a number of maintenance-related tasks:

All tables have a single-field primary key of type Integer, conventionally named 'TablenameId', with the exception of link tables. Thus, table **MapUnit** will use column MapUnitId as a single-field primary key. Link tables use a slightly different naming scheme. A link table named Name1_x_Name2 will use column Name1Name2Id for its single-field primary key. Thus, the primary key field for table **Profile_x_DIF** will be ProfileDIFId. The reason of this procedure is to keep track of primary key values to be assigned to newly created records in the database.

Table **Tally** keeps track of the primary key values to be assigned to newly created records. Upon insertion of a new record in a table, an INSERT triggers on that table:
- read the primary key value for that particular table from the **Tally** table;
- assign this value to the primary key field of the newly created record;
- increase the primary key value for that particular table in the **Tally** table with one.

Tally keeps track of the time and the last login role to insert a record in a particular database table and therefore supplying an audit trail for changes to the database. Webopedia[12] defines an audit trail as 'a record showing who has accessed a computer system and what operations he or she has performed during a given period of time.' Table **Revision** stores an audit trail for changes to WOSIS. The table traces every UPDATE and DELETE operation to any record in the database (with the exception of System tables). The database operation is stored in a column named Action: 'U' for a record update and 'D' for a deleted record. Four more columns pinpoint the record affected by the operation. The table also registers the login role that executed the operation, together with a timestamp for the operation and the version number of the affected record. The (original) values of the affected records are stored in tables **ChangeGeometry** (geometry columns) and **ChangeItem** (all other columns). The following example (Figure 7) may illustrate this.

The last Revision record (with RevisionId 3467) shows that a record with a value of 50 for primary key field 'CountryId' in table **Country** in schema **Location** has been updated. There is also the Id of the previous revision of this record: 3466. The last revision affected field 'Alpha_2': the value 'DC' of version 2 of the record was changed. The previous revision of the record, version 1, affected two fields: 'Name' and 'Alpha_2'. The version number for the corresponding record in the database is 3. Who (Editor) and when (Time) have been left out in the **Revision** table in this example.

[12] http://www.webopedia.com/TERM/A/audit_trail.html

Country

CountryId	Numeric	Alpha_3	Alpha_2	Name	RevisionId	Version
50	180	COD	CD	Congo, Dem. Republic of	3467	3

Revision

RevisionId	Action	Schema Name	Table Name	PK_Name	PK_Value	Previous Revision	Version
3465	U	Reference	Publication	PublicationId	121		1
3466	U	Location	Country	CountryId	50		1
3467	U	Location	Country	CountryId	50	3466	2

ChangeItem

ChangeItemId	RevisionId	ColumnName	Value
657	3465	Publisher	Elsevier B.V.
658	3466	Alpha_2	ZR
659	3466	Name	Zaire
660	3467	Alpha_2	DC

Figure 7
Relations between Revision and ChangeItem.

2.15 Other schema uses

Another important use for schemas in the world soil database is to mimic a federated database system. This is a type of meta-database management system (DBMS) which transparently integrates multiple autonomous database systems into a single virtual database. The constituent databases may be interconnected via a computer network and geographically decentralized. Since the constituent database systems remain autonomous, a federated database system is an alternative to the (sometimes daunting) task of merging together several disparate databases.
A federated database approach prevents replication of master data within other parts of the organization. Master data are often used by several functional groups and stored in different data systems across an organization and may or may not be referenced centrally; therefore, the possibility exists for duplicate and/or inaccurate master data ('Master data' lemma, Wikipedia). For master data, there should be an agreed upon view across the organization.

For example, in an ideal situation the **Contact** schema would not be implemented. The core database would link up with an actively maintained and up-to-date external Customer Relationship Management (CRM) system that would be used for other purposes as well.
The **Analysis** schema might evolve into a database subsystem with standardized descriptions of analytical procedures that is more easily query-able than the current way of describing these procedures. This will make it easier to link the proper procedure to an observation. Furthermore, the link will be more descriptive.
A library catalogue is by far a more appropriate register of bibliographic items than the **Reference** schema of the world soil database.

3 Future developments

WOSIS forms an integral part of the emerging Global Soil Information Facilities (GSIF, see: http://www.isric.org/projects/global-soil-information-facilities-gsif; Batjes et al., 2013; Omuto et al., 2012).
Future expansion, and testing, of WOSIS is seen as an iterative process that is largely governed by the:
- addition of 'new' soil data held in datasets emanating from ISRIC collaborative, international projects,
- addition of soil data from auxiliary sources; this will include spectral data.

An inevitable consequence of aforementioned activities is that new soil properties will be added as appropriate, for soil descriptions guidelines not yet accommodated in version 1.0 of WOSIS.
Instrumental to enhanced usability and accessibility of the data in the world soil database are the following three activities:
- soil property harmonization
- soil property values (i.e., domain) harmonization
- standardization of analytical procedure descriptions - as part of the metadata

When implemented at a satisfactory level of detail and authority, these activities will add value to the database as a queryable source of:
- soil terms
- analytical procedures for soil properties

The content of the world soil database can be handled using standardized interfaces that are based on XML/GML exchange models - notably those defined by the Open GeoSpatial Consortium (OGC). Use of these interfaces will allow for the fulfilment of future demands for global soil information, and enable incorporation of soil data held by third parties.
GeoServer[13] will be used to serve geospatial data from the database - profile locations, SOTER maps, GlobalSoilMap.net products, etc. GeoServer (an open source software server) is the reference implementation of the Open Geospatial Consortium's (OGC) Web Feature Service (WFS) and Web Coverage Service (WCS) standards, as well as a high performance certified compliant Web Map Service (WMS).
Spatial resources in the database will be referenced in an ISRIC GeoNetwork catalog implementation.
NASA's Directory Interchange Format (DIF) will remain the metadata standard for describing ISRIC datasets, unless a new format is adopted by the ICSU World Data System.

[13] GeoServer is an open source software server written in Java that allows users to share and edit geospatial data. Designed for interoperability, it publishes data from any major spatial data source using open standards. See http://www.geoserver.org

A stack of open RESTful[14] web services (serving xml/json, in analogy with, for example, GeoNames[15] web services) is to be specified. Further, (exchange) present and future data models (e.g., e-SOTER, SoilML and GeoSciML etc.) will be supported for data export and import.
Procedures will be developed whereby data held in WOSIS may be provided in an interoperable way for use in Global Soil Partnership projects and deliveries, GlobalSoilMap.net and other applications such as WISE, e-SOTER and similar.

[14] Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web (Fielding RT, 2005). A REST web service (also called a RESTful web API) is a simple web service implemented using HTTP and the principles of REST. It is a collection of resources, with three defined aspects:

- the base URI for the web service, such as http://example.com/resources/
- the Internet media type of the data supported by the web service. This is often JSON, XML or YAML but can be any other valid Internet media type.
- the set of operations supported by the web service using HTTP methods (e.g., POST, GET, PUT or DELETE).
  Unlike SOAP-based web services, there is no 'official' standard for RESTful web services. This is because REST is an architecture, unlike SOAP, which is a protocol. Even though REST is not a standard, a RESTful implementation such as the Web can use standards like HTTP, URI, XML, etc.

[15] http://www.geonames.org

Acknowledgements

Starting point for the world soil database model was a preliminary design by Daniël van Kraalingen of Alterra, Wageningen University and Research centre, as part of an institutional strategy for redesign and improvement of the ISRIC ICT infrastructure ('Advies ICT infrastructuur ISRIC'; March, 2010). Database development was supervised by a steering group that was chaired by Niels Batjes (2010) respectively Hannes I. Reuter (2011-2013). Invaluable comments and suggestions from members of the steering group (Niels Batjes, Koos Dijkshoorn, Vincent van Engelen, Johan Leenaars, Bob MacMillan and Hannes I. Reuter) as well as Prem Bindraban, Daniel van Kraalingen and Rob Lookers have been incorporated in the model in an iterative process. Special thanks are expressed to the external reviewers from University of Nottingham (United Kingdom), Commonwealth Scientific and Industrial Research Organisation (CSIRO, Australia) and the Federal Institute for Geosciences and Natural Resources (BGR, Germany) for their constructive comments.

References

Batjes NH 2009. Harmonized soil profile data for applications at global and continental scales: updates to the WISE database. Soil Use and Management 25, 124-127

Batjes NH, Reuter HI, Tempel P, Leenaars JGB, Hengl T and Bindraban PS 2013. Development of global soil information facilities. Data Science Journal 12, 70-74 p http://dx.doi.org/10.2481/dsj.WDS-009

CGI 2013. CGI I Technical Collaboration I GeoSciML downloads. GeoSciML home page. Viewed April 2013; http://www.cgi-iugs.org/tech_collaboration/geosciml.html

FAO 1988. FAO-Unesco Soil Map of the World, Revised Legend, with corrections and updates. World Soil Resources Report 60, FAO, Rome; reprinted with updates as Technical Paper 20 by ISRIC, Wageningen, 1997, pp. 140.

FAO 2006a. Guidelines for soil description (Fourth ed.). FAO, Rome (Available at: ftp://ftp.fao.org/agl/agll/docs/guidel_soil_descr.pdf), pp. 97

FAO 2006b. World Reference Base for soil resources - A framework for international classification, correlation and communication. World Soil Resources Reports 103, International Union of Soil Sciences, ISRIC - World Soil Information and Food and Agriculture Organization of the United Nations, Rome, pp. 128

FAO-Unesco 1974. Soil Map of the World, 1:5,000,000. Vol. 1 – Legend. United Nations Educational, Scientific, and Cultural Organization, Paris

Fielding, RT, Taylor, RN 2005 Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology (TOIT) (New York: Association for Computing Machinery) 2 (2): 115–150 (2002-05).

Global Change Master Directory staff 2008. Directory Interchange Format (DIF) Writer's Guide. Global Change Master Directory. National Aeronautics and Space Administration. Viewed April 2013; http://gcmd.nasa.gov/User/difguide/difman.html

ISIS 2006. ISRIC Soil Information System, ver. 6.0, ISRIC - World Soil Information http://isis.isric.nl/

ISRIC 2009. ISIS Soil Information System. ISIS data query page. Viewed May 2013; http://isis.isric.org

ISRIC 2010. ISRIC Annual Higlights 2009. ISRIC – World Soil Information, Wagenin g. http://www.isric.org/sites/default/files/ISRIC_Annual_Highlights_2009.pdf

ISRIC 2013. ISRIC - World Soil Information. Soil and Terrain Database (SOTER) Programme. Viewed April 2013; http://www.isric.org/projects/soil-and-terrain-database-soter-programme

IUSS Working Group WRB 2006. World Reference Base for Soil Resources (2nd ed.). World Soil Resources Report 103, FAO, Rome, 145 p http://www.fao.org/ag/agl/agll/wrb/doc/wrb2006final.pdf

IUSS Working Group WRB 2010. Guidelines for constructing small-scale map legends using the World Reference Base for Soil Resources. Addendum to the World Reference Base for Soil Resources. World Soil Resources Report 103, FAO, Rome (Available at: http://www.fao.org/fileadmin/templates/nr/images/resources/pdf_documents/WRB_Legend.pdf), 26 p http://www.fao.org/fileadmin/templates/nr/images/resources/pdf_documents/WRB_Legend.pdf

Kauffman, JH 1987. Comparative classification of some deep, well-drained red clay soils of Mozambique. Technical Paper 16, International Soil Reference and Information Centre (ISRIC), pp. 68. Wageningen. Viewed April 2013; http://www.isric.org/isric/webdocs/Docs/ISRIC_TechPap16.pdf

Landon JR 1991. Booker Tropical Soil Manual. Longman Scientific & Technical, New York, 474 p

Lorenz, K, R. Lal and J. Shipitalo, 2008. Chemical stabilization of organic carbon pools in particle size fractions in no-till and meadow soils. Bio. Fertility Soil, 44: 1043-1051.

Montanarella, L, Wilson P, Cox S, McBratney A, Ahamed S, McMillan B, Jacquier D, Fortner J. 2010. Developing SoilML as a Global Standard for the Collation and Transfer of Soil Data and Information Geophysical Research Abstracts, Vol. 12. EGU General Assembly, 2010-3180

NASA 2013. Earth Science data and services directory: Global Change Master Directory Web Site. Home page. Viewed April 2013; http://gcmd.nasa.gov

NASA 2013. Global Change Master Directory Web Site. Find data sets by topic. Viewed May 2013; http://gcmd.nasa.gov/KeywordSearch/Home.do?Portal=GCMD&MetadataType=0

OECD, 1998 OECD Principles of Good Laboratory Practice (as revised in 1997). OECD Environmental Health and Safety Publications (OECD) 1. 1998

OGC 2013. Welcome to the OGC Website I OGC(R). Viewed April 2013; http://www.opengeospatial.org

Omuto C, Nachtergaele F and Vargas Rojas R 2012. State of the Art Report on Global and Regional Soil Information: Where are we? Where to go?, FAO, Italy, 69 p http://typo3.fao.org/fileadmin/templates/nr/kagera/Documents/Soil_information_Report.pdf

Pourabdollah, A, Leibovici, DG, Simms, DM, Tempel, P, Hallett, SH and Jackson, MJ 2012. Towards a standard for soil and terrain data exchange: SoTerML Computers and Geosciences. http://dx.doi.org/10.1016/j.cageo.2011.11.026.

Sanchez PA, Ahamed S, Carre F, Hartemink AE, Hempel J, Huising J, Lagacherie P, McBratney AB, McKenzie NJ, Mendonca-Santos MdL, Minasny B, Montanarella L, Okoth P, Palm CA, Sachs JD, Shepherd KD, Vagen T-G, Vanlauwe B, Walsh MG, Winowiecki LA and Zhang G-L 2009. Digital Soil Map of the World. Science 325, 680-681, see also http://www.globalsoilmap.net

Schoeneberger PJ, Wysocki DA, EC Benham and Soil Survey Staff 2012. Field book for describing and sampling soils (ver. 3.0). National Soil Survey Center Natural Resources Conservation Service, U.S. Department of Agriculture, Lincoln (NE)

Soil Survey Staff 1975. Soil Taxonomy: A basic system of soil classification for making and interpreting soil surveys. Soil Conservation Service, USDA Agriculture Handbook 436, USDA, Washington

Soil Survey Staff 1983. Soil Survey Manual (rev. ed.). United States Agriculture Handbook 18, USDA, Washington

Soil Survey Staff 1992. Keys to Soil Taxonomy (5th. ed.), SMSS Technical Monograph No. 19. Agency for International Development, United States Department of Agriculture, Soil Conservation Service, Soil Management Support Services. Pocahontas Press, Inc., Blacksburg (Virginia)

Soil Survey Staff 1998. Keys to Soil Taxonomy (8th. ed.). United States Department of Agriculture, Natural Resources Conservation Service, Blacksburg (Virginia), 326 p

Soil Survey Staff 2003. Keys to Soil Taxonomy (9th. ed.). United States Department of Agriculture, Natural Resources Conservation Service, Blacksburg (Virginia), 332 p

Soil Survey Staff 2010. Keys to Soil Taxonomy, 11th ed. USDA-Natural Resources Conservation Service, Washington, DC. (ftp://ftp-fc.sc.egov.usda.gov/NSSC/Soil_Taxonomy/keys/2010_Keys_to_Soil_Taxonom.pdf)

Strzeminsky 1975. Ideas Underlying Soil Systematics. Foreign Scientific Publication, Department of the National Centre for Scientific, Technical and Economic Information, Warsaw, Poland.

Tuček, J, Ligoš, J, 2002. Forest canopy influence on the precision of location with GPS receivers. Journal of Forest Science, 48, 2002 (9): 399–407

Van Engelen VWP 2011. Standardizing soil data (e-SOTER regional pilot platform as EU contribution to a Global Soil Information System). International Innovation June, 48-49

van Engelen VWP and Dijkshoorn JA 2013. Global and National Soils and Terrain Digital Databases (SOTER) - Procedures manual (Ver. 2.0). ISRIC Report 2013/04, IUSS, ISRIC and FAO, Wageningen, 191 p http://www.isric.org/sites/default/files/ISRIC_Report_2012-04_TOTAL_LR-SEC.pdf

van Engelen, VWP 2013. e-SOTER project. Viewed April 20131: http://www.esoter.net

van Reeuwijk LP 1998. Guidelines for quality management in soil and plant laboratories. FAO Soil Bulletin 74, ISRIC and FAO, Rome, 143 p http://www.fao.org/docrep/W7295E/W7295E00.htm

van Waveren, EJ, Bos, AB 1994 Guidelines for the description and coding of soil data. ISRIC Technical Paper No. 14, Revised edition.

Acronyms

eSOTER	Regional pilot platform as EU contribution to a Global Soil Observing System
FAO	Food and Agriculture Organization of the United Nations
GML	Geography Markup Language
ISIS	ISRIC Soil Information System
ISRIC	ISRIC - World Soil Information; formerly: International Soil Reference and Information Centre
IUSS	International Union of Soil Science
OGC	Open Geospatial Consortium
SoilML	Soil Markup Language
SOTER	Soil and Terrain database programme
UNEP	United Nations Environmental Program
USDA	United States Department of Agriculture
WDC	World Data Center of the ICSU World Data System (ICSU-WDS)
WISE	World Inventory of Soil Emission potentials (database)
WOSIS	World Soil Information System
XML	Extensible Markup Language

Appendix 1
Database computational and personal capacity

Every database is in need for time and qualified personal on appropriate resources (e.g. computers, network connections). If these are not provided, performances might be not sufficient for specific tasks. High availability issues (e.g. Standby server, Master-Slave replication or load balancing) are not discussed as situations for deployments might be different.

At the time of publication we recommend in terms of computing resources as minimum:

CPU: WOSIS has been running on a 1 core machine, but for productivity purposes 4 or more cores are recommended.

Memory: WOSIS has been running on a 1024 MB system, but for productivity purposes 6GB or more are recommended.

Hard drive Space: WOSIS has been running with 100GB hard drive space, but for productivity purposes 400GB or more are recommended. Especially with storing binary data sets (these are stored directly on disk), file/disk sizes need to be adjusted appropriately. For outmost performance the database system itself should be stored on a Solid State Drive (SDD), while the binary data can be stored on a SATA 6 or 10 drives or similar.

An incremental daily backup of the system/database should be the minimum stored at a different system than the production machine, with a monthly full backup stored at a third location.

In terms of personal at least one database administrator needs to allocate sufficient times to maintain, monitor and diagnosis the Database; investigate performance issues and similar. This can be between a 25% position up to a 2 times full positions depending on the capabilities of the person(s), the data loaded into the system and the requests issued against the database.

Appendix 2
Database design

Database modelling and database design was performed in Enterprise Architect® from Sparx Systems using the UML Data Modelling Profile. This profile provides easy-to-use and easy-to-understand extensions to the UML standard, mapping the database concepts of tables and relationships onto the UML concepts of Classes and associations.

Data Model Detail

This Appendix provides an overview of the data model.

ISRIC Soil Data Repository

Type: *Package*
Package: *Model*
Detail: *Created on 17-11-2010. Last modified on 20-12-2012.*

Data Model

Created By: *Piet Tempel on 22-11-2005*
Last Modified: *20-12-2012,* Version:1.2

Overview
Database: *<none>, Stereotype: ,* Package: ISRIC Soil Data Repository
Detail: *Created on 18-3-2011. Last modified on 18-3-2011.*
Notes: *All packages depend on 'System'.*

See Figure 8 on next page.

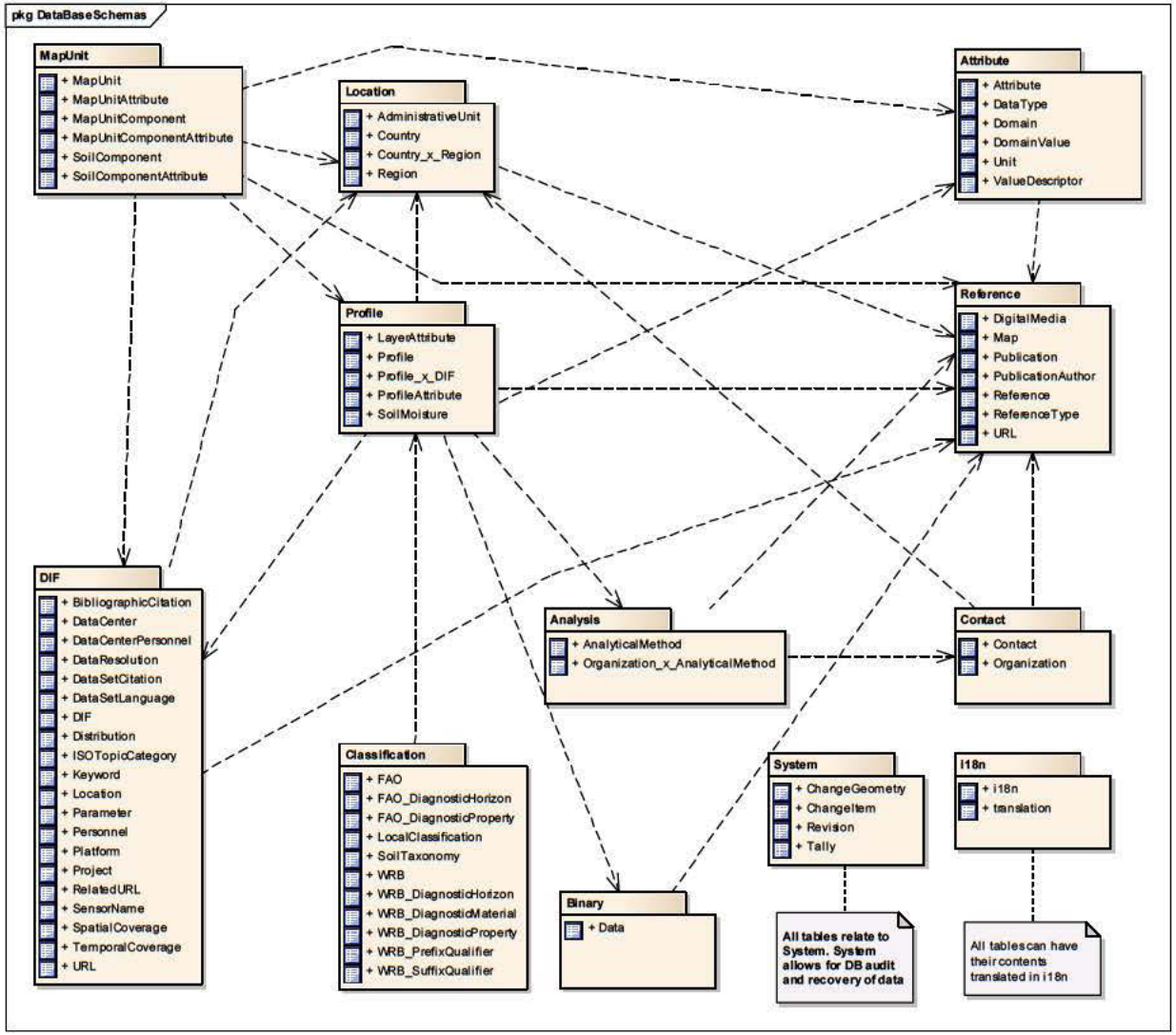

Figure 8
ISRIC Soil Data repository diagram.

Analysis

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 5-1-2006. Last modified on 28-3-2011.*
Notes: This schema comprises all those relations that deal with data describing the processes that have been used to attain an observation result (procedures, in OGC O&M parlance). These processes are essentially laboratory procedures for soil analysis. There is no globally accepted standard to describe these procedures, but in most laboratory manuals procedure descriptions are more or less arranged as follows:

- Principle: description of the theory underlying the procedure. If necessary, it outlines how to calculate the final result.
- Equipment: a listing of the apparatus that is used in the procedure.
- Reagents: a listing of the chemical substances or compounds used in the procedure.
- Procedure: a step-by-step description of the procedure. The steps in the procedure may be subdivided into groups denoting a distinct phase in the procedure. Notes may be used to denote constraints, special conditions, etc.
- References: to sources with details regarding the described procedure.

The schema currently contains two relations see .Figure 9.

Tables in this schema describe Analytical procedures - predominantly laboratory procedures - for soil analysis, and link procedures to laboratories (organizations) that use them.

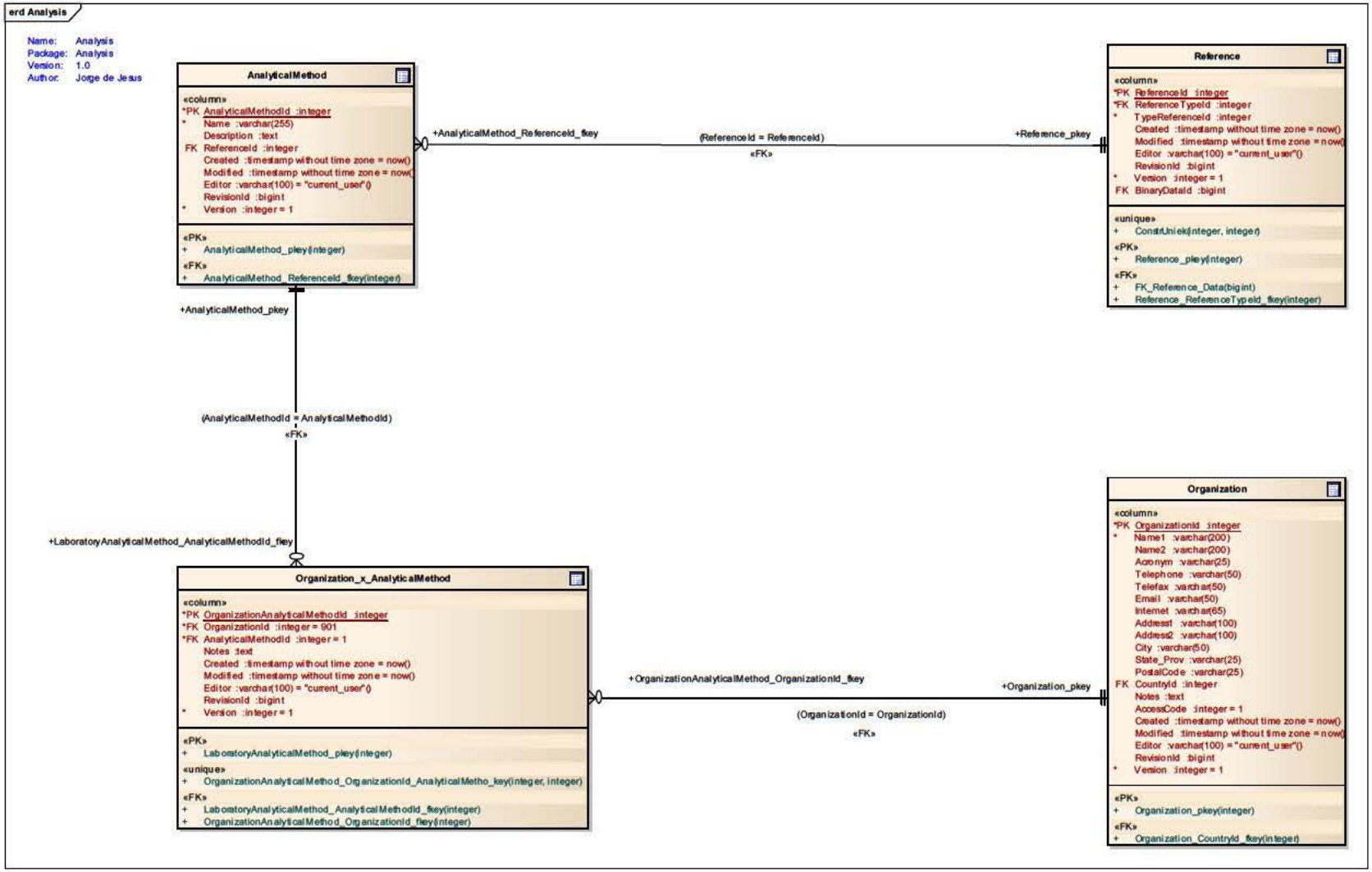

Figure 9
Analysis diagram.

AnalyticalMethod

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Analysis
Detail: *Created on* 17-11-2010. *Last modified on* 20-12-2011.
Notes: Descriptions of analytical determination methods for (predominantly soil) properties like pH, cation exchange capacity, soluble salts, etc. Each named method is amply described and - if available - has a reference to an authoritative source.
Methods may apply to more than one property - e.g. the hydrometer method for various particle-size classes. To avoid redundant repetition of methods, no direct link with properties (in table *Attribute.Attribute*) has been made.
The very first analytical method (with *AnalyticalMethodId* = 1) is the 'unknown' method.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	AnalyticalMethodId	Bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Name	varchar	True	False	255		Descriptive name for the analytical method.
False	Description	Text	False	False			An ample description of the analytical method.
False	ReferenceId	Bigint	False	False			Reference to an entry in the Reference.Reference table - a publication, web site, organization, etc.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	Bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
AnalyticalMethod_pkey	Public	AnalyticalMethodId		
AnalyticalMethod_ReferenceId_fkey	Public	ReferenceId		

Relationships

Columns	Association		Notes
(AnalyticalMethodId = AnalyticalMethodId)	**0..***	**Organization_x_AnalyticalMethod**.LaboratoryAnalyticalMethod_AnalyticalMethodId_fkey	
	1	**AnalyticalMethod**.AnalyticalMethod_pkey	
	0..*	**AnalyticalMethod**.**AnalyticalMethod_ReferenceId_fkey**	
	1	**Reference**.Reference_pkey	

Organization_x_AnalyticalMethod

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Analysis
Detail: *Created on* 17-11-2010. *Last modified on 23*-12-2012.
Notes: The *Organization_x_AnalyticalMethod* table links organizations (in effect, laboratories - listed in the *Contact.Organization* table) to analytical procedures in the *AnalyticalMethod* table. This link table thus lists all analytical methods used by an organization (past and present).
The *Notes* field allows for information on an organization-specific implementation of a particular method or procedure.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	OrganizationAnalyticalMethodId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	OrganizationId	integer	True	True		901	Reference to an entry in the Organization table that represents an organization. Default organization is Unknown (901).
False	AnalyticalMethodId	integer	True	True		1	Reference to an analytical procedure in the AnalyticalMethod table. Default analytical method is Unknown (1).
False	Notes	text	False	False			Additional information - e.g. application specifics of the procedure by the Organization (laboratory).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
LaboratoryAnalyticalMethod_pkey	Public	OrganizationAnalyticalMethodId		
OrganizationAnalyticalMethod_OrganizationId_fkey	Public	OrganizationId		
OrganizationAnalyticalMethod_OrganizationId_AnalyticalMetho_key	Public	OrganizationId AnalyticalMethodId		

Relationships

Columns	Association	
(AnalyticalMethodId = AnalyticalMethodId)	**0..***	**Organization_x_AnalyticalMethod.LaboratoryAnalyticalMethod_AnalyticalMethodId_fkey**
	1	**AnalyticalMethod.**AnalyticalMethod_pkey
(OrganizationAnalyticalMethodId=OrganizationAnalyticalMethodId)	**0..***	**ProfileAttribute**.FK_ProfileAttribute_LaboratoryAnalyticalMethod
	1	**Organization_x_AnalyticalMethod.OrganizationAnalyticalMethodId**
	0..*	**LayerAttribute**.FK_HorizonAttribute_LaboratoryAnalyticalMethod
	1	**Organization_x_AnalyticalMethod**.LaboratoryAnalyticalMethod_pkey
	0..*	**SoilMoisture.**FK_SoilMoisture_OrganizationAnalyticalMethod
	1	**Organization_x_AnalyticalMethod.**LaboratoryAnalyticalMethod_pkey
(OrganizationId=OrganizationId)	**0..***	**Organization_x_AnalyticalMethod**.OrganizationAnalyticalMethod_OrganizationId_fkey
	1	**Organization**.Organization_pkey

Attribute

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 14-12-2010. Last modified on 28-3-2011.*
Notes: The foremost role of this schema is that of a controlled vocabulary. Controlled vocabularies make databases easier to search. Since there are many different ways of describing concepts, drawing all of these terms together under a single word or phrase in a database makes searching the database more efficient as it eliminates guess work. However, arriving at this efficiency requires consistency on the part of the individual indexing the database and the use of pre-determined terms.
Conducting a search in a database that uses controlled vocabulary or indexing terms is efficient and precise. The biggest advantage to controlled vocabulary is that once the user finds the correct term, most of the information he needs is grouped together in one place, saving the time of having to search under all of the other synonyms for that term. In the future, this controlled vocabulary may evolve into a full-fledged *ontology.*

See Figure 10.

Attribute

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Attribute
Detail: *Created on* 14-12-2010. *Last modified on* 24-3-2011.
Notes: The attribute table defines soil, site, and terrain characteristics which have received wide acceptance amongst surveyors in the description of soils and terrain. Each named characteristic is amply described and - if available - has a reference to an authoritative source. Is an authoritative source available, then the characteristic is described in accordance with its source of supply.
Characteristics may be aspects of the same phenomenon; e.g. type, degree, and extent are all aspects of erosion. Some characteristics may be similar in concept, but differ in their scale of application, e.g. 'slope' as a site and terrain characteristic. These scale differences have been honoured in the definition of characteristics.

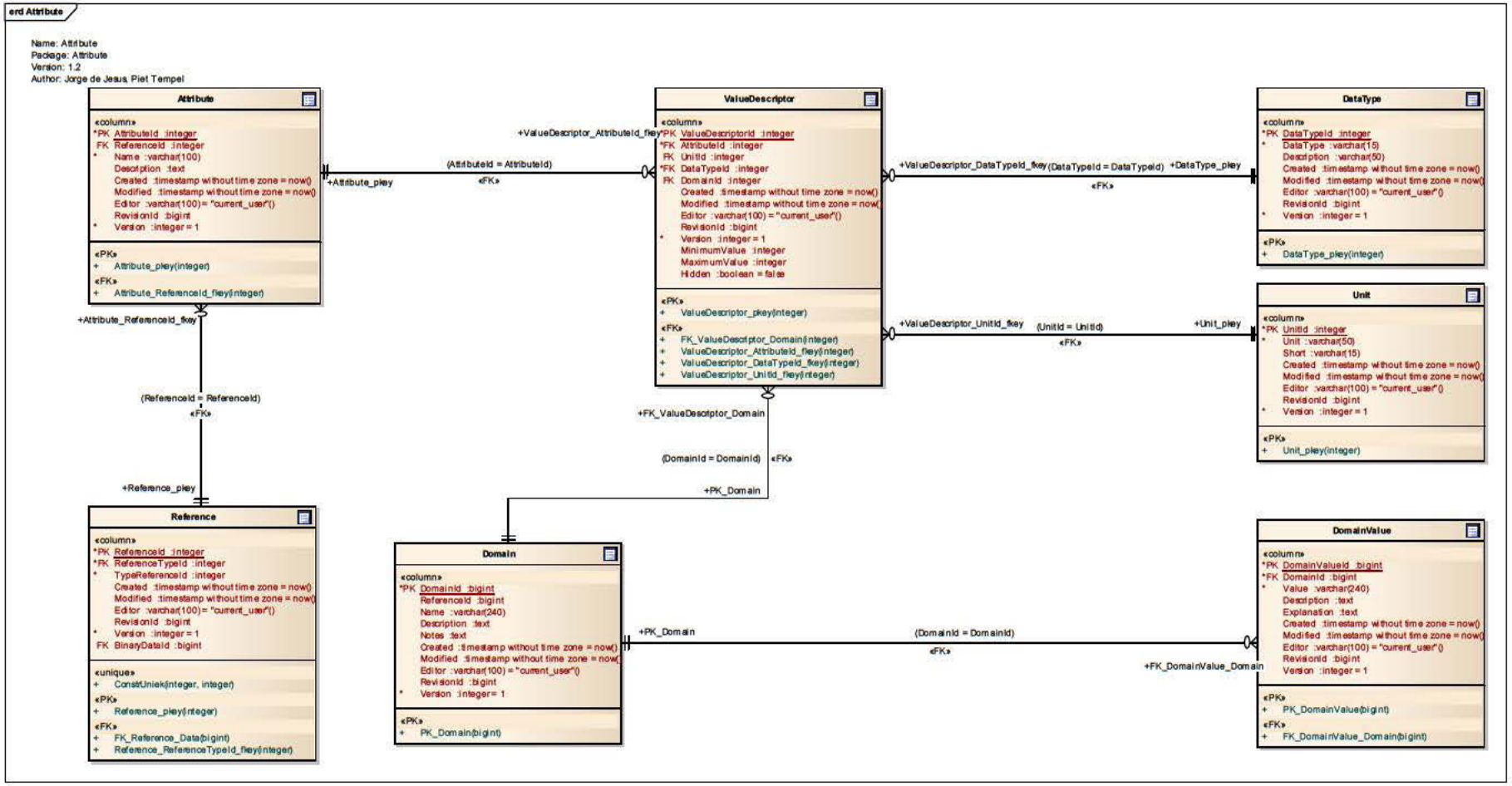

Figure 10
Attribute diagram.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	AttributeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ReferenceId	integer	False	False			Reference to an entry in the Reference.Reference table - a publication, web site, organization, etc.
False	Name	varchar	True	False	100		Apt name for the property, preferably in accordance with its source of supply (see referenceId).
False	Description	text	False	False			Description of the property, preferably in accordance with its source of supply (see referenceId).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Attribute_ReferenceId_fkey	Public	ReferenceId		

Relationships

Columns	Association		Notes
(AttributeId = AttributeId)	**0..***	**ValueDescriptor**.ValueDescriptor_AttributeId_fkey	
	1	**Attribute**.Attribute_pkey	
	0..*	**Attribute**.Attribute_ReferenceId_fkey	
	1	**Reference**.Reference_pkey	

DataType

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Attribute
Detail: *Created on* 14-12-2010. *Last modified on* 24-3-2011.
Notes: The *DataType* table identifies the various types of data in the database, such as floating-point, integer, or Boolean, that determines the possible values for that type; the operations that can be done on that type; and the way the values of that type are stored.
Data types are linked to ValueDescriptors. The latter describe the way that soil, site and terrain characteristics are expressed.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DataTypeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	DataType	varchar	True	True	15		Name of the data type - integer, string, etc.
False	Description	text	False	False			Description of the data type.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for initial v.

Constraints

Name	Type	Columns	Initial Code	Notes
DataType_pkey	Public	DataTypeId		

Relationships

Columns	Association		Notes
(DataTypeId = DataTypeId)	**0..***	**ValueDescriptor**.ValueDescriptor_DataTypeId_fkey	
	1	**DataType**.DataType_pkey	
	1	**ValueDescriptor**.ValueDescriptor_DataTypeId_fkey	
	1	**Data.**FK_Data_DataType	

Domain

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Attribute
Detail: *Created on* 14-12-2010. *Last modified on* 24-3-2011.
Notes: The *Domain* table lists all data domains that are available for the attributes in the *Attribute* table. A data domain refers to all the unique values which a data element (attribute) may contain. An example of a data domain is the set of allowable values for the site characteristic 'vegetation', in accordance with an authoritative source (referenced by the *ReferenceId* field). Domains are linked to ValueDescriptors. The latter describe the way that soil, site and terrain characteristics are expressed. Obviously, domains are also linked to allowable domain values.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DomainId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ReferenceId	bigint	False	False			Reference to an entry in the Reference. Reference table - a publication, web site, organization, etc. that is the source of supply for the domain.
False	Name	varchar	True	False	240		Domain name, preferably in accordance with its source of supply (see referenceId).
False	Description	text	False	False			Description of the domain, preferably in accordance with its source of supply (see referenceId).
False	Notes	text	False	False			Additional remarks.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System. Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_Domain	Public	DomainId		

Relationships

Columns	Association	Notes
(DomainId = DomainId)	**0..*** **ValueDescriptor**.FK_ValueDescriptor_Domain **1** **Domain**.PK_Domain	
	0..* **Domain**.FK_Domain_Reference **1** **Reference**.Reference_pkey	
(DomainId = DomainId)	**0..*** **DomainValue**.FK_DomainValue_Domain **1** **Domain**.PK_Domain	

DomainValue

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Attribute
Detail: *Created on* 7-2-2011. *Last modified on* 24-3-2011.
Notes: The *DomainValue* table describes per domain (defined in the *Domain* table) all unique values which a site, soil, or terrain characteristic may contain. Thus, domain values are linked to domains, and domains are linked to ValueDescriptors. The latter describe the way that soil, site and terrain characteristics are expressed.
An example of a data domain is the set of allowable values for the site characteristic "land use", in accordance with an authoritative source (referenced by the *ReferenceId* field in the *Domain* table).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DomainValueId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	DomainId	bigint	True	True			Reference to a domain in the *Domain* table.
False	Value	varchar	True	True	240		Admissible value in a domain.
False	Description	text	False	False			Description (often decoding) of the domain value.
False	Explanation	text	False	False			Explanation of the domain value - elaboration of the description.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_DomainValue	Public	DomainValueId		

Relationships

Columns	Association	Notes
(DomainId = DomainId)	**0..*** **DomainValue**.FK_DomainValue_Domain **1** **Domain**.PK_Domain	

Unit

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Attribute
Detail: *Created on* 14-12-2010. *Last modified on* 24-3-2011.
Notes: A unit of measurement is a definite magnitude of a physical quantity, defined and adopted by convention and/or by law, that is used as a standard for measurement of the same physical quantity. Any other value of the physical quantity can be expressed as a simple multiple of the unit of measurement. In physics, units are standards for measurement of physical quantities that need clear definitions to be useful. Reproducibility of experimental results is central to the scientific method. A standard system of units facilitates this.
The *Units* table defines the units used for measurement of soil, site, and terrain characteristics.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	UnitId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Unit	varchar	True	True	50		Unit - full description.
False	Short	varchar	True	False	15		Unit - short description.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Unit_pkey	Public	UnitId		

Relationships

Columns	Association	Notes
(UnitId = UnitId)	**0..*** **ValueDescriptor.ValueDescriptor_UnitId_fkey** **1** **Unit**.Unit_pkey	

ValueDescriptor

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Attribute
Detail: *Created on* 14-12-2010. *Last modified on* 24-3-2011.
Notes: ValueDescriptors describe the way that soil, site and terrain characteristics are expressed. The *ValueDescriptor* table links soil, site, and terrain characteristic (defined in table *Attribute*) to value domains (defined in table *Domain*), data types (defined in table *DataType*), and units (defined in table *Unit*).
As an example, terrain characteristic 'land form' may be described according to various (local) systems (or 'domains', with allowable values). ValueDescriptors link aforementioned characteristic to these different domains.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ValueDescriptorId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	AttributeId	integer	True	False			Reference to a property in the *Attribute* table.
False	UnitId	integer	False	False			Reference to a unit in the *Unit* table - that is, the unit in which the property value is expressed.
False	DataTypeId	integer	True	False			Reference to a data type in the *DataType* table - string, integer boolean, etc.
False	DomainId	integer	False	False			Reference to a domain in the *Domain* table - that is, the set of valid values.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	Bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
ValueDescriptor_pkey	Public	ValueDescriptorId		
ValueDescriptor_AttributeId_fkey	Public	AttributeId		
FK_ValueDescriptor_Domain	Public	DomainId		
ValueDescriptor_UnitId_fkey	Public	UnitId		

Relationships

Columns	Association	Notes
(AttributeId = AttributeId)	**0..*** **ValueDescriptor**.ValueDescriptor_AttributeId_fkey **1** **Attribute**.Attribute_pkey	
(UnitId = UnitId)	**0..*** **ValueDescriptor**.ValueDescriptor_UnitId_fkey **1** **Unit**.Unit_pkey	
(DataTypeId = DataTypeId)	**0..*** **ValueDescriptor**.ValueDescriptor_DataTypeId_fkey **1** **DataType**.DataType_pkey	
(DomainId = DomainId)	**0..*** **ValueDescriptor**.FK_ValueDescriptor_Domain **1** **Domain**.PK_Domain	
	0..* **LayerAttribute**.FK_HorizonAttribute_ValueDescriptor **1** **ValueDescriptor**.ValueDescriptor_pkey	
(ValueDescriptorId=ValueDescriptorId)	**0..*** **ProfileAttribute**.FK_ProfileAttribute_ValueDescriptor **1** **ValueDescriptor**.ValueDescriptor_pkey	
	0..* **SoilComponentAttribute**.SoilComponentAttribute_ValueDescriptorId_fkey **1** **ValueDescriptor**.ValueDescriptor_pkey	
	0..* **MapUnitComponentAttribute**.TerrainComponentAttribute_ValueDescriptorId_fkey **1** **ValueDescriptor**.ValueDescriptor_pkey	
	0..* **MapUnitAttribute**.MapUnitAttribute_ValueDescriptorId_fkey **1** **ValueDescriptor**.ValueDescriptor_pkey	

Classification

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 24-11-2010. Last modified on 28-3-2011.*
Notes: Soil classification deals with the systematic categorization of soils based on distinguishing characteristics as well as criteria that dictate choices in use. The *Classification* schema only permits the characterization and classification of individual soil profiles, soil map legends for map units are not supported (yet).
Three major classification systems are supported by this schema:

- USDA Soil Taxonomy: classifies soils at six levels or categories based on diagnostic soil horizons and soil climatic conditions. The broadest category is Order. Lower categories, in which classes are successively more narrowly defined, are the Suborder, Great Group, and Subgroup. Table *SoilTaxonomy* stores Great Group, Subgroup, Texture Class, and Mineralogy.
- FAO Soil Map of the World: this system was originally intended only as a legend for the Soil Map of the World, 1:5M, and was originally published between 1974 and the early 1980's, along with the map sheets. Since then it has become more of a classification system and in fact is now subsumed into the WRB. There have been two versions. The most recent is the 1988 Revised Legend. Diagnostic horizons and diagnostic properties are stored in separate tables.
- World Reference Base for Soil Resources (WRB): the international, standard taxonomic soil classification system endorsed by the International Union of Soil Sciences (IUSS). It was developed by an international collaboration coordinated by the International Soil Reference and Information Centre (ISRIC), the IUSS and the FAO via its Land & Water Development division. WRB supercedes the previous FAO Soil Legend. Diagnostic horizon, diagnostic property, diagnostic material, and the prefix and suffix qualifier are stored in separate tables, since they are repetitive elements within a WRB classification.

Also, the local (or national) classification for a profile can be stored in this schema.
For the same soil classification system, different datasets may show different classifications for a profile. Therefore, reference to a classified profile is indirect, through a *ProfileDlFld*. A *ProfileDlFld* points to a profile in a particular dataset (in table *Profile.Profile_x_DlF and*), thus enabling more than one classification per profile, per dataset - and even within the same data set. See Figure 11.

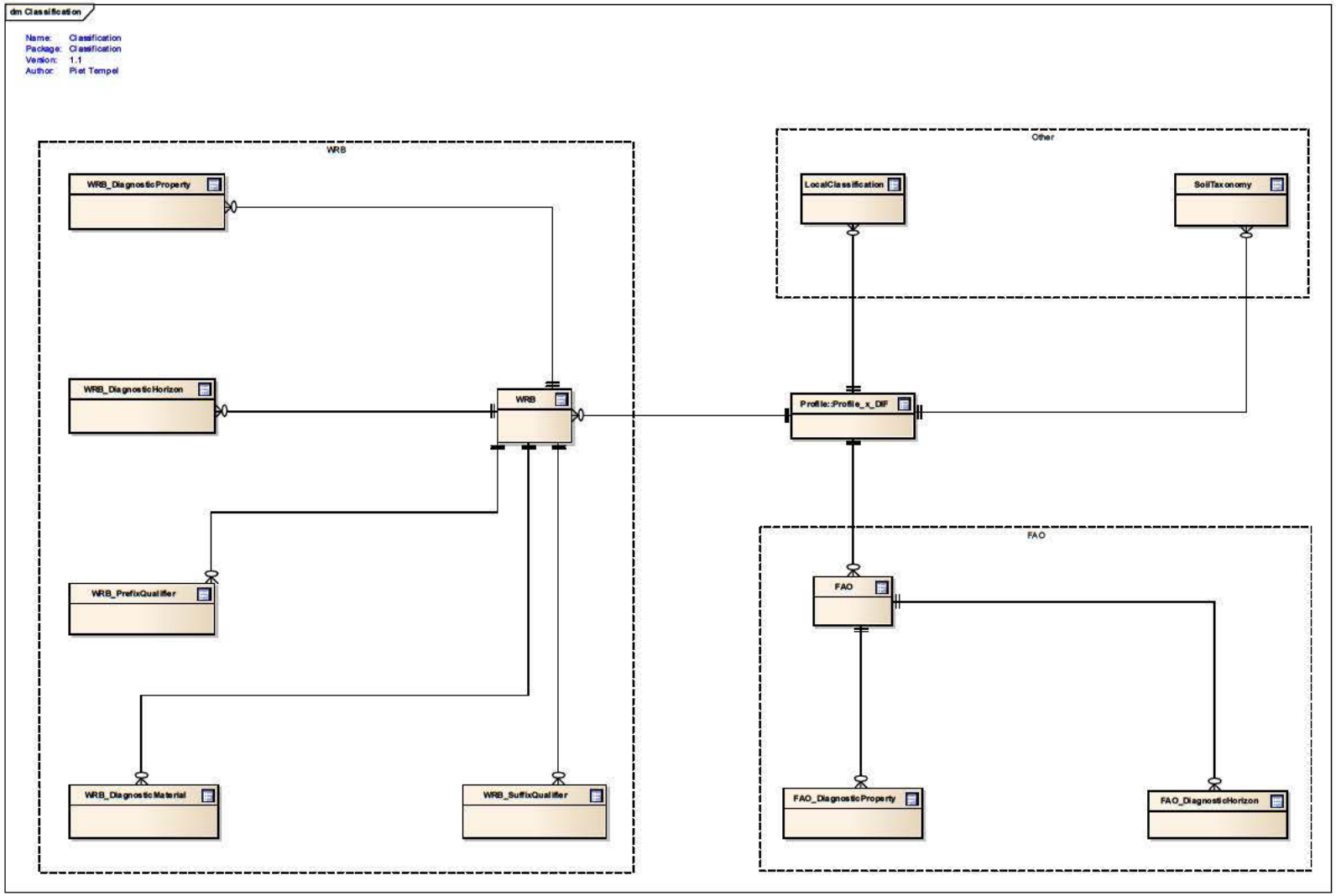

Figure 11
Classification schema diagram.

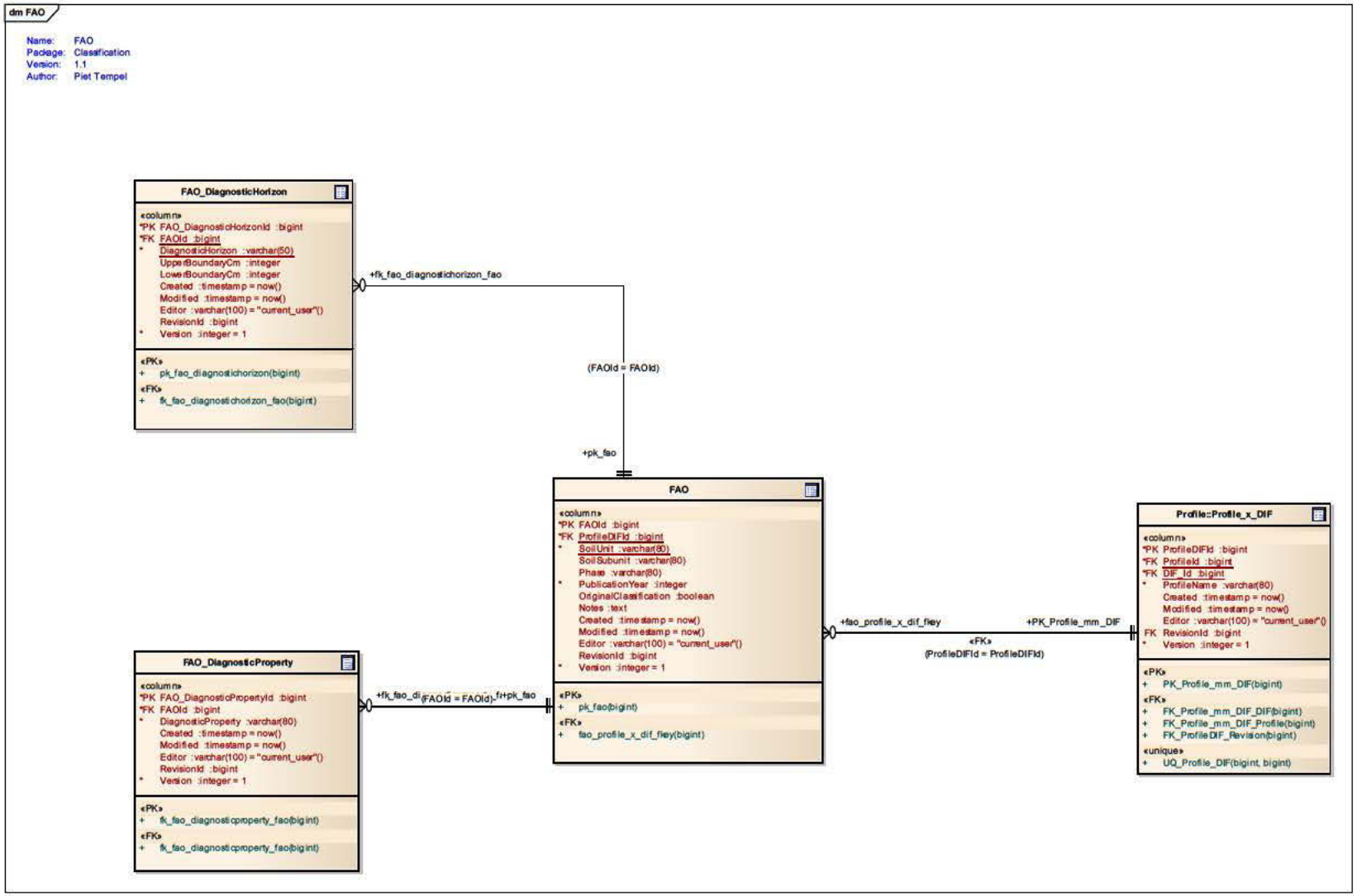

Figure 12
FAO classification diagram.

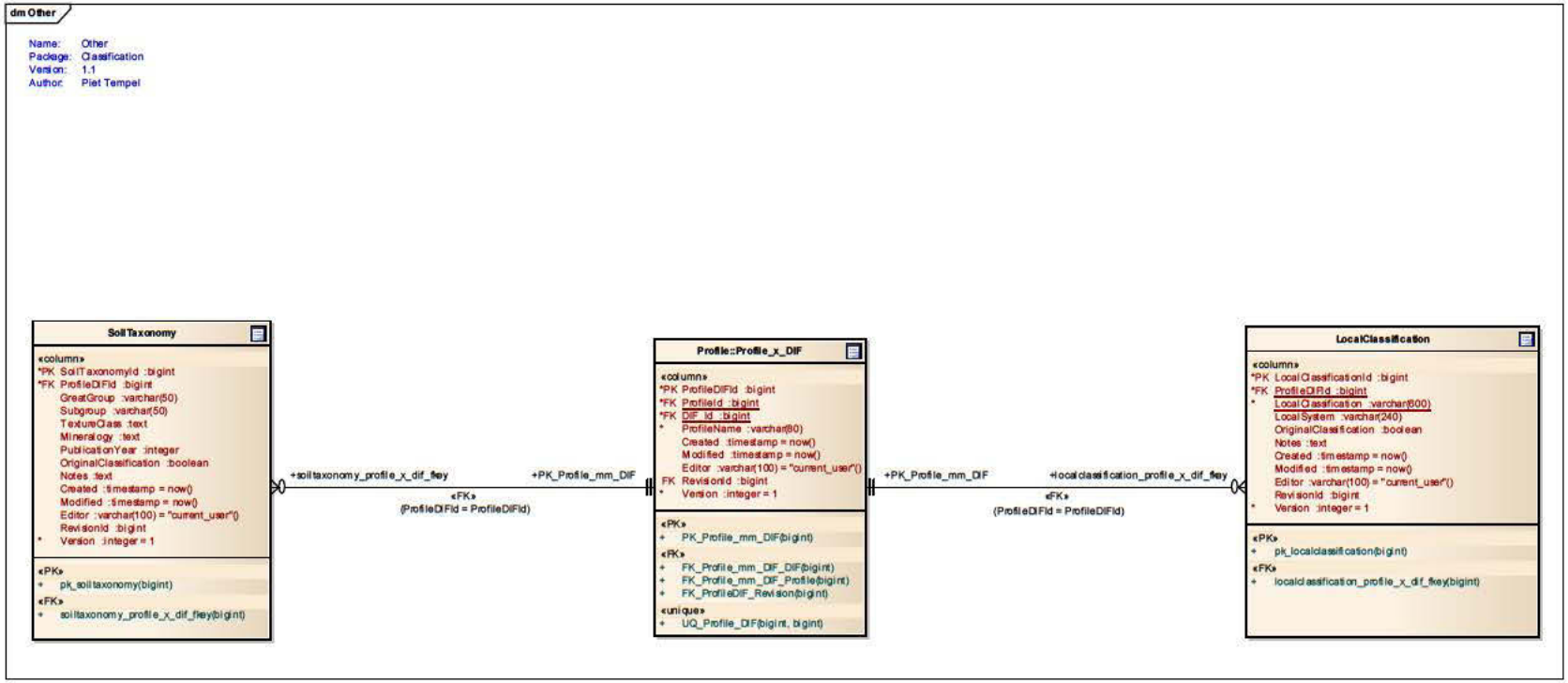

Figure 13

Soil Taxonomy and Local Classification diagram.

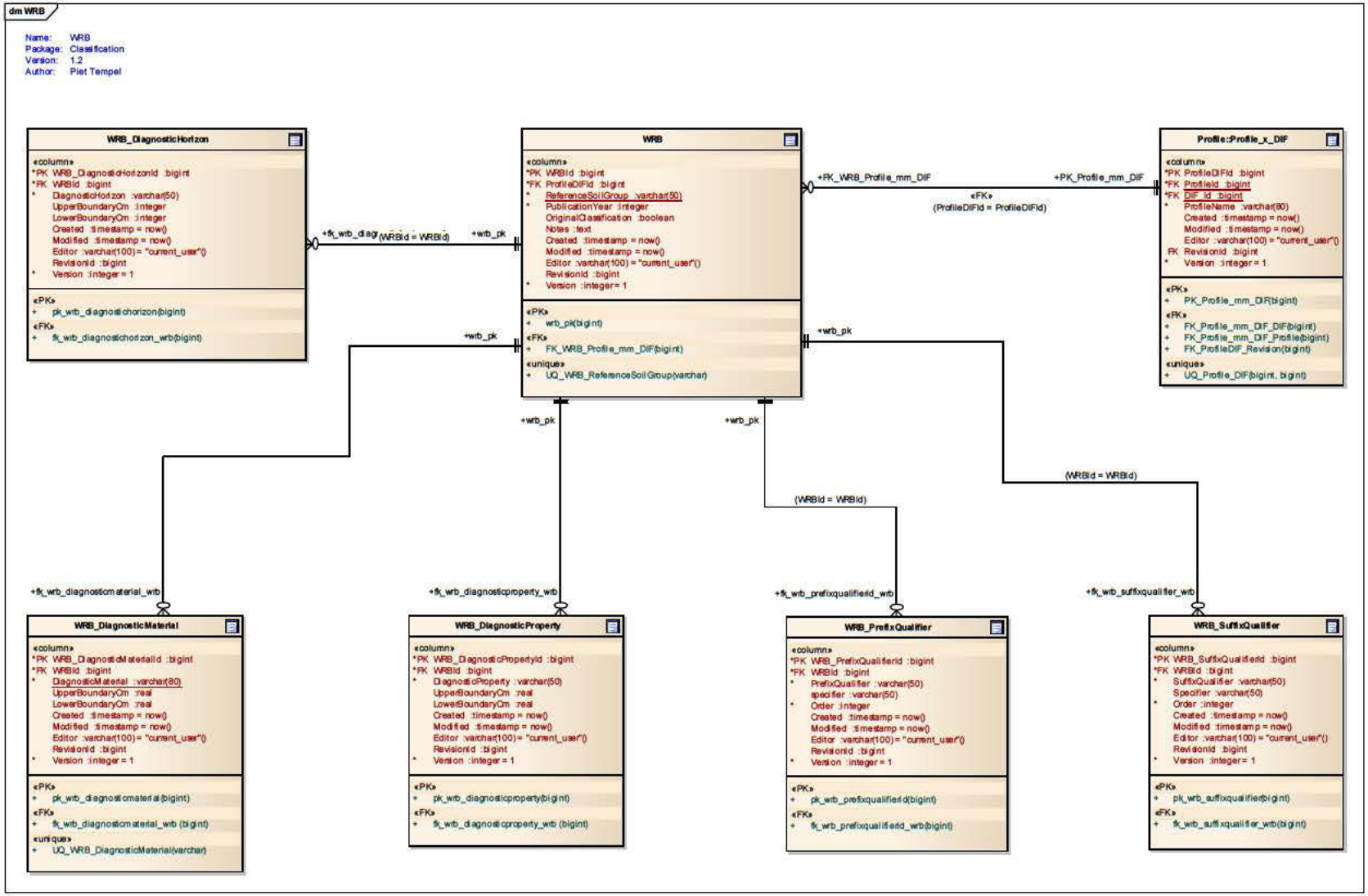

Figure 14

WRB classification diagram.

FAO

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification

Detail: *Created on* 24-11-2010. *Last modified on* 24-3-2011.

Notes: The Food and Agriculture Organization of the United Nations (FAO) developed a supra-national classification, also called World Soil Classification, which offers useful generalizations about soils pedogenesis in relation to the interactions with the main soil-forming factors. It was first published in form of the UNESCO Soil Map of the World (1974) at scale 1: 5 M. Many of the names offered in that classification are known in many countries and do have similar meanings. Originally developed as a legend to the Soil Map of the World, Soil Units were mapped as Soil Associations, designated by the dominant soil unit.

The FAO soil map is intended for mapping soils at a continental scale but not at local scale. In 1998, this system was replaced by the World Reference Base for Soil Resources.

Table *FAO* stores this FAO World Soil Classification for any soil profile in the database - both real-world and artificial. Since different datasets may show different FAO World Soil Classifications for the same profile, reference to a profile is indirect, through a *ProfileDIFId*. A *ProfileDIFId* points to a profile in a particular dataset (in table *Profile.Profile_x_DIF*), thus enabling more than one FAO World Soil Classification per profile, per dataset (see discussion in Kauffman, 1985).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	FAOId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileDIFId	bigint	True	True			Foreign key that refers to a soil profile in a particular dataset in the Profile_x_DIFtable.
False	SoilUnit	varchar	True	True	50		FAO Soil unit name.
False	SoilSubunit	varchar	False	False	50		FAO Soil subunit name.
False	Phase	varchar	False	False	50		FAO Phase - limiting factor related to (sub) surface features of the land.
False	PublicationYear	integer	True	False			The year of publication of the version of the FAO Legend used for the characterization.
False	OriginalClassification	boolean	False	False			The profile has been described originally (e.g. in the field or when made the map, value is TRUE), or has been classified later (which normally includes tacit suppositions, value is FALSE).
False	Notes	text	False	False			Additional information.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

pk_fao	Public	FAOId		
fao_profile_x_dif_fkey	Public	ProfileDIFId		

Relationships

Columns	Association	Notes
(ProfileId=ProfileId)	**0..*** **FAO_DiagnosticProperty**.fk_fao_diagnosticproperty_fao **1** **FAO**.pk_fao	
	0..* **FAO_DiagnosticHorizon**.fk_fao_diagnostichorizon_fao **1** **FAO**.pk_fao	
	0..* **FAO**.fao_profile_x_dif_fkey **1** **Profile_x_DIF.**Profile_x_DIF_pkey	

FAO_DiagnosticHorizon

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 24-3-2011.
Notes: Table *FAO_DiagnosticHorizon* lists FAO Legend diagnostic horizons, identifying soil units - a set of quantitatively defined properties defined by soil forming processes. The diagnostic horizon is part of a FAO World Soil Classification in table *FAO* (referred to by *FAOId*). There may be more than one diagnostic horizon per FAO World Soil Classification.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	FAO_DiagnosticHorizonId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	FAOId	bigint	True	True			Reference to an entry in the FAO table that represents an FAO Legend soil classification.
False	DiagnosticHorizon	varchar	True	True	50		Name of the diagnostic horizon.
False	UpperBoundaryCm	real	False	False			Depth to upper diagnostic horizon boundary (centimetres).
False	LowerBoundaryCm	real	False	False			Depth to lower diagnostic horizon boundary (centimetres).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_fao_diagnostichorizon	Public	FAO_DiagnosticHorizonId		
fk_fao_diagnostichorizon_fao	Public	FAOId		

Relationships

Columns	Association	Notes
(FAOId=FAOId)	**0..*** **FAO_DiagnosticHorizon**.fk_fao_diagnostichorizon_fao **1** **FAO**.pk_fao	

FAO_DiagnosticProperty

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 25-3-2011.
Notes: Table *FAO_DiagnosticProperty* lists FAO World Soil Classification diagnostic properties. Diagnostic properties are soil characteristics which do not constitute distinct horizons, but which are of importance for classification purposes.
A diagnostic property is part of a FAO World Soil Classification in table *FAO* (referred to by *FAOId*). There may be more than one diagnostic property per FAO World Soil Classification.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	FAO_DiagnosticPropertyId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	FAOId	bigint	True	False			Reference to an entry in the FAO table that represents an FAO Legend soil classification.
False	DiagnosticProperty	varchar	True	False	80		Name of the diagnostic property.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
fk_fao_diagnosticproperty_fao	Public	FAOId		
uq_faoid_diagnosticproperty	Public	PublicFAOId , DiagnosticProperty		

Relationships

Columns	Association	Notes
(FAOId=FAOId)	**0..*** **FAO_DiagnosticProperty**.fk_fao_diagnosticproperty_fao **1** **FAO**.pk_fao	

LocalClassification

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 24-3-2011.
Notes: Soil is classified into categories in order to understand relationships between different soils and to determine the usefulness of a soil for a particular use. The first soil classification systems were based on the idea that soils have a particular morphology based on the materials and factors that form them. In the 1960s, a different classification system began to emerge, that focused on soil morphology instead of parental materials and soil-forming factors. Since then soil classification has undergone further modifications.

Table *LocalClassification* classifies soils - either real-world or artificial - according to a local or national classification system other than the FAO World Soil Classification, USDA Soil Taxonomy, or the World Reference Base for Soil Resources (WRB). Since different datasets may show different local classifications for the same profile, reference to a profile is indirect, through a *ProfileDIFId*. A *ProfileDIFId* points to a profile in a particular dataset (in table *Profile.Profile_x_DIF*), thus enabling more than one local classification per profile, per dataset.
The LocalSystem field allows for the specification of the local soil classification system that has been used.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	LocalClassificationId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileDIFId	bigint	True	True			Foreign key that refers to a soil profile in a particular dataset"Profile"."Profile_x_DIF"
False	LocalClassification	varchar	True	True	600		Soil classification according to a local or national system of soil classification.
False	LocalSystem	varchar	False	False	600		Name of the local or national soil classification system.
False	OriginalClassification	boolean	False	False			The profile has been described originally (e.g. in the field or when made the map, value is TRUE), or has been classified later (which normally includes tacit suppositions, value is FALSE).
False	Notes	text	False	False			Additional information.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_localclassification	Public	LocalClassificationId		
localclassification_profile_x_dif_fkey	Public	ProfileDIFId		

Relationships

Columns	Association	Notes
(ProfileDIFId = ProfileDIFId)	**0..*** **LocalClassification**.FK_LocalClassification_Profile_x_DIF **1** **Profile_x_DIF**.Profile_x_DIF_pkey	

SoilTaxonomy

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 24-3-2011.
Notes: USDA Soil Taxonomy developed by United States Department of Agriculture and the National Cooperative Soil Survey provides an elaborate classification of soil types according to several parameters (most commonly their properties) and in several levels: Order, Suborder, Great Group, Subgroup, Family, and Series.

Table ***SoilTaxonomy*** stores this FAO World Soil Classification for any soil profile in the database - both real-world and artificial. Since different datasets may show different Soil Taxonomy classifications for the same profile, reference to a profile is indirect, through a *ProfileDIFId*. A *ProfileDIFId* points to a profile in a particular dataset (in Table 'Profile'.'Profile_x_DIF'), thus enabling more than one Soil Taxonomy classification per profile, per dataset.

NB. Although part of Soil Taxonomy, Soil Moisture Regime and Soil Temperature Regime are considered soil characteristics in their own right in WRB. Both characteristics occur as such, for example, in the latest edition of the FAO Guidelines for Soil Description (2006).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	SoilTaxonomyId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileDIFId	bigint	True	False			Foreign key that refers to a soil profile in a particular dataset "Profile"."Profile_x_DIF"
False	GreatGroup	varchar	False	False	50		USDA Soil Taxonomy Great Group.
False	Subgroup	varchar	False	False	50		USDA Soil Taxonomy Subgroup.
False	TextureClass	text	False	False			USDA Soil Taxonomy texture class.
False	Mineralogy	text	False	False			USDA Soil Taxonomy mineralogy class.
False	PublicationYear	integer	False	False			The year of publication of the version of the Soil Taxonomy key used for the characterization.
False	OriginalClassification	boolean	False	False			The profile has been described originally (e.g. in the field or when made the map, value is TRUE), or has been classified later (which normally includes tacit suppositions, value is FALSE).
False	Notes	text	False	False			Additional information.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_soiltaxonomy	Public	SoilTaxonomyId		
soiltaxonomy_profile_x_dif_fkey	Public	ProfileDIFId		

Relationships

Columns	Association	Notes
(ProfileDIFId = ProfileDIFId)	**0..*** **SoilTaxonomy**.soiltaxonomy_profile_x_dif_fkey **1** **Profile_x_DIF**.Profile_x_DIF_pkey	

WRB

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 25-3-2011.
Notes: The World Reference Base for Soil Resources (WRB) is the international standard taxonomic soil classification system endorsed by the International Union of Soil Sciences (IUSS).
The WRB borrows heavily from modern soil classification concepts, including USDA soil taxonomy, the legend for the FAO Soil Map of the World 1988, the Référentiel Pédologique and Russian concepts. The classification is based mainly on soil morphology as an expression pedogenesis. A major difference with USDA soil taxonomy is that soil climate is not part of the system, except insofar as climate influences soil profile characteristics. As far as possible, diagnostic criteria match those of existing systems, so that correlation with national and previous international systems is as straightforward as possible.
The WRB is meant for correlation of national and local systems. The level of detail corresponds to USDA soil taxonomy subgroups, without the soil climate information. It is not detailed enough for mapping at scales larger than about 1:200k, although proposal have been made to couple WRB with substrate information to map at 1:50k in regional studies.
Table *WRB* stores the fully-named WRB Reference Soil Group for any soil profile in the database - both real-world and artificial. Since different datasets may show different WRB classifications for the same profile, reference to a profile is indirect, through a *ProfileDIFId*. A *ProfileDIFId* points to a profile in a particular dataset (in table *Profile.'Profile'.'Profile_x_DIF'*), thus enabling more than one WRB classification per profile, per dataset. Every WRB classification includes the year of publication of the WRB version used for the characterization.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	WRBId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileDIFId	bigint	True	False			Foreign key that refers to a soil profile in a particular dataset in the "Profile"."Profile_x_DIF"table.
False	ReferenceSoilGroup	varchar	True	True	50		WRB Reference Soil Group
False	PublicationYear	integer	True	False			The year of publication of the WRB version used for the characterization.
False	OriginalClassification	boolean	False	False			The profile has been described originally (e.g. in the field or when made the map, value is TRUE), or has been classified later (which normally includes tacit suppositions, value is FALSE).
False	Notes	text	False	False			Additional information.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.

PK	Name	Type	Not Null	Unique	Len	Init	Notes
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_wrb	Public	WRBId		
FK_WRB_Profile_x_DIF	Public	ProfileDIFId		

Relationships

Columns	Association	Notes
(ProfileDIFId = ProfileDIFId)	**0..*** **WRB**.FK_WRB_Profile_x_DIF **1** **Profile_x_DIF**.PK_WRB	
	0..* **WRB_DiagnosticProperty**.fk_wrb_diagnosticproperty_wrb **1** **WRB**.uq_profile_publicationyear	
	0..* **WRB_DiagnosticHorizon**.fk_wrb_diagnostichorizon_wrb **1** **WRB**.uq_profile_publicationyear	
	0..* **WRB_PrefixQualifier**.fk_wrb_prefixqualifierid_wrb **1** **WRB**.uq_profile_publicationyear	
	0..* **WRB_DiagnosticMaterial**.fk_wrb_diagnosticmaterial_wrb **1** **WRB**.uq_profile_publicationyear	
	0..* **WRB_SuffixQualifier**.fk_wrb_suffixqualifier_wrb **1** **WRB**.uq_profile_publicationyear	

WRB_DiagnosticHorizon

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 25-3-2011.
Notes: In the WRB, horizons are named diagnostic when they possess a minimum degree of soil development. This is defined by:

- Visibility
- Prominence
- Measurability
- Importance and relevance for soil formation and soil usage (that is, reference to bio-Climatological factors)

A WRB diagnostic horizon is part of a WRB soil classification in table *WRB* (referred to by *WRBId*). There may be more than one diagnostic horizon per WRB soil classification.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	WRB_DiagnosticHorizonId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	WRBId	bigint	True	False			Reference to an entry in the WRB table that represents a WRB soil classification.
False	DiagnosticHorizon	varchar	True	False	50		Name of the WRB diagnostic horizon.
False	UpperBoundaryCm	real	False	False			Depth to upper diagnostic horizon boundary (centimetres).
False	LowerBoundaryCm	real	False	False			Depth to lower diagnostic horizon boundary (centimetres).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - 1 for the initial en.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_wrb_diagnostichorizon	Public	WRB_DiagnosticHorizonId		
fk_wrb_diagnostichorizon_wrb	Public	WRBId		

Relationships

Columns	Association	Notes
	0..* **WRB_DiagnosticHorizon**.fk_wrb_diagnostichorizon_wrb **1** **WRB**.pk_wrb	

WRB_DiagnosticMaterial

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 25-3-2011.
Notes: In the WRB, diagnostic material reflects:

- Original parent material
- Pedogenetic processes have not yet been so active that they left a significant mark
- Comprise anthropogenic, calcaric, fluvic, gypsic, organic, sulfidic, tephric

WRB diagnostic material is part of a WRB soil classification in table *WRB* (referred to by *WRBId*). There may be more than one diagnostic horizon per WRB soil classification.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	WRB_DiagnosticMaterialId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	WRBId	bigint	True	False			Reference to an entry in the WRB table that represents a WRB soil classification.
False	DiagnosticMaterial	varchar	True	True	50		Name of the WRB diagnostic material.
False	UpperBoundaryCm	real	False	False			Depth to upper diagnostic horizon boundary (centimetres).
False	LowerBoundaryCm	real	False	False			Depth to lower diagnostic horizon boundary (centimetres).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_wrb_diagnosticmaterial	Public	WRB_DiagnosticMaterialId		
uq_wrbid_diagnosticmaterial	Public	WRBId, DiagnosticMaterial		

Relationships

Columns	Association	Notes
	0..* **WRB_DiagnosticMaterial**.fk_wrb_diagnosticmaterial_wrb **1** **WRB**.pk_wrb	

WRB_DiagnosticProperty

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 25-3-2011.
Notes: In WRB, diagnostic soil properties comprise an assemblage of several soil characteristics and reflect present or past mechanism of soil formation. Soil characteristics are:

- Sole parameters (laboratory, field, microscopy)
- Characteristics such as texture, soil depth, voids, mottles, cutans, nodules
- Analytical determination like pH, cation exchange capacity, soluble salts

A WRB diagnostic property is part of a WRB soil classification in table *WRB* (referred to by *WRBId*). There may be more than one diagnostic property per WRB soil classification.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	WRB_DiagnosticPropertyId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	WRBId	bigint	True	False			Reference to an entry in the WRB table that represents a WRB soil classification.
False	DiagnosticProperty	varchar	True	False	50		Name of the WRB diagnostic property.
False	UpperBoundaryCm	real	False	False			Depth to upper diagnostic horizon boundary (centimetres).
False	LowerBoundaryCm	real	False	False			Depth to lower diagnostic horizon boundary (centimetres).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_wrb_diagnosticproperty	Public	WRB_DiagnosticPropertyId		
k_wrb_diagnosticproperty_wrb	Public	WRBId		

Relationships

Columns	Association	Notes
	0..* **WRB_DiagnosticProperty**.fk_wrb_diagnosticproperty_wrb **1** **WRB**.pk_wrb	

WRB_PrefixQualifier

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 29-3-2011.
Notes: Qualifiers are meant to complete the soil description through Reference soil groups (RSG). These qualifiers are either prefixes or suffixes. For each RSG, there are several possible prefixes and suffixes. When using qualifiers, prefixes are listed before the RSG.

A WRB prefix qualifier is part of a WRB soil classification in table *WRB* (referred to by *WRBId*) There may be more than one (order-relevant) prefix qualifier per WRB soil classification.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	WRB_PrefixQualifierId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	WRBId	bigint	True	False			Reference to an entry in the WRB table that represents a WRB soil classification.
False	PrefixQualifier	varchar	True	False	50		Prefix qualifiers are typically associated qualifiers and intergrade qualifiers.
False	Specifier	varchar	False	False	50		Specifiers such as Epi-, Endo-, Hyper-, Hypo-, Thapto-, Bathy-, Para-, Proto-, Cumuli- and Ortho- are used to indicate a certain expression of the qualifier.
False	Order	integer	True	False			The sequence of the intergrade qualifiers follows that of the RSGs in the WRB Key (for exceptions, see WRB framework). Nearest to the RSG name first (1), then next (2)...
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_wrb_prefixqualifierid	Public	WRB_PrefixQualifierId		
fk_wrb_prefixqualifierid_wrb	Public	WRBId		

Relationships

Columns	Association	Notes
(WRBId=WRBId)	**0..*** **WRB_PrefixQualifier**.fk_wrb_prefixqualifierid_wrb **1** **WRB**.pk_wrb	

WRB_SuffixQualifier

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Classification
Detail: *Created on* 24-11-2010. *Last modified on* 25-3-2011.
Notes: Qualifiers are meant to complete the soil description through Reference soil groups (RSG). These qualifiers are either prefixes or suffixes. For each RSG, there are several possible prefixes and suffixes. When using qualifiers, suffixes are placed between brackets following the RSG name.
A WRB suffix qualifier is part of a WRB soil classification in table *WRB* (referred to by *WRBId*). There may be more than one (order-relevant) suffix qualifier per WRB soil classification.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	WRB_SuffixQualifierId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	WRBId	bigint	True	False			Reference to an entry in the WRB table that represents a WRB soil classification.
False	SuffixQualifier	varchar	True	False	50		Suffix qualifiers are all other qualifiers. That is, not associated or intergrade.
False	Specifier	varchar	False	False	50		Specifiers such as Epi-, Endo-, Hyper-, Hypo-, Thapto-, Bathy-, Para-, Proto-, Cumuli- and Ortho- are used to indicate a certain expression of the qualifier.
False	Order	integer	True	False			The order of suffix qualifiers is as follows: (1) qualifiers related to diagnostic horizons, properties or materials; (2) qualifiers related to chemical characteristics; (3) qualifiers related to physical characteristics; (4) qualifiers related to mineralogical characteristics; (5) qualifiers related to surface characteristics; (6) qualifiers related to textural characteristics, including coarse fragments; (7) qualifiers related to colour; and (8) remaining qualifiers. Nearest to the RSG name first (1), then next (2)...
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_wrb_suffixqualifier	Public	WRB_SuffixQualifierId		
fk_wrb_suffixqualifier_wrb	Public	WRBId		

Relationships

Columns	Association	Notes
(WRBId= WRBId)	**0..*** **WRB_SuffixQualifier**.fk_wrb_suffixqualifier_wrb **1** **WRB**.pk_wrb	

Contact

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 14-12-2010. Last modified on 28-3-2011.*
Notes: The Contact schema contains relations that deal with data describing organizations and / or persons that have been instrumental in obtaining an observation result. It is the single entry point to authoritative names and contact information in the overall database system. This is to prevent the use of different names or spellings for the same organization or individual in various parts of the database where this information is required (e.g. KIT, Tropen-Instituut, Royal Tropical Institute, Koninklijk Instituut voor de Tropen).

In an ideal situation this schema would not be implemented. The core database would link up with an actively maintained and up-to-date external Customer Relationship Management (CRM) system that would be used for other purposes as well.

The current Contact schema contains two relations. See Figure 15.

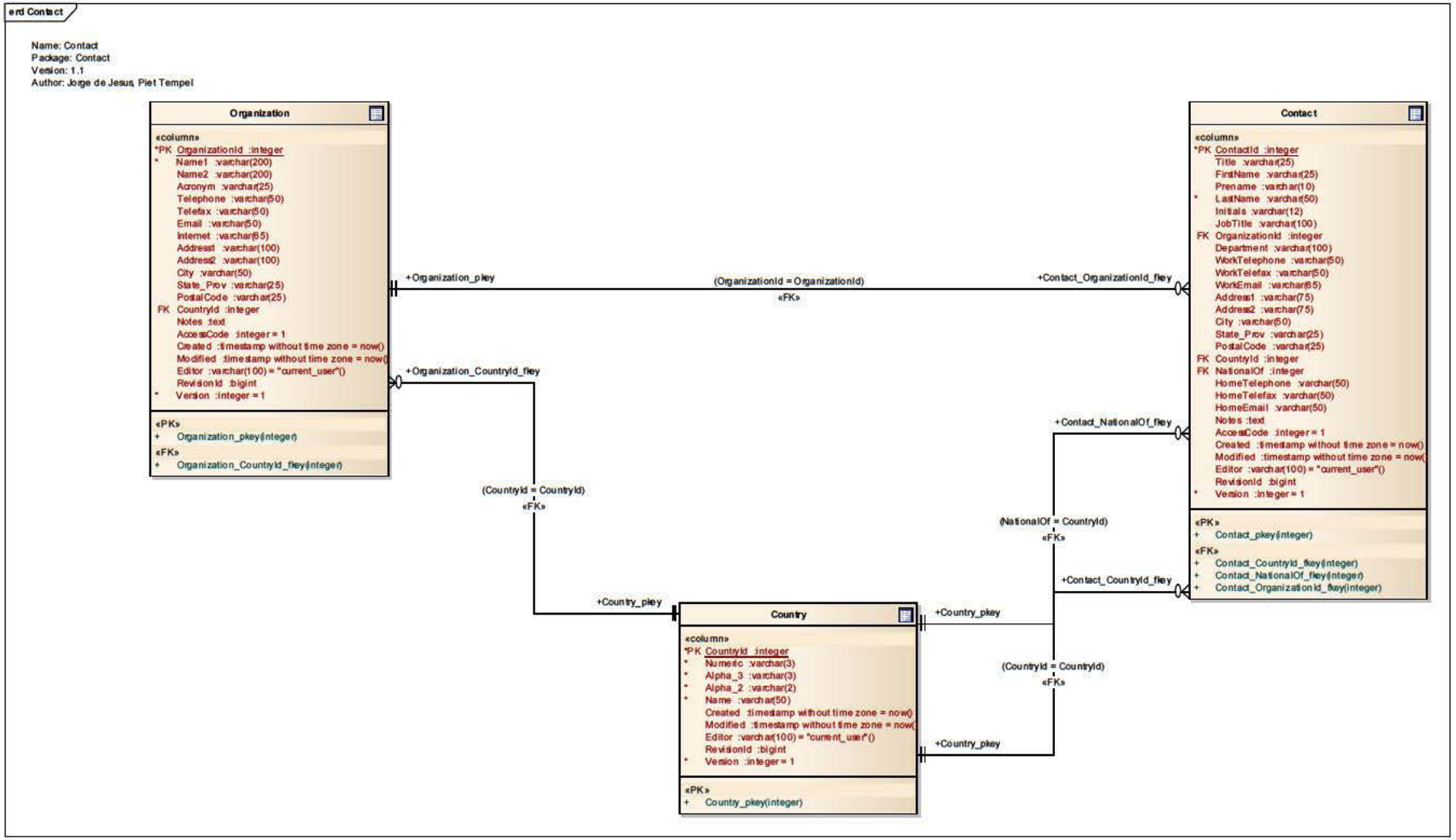

Figure 15
Contact schema.

Contact

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Contact
Detail: *Created on* 14-12-2010. *Last modified on* 24-3-2011.
Notes: Table *Contact* captures contact information about real-world people ('contacts') that in some capacity have played a role in the creation, gathering, management, and dissemination of the data in the database. The *OrganizationId* field links a person to an organization.

Columns

PK	Name	Type	Not Null	Unique	Scale	Init	Notes
True	ContactId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Title	varchar	False	False	25		
False	FirstName	varchar	False	False	25		
False	Prename	varchar	False	False	10		
False	LastName	varchar	True	False	50		
False	Initials	varchar	False	False	12		
False	JobTitle	varchar	False	False	100		
False	OrganizationId	integer	False	False			Reference to an entry in the Organization table that represents an organization.
False	Department	varchar	False	False	100		
False	WorkTelephone	varchar	False	False	50		
False	WorkTelefax	varchar	False	False	50		
False	WorkEmail	varchar	False	False	65		
False	Address1	varchar	False	False	75		
False	Address2	varchar	False	False	75		
False	City	varchar	False	False	50		
False	State_Prov	varchar	False	False	25		
False	PostalCode	varchar	False	False	25		
False	CountryId	integer	False	False			Reference to an entry in the Country table that represents a country (according to ISO 3166-1:2004).
False	NationalOf	integer	False	False			Reference to an entry in the Country table that represents a national of that country (according to ISO 3166-1:2004).
False	HomeTelephone	varchar	False	False	50		
False	HomeTelefax	varchar	False	False	50		

PK	Name	Type	Not Null	Unique	Scale	Init	Notes
False	HomeEmail	varchar	False	False	50		
False	Notes	text	False	False			
False	AccessCode	integer	False	False		1	Code 1 implies full access. Codes may be added as needed.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Contact_CountryId_fkey	Public	CountryId		
Contact_NationalOf_fkey	Public	NationalOf		

Relationships

Columns	Association		Notes
(CountryId = CountryId)	**0..***	**Contact**.Contact_CountryId_fkey	
	1	**Country**.Country_pkey	
(NationalOf = CountryId)	**0..***	**Contact**.Contact_NationalOf_fkey	
	1	**Country**.Country_pkey	
(OrganizationId = OrganizationId)	**0..***	**Contact**.**Contact_OrganizationId_fkey**	
	1	**Organization**.Organization_pkey	

Organization

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Contact
Detail: *Created on* 14-12-2010. *Last modified on* 24-3-2011.
Notes: Table *Organization* captures contact information about organizations that in some capacity have played a role in the creation, gathering, management, and dissemination of the data in the database. An organization is a social arrangement which pursues collective goals, controls its own performance, and has a boundary separating it from its environment. There are a variety of legal types of organizations, including: corporations, governments, non-governmental organizations, international organizations, charities, not-for-profit corporations, partnerships, cooperatives, and universities.
Note that a soil analytical laboratory is considered (part of) an organization.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	OrganizationId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Name1	varchar	True	False	200		
False	Name2	varchar	False	False	200		
False	Acronym	varchar	False	False	25		
False	Telephone	varchar	False	False	50		
False	Telefax	varchar	False	False	50		
False	Email	varchar	False	False	50		
False	Internet	varchar	False	False	65		
False	Address1	varchar	False	False	100		
False	Address2	varchar	False	False	100		
False	City	varchar	False	False	50		
False	State_Prov	varchar	False	False	25		
False	PostalCode	varchar	False	False	25		
False	CountryId	integer	False	False			Reference to an entry in the Country table that represents a country (according to ISO 3166-1:2004).
False	Notes	text	False	False			
False	AccessCode	integer	False	False		1	Code 1 implies full access. Codes may be added as needed.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Organization_pkey	Public	OrganizationId		
Organization_CountryId_fkey	Public	CountryId		

Relationships

Columns	Association	Notes
(OrganizationId = OrganizationId)	**0..*** **Contact**.Contact_OrganizationId_fkey **1** **Organization**.Organization_pkey	
	0..* **Organization_x_AnalyticalMethod**.OrganizationAnalyticalMethod_OrganizationId_fkey **1** **Organization**.Organization_pkey	
(Dataset_Publisher = OrganizationId)	**0..*** **DataSetCitation**.DataSetCitation_Dataset_Publisher_fkey **1** **Organization**.Organization_pkey	
	0..* **DataCenter**.DataCenter_OrganizationId_fkey **1** **Organization**.Organization_pkey	

DIF

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 8-2-2011. Last modified on 28-3-2011.*
Notes: ISRIC's Core Database metadata is a database implementation of the Directory Interchange Format (with minor modifications).
The DIF is used to create directory entries which describe a group of data - that is, data sets. A DIF consists of a collection of fields which detail specific information about the data. The DIF allows users of data to understand the contents of a data set and contains those fields which are necessary for users to decide whether a particular data set would be useful for their needs.
The DIF table is a pivotal table in the Core metadata database. This table contains the eight fields that are required in the DIF; the others expand upon and clarify the information but are not repetitive. Repeatable DIF fields are stored in separate, but related tables. That is, nearly all other tables are linked to the DIF table, and represent (groups of) repetitive data elements, e.g. keywords associated with a dataset.
Some minor elements that are repeatable according to the DIF standard are not repeatable in the metadata database (to keep the schema somewhat simpler).

See Figures 16 - 20.

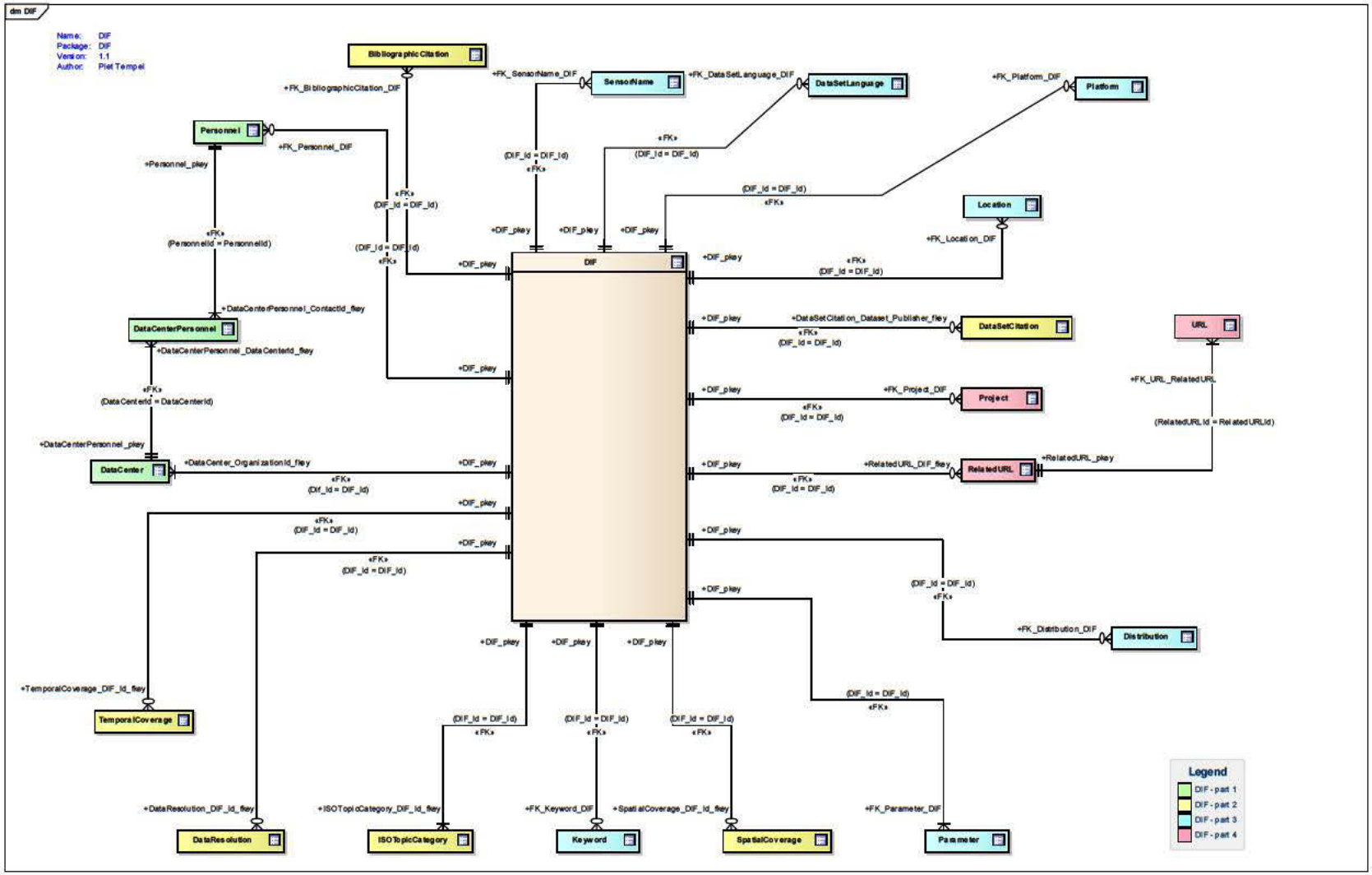

Figure 16
DIF schema.

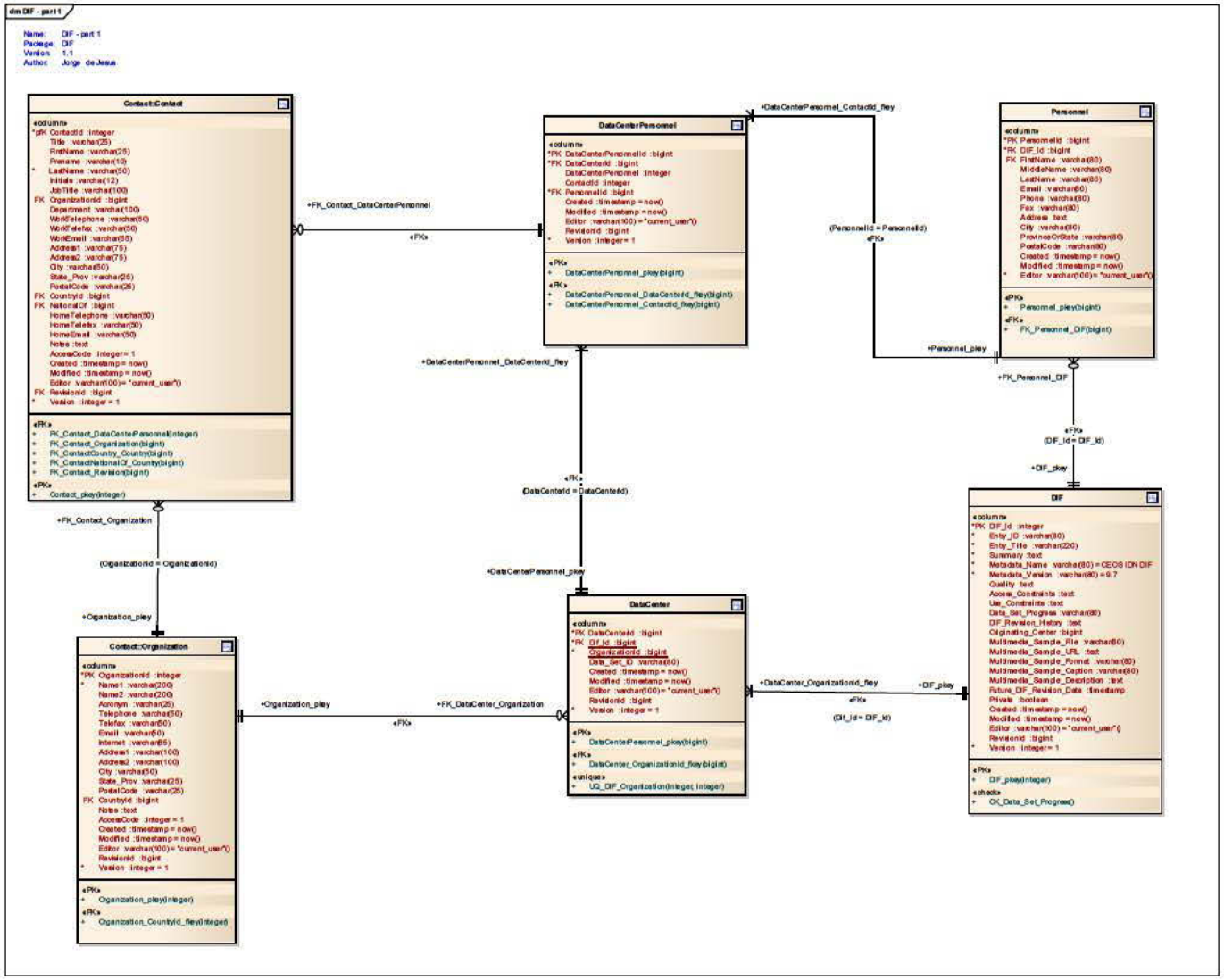

Figure 17

DIF schema - part 1.

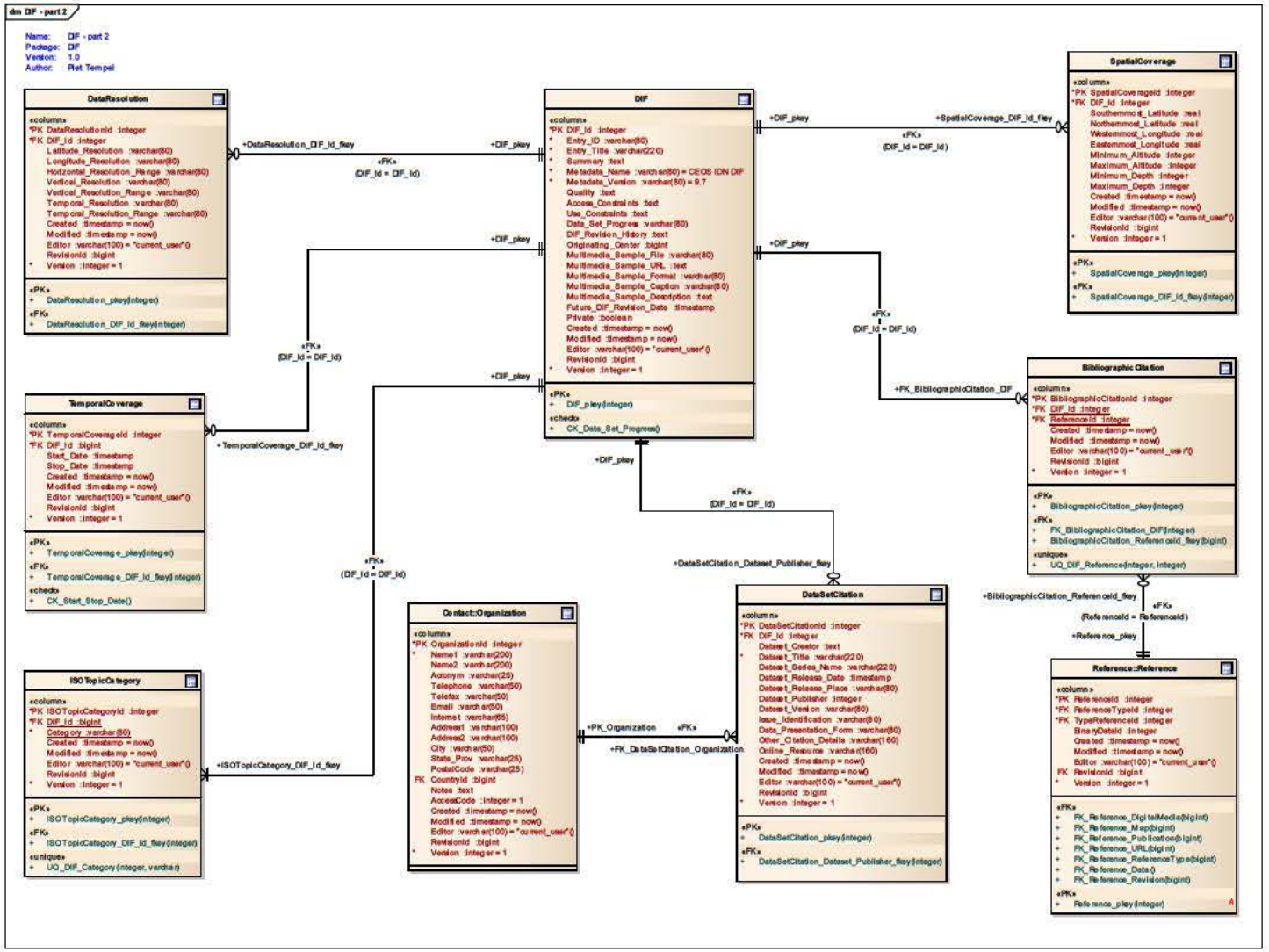

Figure 18

DIF schema - part 2.

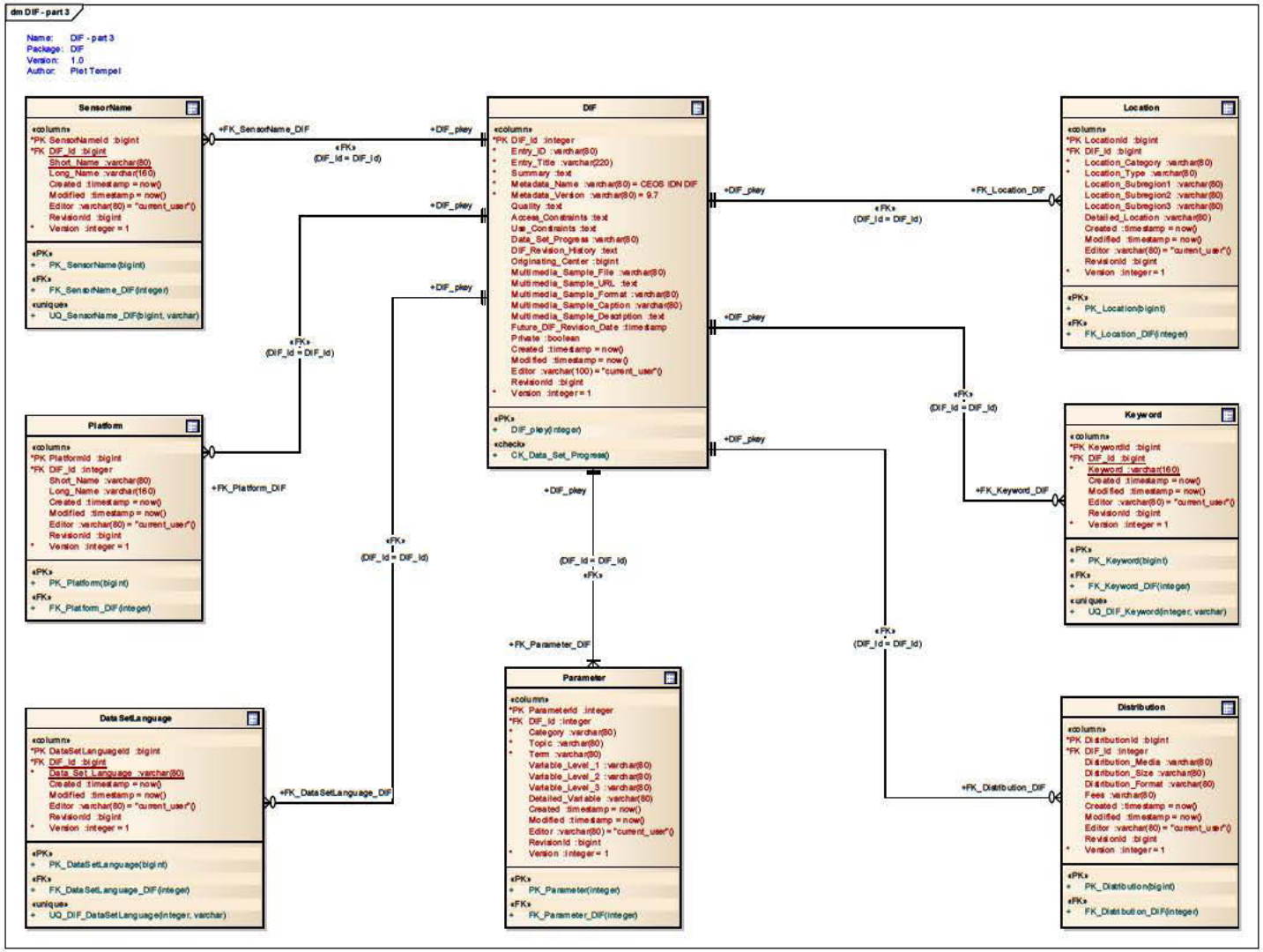

Figure 19
DIF schema - part 3.

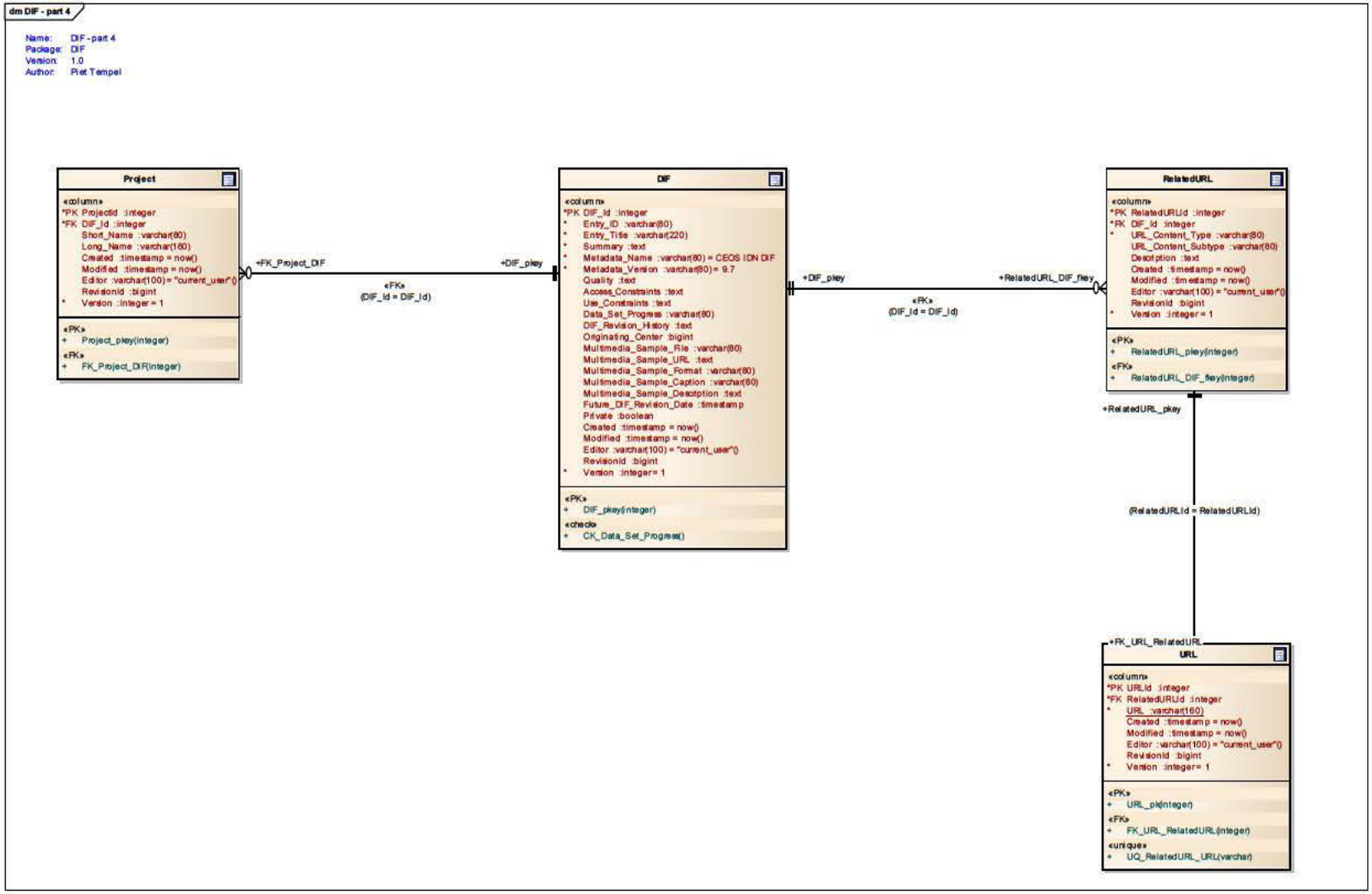

Figure 20
DIF schema - part 4.

BibliographicCitation

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 22-11-2005. *Last modified on* 10-3-2011.
Notes: The BibliographicCitations table describes key bibliographic citations pertaining to a data set. Each data set is identified by its *DIF_Id*, each bibliographic citation by its *ReferenceID*.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	BibliographicCitationId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	True			Reference to an entry in the DIF table that is describing a data set.
False	ReferenceId	integer	True	True			Reference to an entry in the Reference table that is a bibliographic citation.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
BibliographicCitationId	Public	OrganizationId		
BibliographicCitation_ReferenceId_fkey	Public	ReferenceId		

Relationships

Columns	Association	Notes
(ReferenceId=ReferenceId)	**0..*** **BibliographicCitation**.BibliographicCitation_ReferenceId_fkey **1** **Reference**.Reference_pkey	
(DIF_Id = DIF_Id)	**0..*** **BibliographicCitation.BibliographicCitation_DIF_Id_fkey** **1** **DIF**.DIF_pkey	

DIF

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The Directory Interchange Format, the DIF, is the product of an Earth Science and Applications Data Systems Workshop (ESADS) held February 24-26, 1987 on catalog interoperability (CI). It is a descriptive and standardized format for exchanging information about scientific data sets. The DIF is used to create directory entries which describe a group of data. A DIF consists of a collection of fields which detail specific information about the data. Eight fields are required in the DIF; the others expand upon and clarify the information. Some of the fields are text fields, others require the use of controlled keywords (sometimes known as 'valids'). Repeatable DIF fields are stored in separate, but related tables.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DIF_Id	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Entry_ID	varchar	True	False	80		The unique identifier of the metadata record. The Entry_ID consists of 1 to 80 alphanumeric characters of the UTF-8 set, including underbar (_), hyphen (-) and period (.).
False	Entry_Title	varchar	True	False	220		The Entry_Title is the title of the data set described by the metadata. It should be descriptive enough so that when a user is presented with a list of titles the general content of the data set can be determined.
False	Summary	text	True	False			A brief description of the data set that allows potential users to determine if the data set is useful for their needs.
False	Metadata_Name	varchar	True	False	80		CEOS IDN DIF Identifies the current DIF standard name.
False	Metadata_Version	varchar	True	False	80		9.7 Identifies the current DIF metadata standard version.
False	Quality	text	False	False			This field allows the author to provide information about the quality of the data or any quality assurance procedures followed in producing the data described in the metadata.
False	Access_Constraints	text	False	False	0		Allows the author to provide information about any constraints for accessing the data set. This includes any special restrictions, legal prerequisites, limitations and/or warnings on obtaining the data set.
False	Use_Constraints	text	False	False			Allows the author to describe how the data may or may not be used after access is granted to assure the protection of privacy or intellectual property.
False	Data_Set_Progress	varchar	False	False	50		Describes the production status of the data set regarding its completeness.

False	DIF_Revision_History	text	False	False			Allows the author to provide a list of changes made to the DIF over time. This provides a mechanism for tracking revisions to DIF content.
False	Originating_Center	bigint	False	False			Reference to an entry in the Organizations table that represents the data center or data producer who originally generated the dataset.
False	Multimedia_Sample_File	varchar	False	False	80		Describes the filename where the dataset's multimedia sample can be found.
False	Multimedia_Sample_URL	text	False	False			URL of the (multimedia) sample file.
False	Multimedia_Sample_Format	varchar	False	False	80		The format of the multimedia sample file (which may differ from the format of the data files), i.e. GIF, TIFF, JPEG.
False	Multimedia_Sample_Caption	varchar	False	False	80		A one-line description of the multimedia sample is used as a caption when the sample is displayed. The caption is especially useful for images such as graphs and photos.
False	Multimedia_Sample_Description	text	False	False			
False	Future_DIF_Revision_Date	timestamp	False	False			Future date at which the DIF should be reviewed for accuracy of scientific or technical content.
False	Private	boolean	False	False			Allows the author to restrict the data set description from being publicly available.
False	Created	timestamp	False	False	now()		Date of record creation.
False	Modified	timestamp	False	False	now()		Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Data_Set_Progress_check	Public		((("Data_Set_Progress")::text = 'Planned'::text) OR (("Data_Set_Progress")::text = 'In Work'::text)) OR (("Data_Set_Progress")::text = 'Complete'::text))	

Relationships

Columns	Association	Notes
	0..* **SensorName**.SensorName_DIF_Id_fkey **1** **DIF**.DIF_pkey	
	0..* **Profile_x_DIF**.Profile_x_DIF_DIF_Id_fkey **1** **DIF**.DIF_pkey	
	1..* **DataCenter**.DataCenter_Dif_Id_fkey **1** **DIF**.DIF_pkey	

DataCenter

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 22-11-2005. *Last modified on* 28-2-2011.
Notes: The DataCenter table stores information regarding the data center, organization, or institution responsible for distributing the data - indirectly, through a reference to an organization in the Organization table.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DataCenterId	integer	True	False			Unique identifier for each possible record in the table.
False	Dif_Id	integer	True	True	0		Reference to an entry in the DIF table that is describing a data set.
False	OrganizationId	integer	True	True			Reference to an entry in the Organization table that is describing an organization.
False	Data_Set_ID	varchar	False	False	80		Data set identifier assigned by the data center (may or may not be the same as the Entry_ID).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
DataCenter_Dif_Id_fkey	Public	Dif_Id		
DataCenter_OrganizationId_fkey	Public	PublicOrganizationId		

Relationships

Columns	Association		Notes
(DataCenterId = DataCenterId)	**1..***	**DataCenterPersonnel**.DataCenterPersonnel_DataCenterId_fkey	
	1	**DataCenter**.DataCenter_pkey	
(OrganizationId=OrganizationId)	**0..***	**DataCenter**.DataCenter_OrganizationId_fkey	
	1	**Organization**.Organization_pkey	
(Dif_Id = DIF_Id)	**1..***	**DataCenter**.DataCenter_Dif_Id_fkey	
	1	**DIF**.DIF_pkey	

DataCenterPersonnel

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The DataCenterPersonnel table refers to one or more contact persons in the Personnel table for the data set. The only admissible role for a contact person in the Personnel table is 'Data Center Contact'. This role is assumed implicitly.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DataCenterPersonnelId	integer	True	False			Unique identifier for each possible record in the table.
False	DataCenterId	integer	True	False			Reference to an entry in the DataCenter table that represents a data center.
False	PersonnelId	integer	True	False			Reference to an entry in the Personnel table that represents a contact person for the data set.
False	ContactId	integer	True	False			Foreign key to ContactId
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
DataCenterPersonnel_pkey	Public	DataCenterPersonnelId		
DataCenterPersonnel_DataCenterId_fkey	Public	DataCenterId		
DataCenterPersonnel_ContactId_fkey	Public	ContactId		

Relationships

Columns	Association	Notes
(ContactId = ContactId)	**1** **DataCenterPersonnel**.DataCenterPersonnel_ContactId_fkey **1** **Contact.Contact_pkey**	
(DataCenterId = DataCenterId)	**1..*** **DataCenterPersonnel**.DataCenterPersonnel_DataCenterId_fkey **1** **DataCenter**.DataCenter_pkey	

DataResolution

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The DataResolution table specifies the resolution of the data, which is the difference between two adjacent geographic, vertical, or temporal values. Selection of data resolution ranges will assist users in refining their search for data within specific resolution ranges.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DataResolutionId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	Latitude_Resolution	varchar	False	False	50		The minimum difference between two adjacent latitude values (for example "2.5 degrees").
False	Longitude_Resolution	varchar	False	False	50		The minimum difference between two adjacent longitude values (for example "2.5 degrees").
False	Horizontal_Resolution_Range	varchar	False	False	80		The latitude/longitude resolution range. The range should be selected based on the Latitude_Resolution and Longitude_Resolution.
False	Vertical_Resolution	varchar	False	False	50		The minimum difference possible between two adjacent vertical values.
False	Vertical_Resolution_Range	varchar	False	False	50		The vertical resolution range. The range should be selected based on the specified Vertical_Resolution.
False	Temporal_Resolution	varchar	False	False	50		The frequency of data sampled.
False	Temporal_Resolution_Range	varchar	False	False	50		The temporal resolution range. The range should be selected based on the specified Temporal_Resolution.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
DataResolution_pkey	Public	DataResolutionId		
DataResolution_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **DataResolution**.DataResolution_DIF_Id_fkey **1** **DIF**.DIF_pkey	

DataSetCitation

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The DataSetCitation table allows for the proper citation of the data set producer. The table has two functions:

- To indicate how this data set should be cited in the professional scientific literature, and
- If this data set is a compilation of other data sets, to document and credit the data sets that were used in producing this compilation.

This table is not to be used to list bibliographic references of scientific research articles arising from the data set. This table provides a citation for the data set itself, not articles related to the research results. To list references related to the research results, use the BibliographicCitation table.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DataSetCitationId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	Dataset_Creator	text	False	False			The name of the organization(s) or individual(s) with primary intellectual responsibility for the data set's development.
False	Dataset_Title	varchar	True	False	220		The Title of the data set; this may be the same as Entry_Title in the DIF table.
False	Dataset_Series_Name	varchar	False	False	220		The name of the dataset series, or aggregate dataset of which the dataset is a part.
False	Dataset_Release_Date	timestamp	False	False			The date when the data set was made available for release.
False	Dataset_Release_Place	varchar	False	False	80		The name of the city (and state or province, and country if needed) where the data set was made available for release.
False	Dataset_Publisher	bigint	False	False			
False	Dataset_Version	varchar	False	False	80		The Version of the data set.
False	Issue_Identification	varchar	False	False	80		The volume or issue number of the publication (if applicable).
False	Data_Presentation_Form	varchar	False	False	80		The mode in which the data are represented, e.g. atlas, image, profile, text, etc.
False	Other_Citation_Details	varchar	False	False	160		Additional free-text citation information.
False	Online_Resource	varchar	False	False	160		The URL of the online resource containing the data set.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
DataSetCitation_pkey	Public	DataSetCitationId		
DataSetCitation_Dataset_Publisher_fkey	Public	Dataset_Publisher		

Relationships

Columns	Association	Notes
(Dataset_Publisher = OrganizationId)	**0..*** **DataSetCitation**.DataSetCitation_Dataset_Publisher_fkey **1** **Organization**.Organization_pkey	

DataSetLanguage

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The DataSetLanguage table describes the language used in the preparation, storage, and description of the data. It is the language of the information object, not the language used to describe or interact with the metadata record. DataSetLanguage does not refer to the language of the metadata.
A DataSetLanguage entry can be repeated if data is available in more than one language.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DataSetLanguageId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	True			Reference to an entry in the DIF table that is describing a data set.
False	Data_Set_Language	varchar	True	True	50		The language used in the preparation, storage, and description of the data.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
DataSetLanguage_pkey	Public	DataSetLanguageId		
DataSetLanguage_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **DataSetLanguage**.DataSetLanguage_DIF_Id_fkey **1** **DIF**.DIF_pkey	

Distribution

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The Distribution table describes media options, size, data format, and fees involved in distributing the data set.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DistributionId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False	0		Reference to an entry in the DIF table that is describing a data set.
False	Distribution_Media	varchar	False	False	80		The media options for the user receiving the data.
False	Distribution_Size	varchar	False	False	80		An approximate size (in KB, MB or GB) for the entire data set. Specify if data are compressed and the method of compression.
False	Distribution_Format	varchar	False	False	80		The data format used to distribute the data.
False	Fees	varchar	False	False	80		Cost of distribution media, or distribution costs, if any.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Distribution_pkey	Public	DistributionId		
Distribution_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **Distribution**.Distribution_DIF_Id_fkey **1** **DIF**.DIF_pkey	

ISOTopicCategory

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The ISOTopicCategory table is used to identify the keywords in the ISO 19115 - Geographic Information Metadata (http://www.isotc211.org/) Topic Category Code List. It is a high-level geographic data thematic classification to assist in the grouping and search of available geographic data sets.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ISOTopicCategoryId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	True			Reference to an entry in the DIF table that is describing a data set.
False	Category	varchar	True	True	80		Single keyword from the ISO 19115 - Geographic Information Metadata Topic Category Code List.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

ISOTopicCategory_pkey	Public	ISOTopicCategoryId		
ISOTopicCategory_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**1..*** **ISOTopicCategory**.ISOTopicCategory_DIF_Id_fkey **1** **DIF**.DIF_pkey	

Keyword

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 10-3-2011.
Notes: The Keyword table allows authors to provide any words or phrases needed to further describe the data set.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	KeywordId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	True			
False	Keyword	varchar	True	True	160		
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Keyword_pkey	Public	KeywordId		
Keyword_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **Keyword**.Keyword_DIF_Id_fkey **1** **DIF**.DIF_pkey	

Location

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 10-3-2011.
Notes: The Location table specifies the name of a place on Earth, a location within the Earth, a vertical location, or a location outside of Earth. The Location keywords are a 5-level hierarchy of controlled keywords.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	LocationId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	Location_Category	varchar	True	False	50		Location Category keyword.
False	Location_Type	varchar	True	False	50		Location type keyword.
False	Location_Subregion1	varchar	False	False	50		First location subregion keyword.
False	Location_Subregion2	varchar	False	False	50		Second location subregion keyword.
False	Location_Subregion3	varchar	False	False	50		Third location subregion keyword.
False	Detailed_Location	varchar	False	False	50		Allows authors to include any location keywords that are not available in the controlled keyword list.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Location_pkey	Public	LocationId		
Location_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **Location**.Location_DIF_Id_fkey **1** **DIF**.DIF_pkey	

Parameter

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The Parameter table allows for the specification of Earth science keywords that are representative of the data set being described. These keywords are important for the precise search and retrieval of information. The author must select these keywords from a controlled set of science keywords. The Parameters table consists of a 7-level hierarchical classification of science keywords.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ParameterId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	Category	varchar	True	False	80		Highest keyword category.
False	Topic	varchar	True	False	80		Topic keyword.
False	Term	varchar	True	False	80		Term keyword.
False	Variable_Level_1	varchar	False	False	80		Variable Level 1 keyword.
False	Variable_Level_2	varchar	False	False	80		Variable Level 2 keyword.
False	Variable_Level_3	varchar	False	False	80		Variable Level 3 keyword.
False	Detailed_Variable	varchar	False	False	80		Uncontrolled free text field that allows a metadata author to specify any keywords to more exactly describe the measurement represented by the data.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Parameter_pkey	Public	ParameterId		
Parameter_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**1..*** **Parameter**.Parameter_DIF_Id_fkey **1** **DIF**.DIF_pkey	

Personnel

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 23-12-2011.
Notes: The Personnel table defines the point of contact for more information about the data set or the metadata.
The table relates to Contact.Contact where personal data is stored

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	PersonnelId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	ContactId	integer	True	False			Relates to Contact.Contact table
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	True	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Personnel_pkey	Public	PersonnelId		
Personnel_ContactId_fkey	Public	ContactId		
Personnel_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **Personnel**.Personnel_DIF_Id_fkey **1** **DIF**.DIF_pkey	
(ContactId=ContactId)	1 **Personnel**.Personnel_ContactId_fkey 1 Contact.Contact_pkey	

Platform

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The Platform table holds information about the platform used to acquire the data.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	PlatformId	bigint	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False	0		Reference to an entry in the DIF table that is describing a data set.
False	Short_Name	varchar	False	False	80		The abbreviated name of the platform (source) used to acquire the data.
False	Long_Name	varchar	False	False	160		The full name of the platform (source) used to acquire the data.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Platform_pkey	Public	PlatformId		
Platform_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **Platform**.Platform_DIF_Id_fkey **1** **DIF**.DIF_pkey	

Project

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 10-3-2011.
Notes: The Project table holds the name(s) of the scientific program, field campaign, or project from which the data were collected.
The table allows for the specification of keywords that are the names of the projects. These keywords are important for the search and retrieval of information. The table may store the short name and the long name of the project. There is a 1:1 correspondence between the short name and the long name.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ProjectId	bigint	True	False			The Project table holds information about the scientific program, field campaign, or project from which the data were collected. This table allows for the specification of keywords that are the names of the projects
False	DIF_Id	bigint	True	False			Reference to an entry in the DIF table that is describing a data set.
False	Short_Name	varchar	False	False	80		The abbreviated name of the project from which the data were collected.
False	Long_Name	varchar	False	False	160		The full name of the project from which the data were collected.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Project_pkey	Public	ProjectId		

Relationships

Columns	Association	Notes

RelatedURL

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The RelatedURL table specifies links to Internet sites that contain information related to the data, as well as related Internet sites such as project home pages, related data archives/servers, metadata extensions, online software packages, web mapping services, and calibration/validation data.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	RelatedURLId	bigint	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False	0		Reference to an entry in the DIF table that is describing a data set.
False	URL_Content_Type	varchar	True	False	80		The type of resources being referenced by the URL.
False	URL_Content_Subtype	varchar	False	False	80		Describes the subtype of the resource being referenced by the URL.
False	Description	text	False	False	0		Provides information about the resource(s).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
RelatedURL_pkey	Public	RelatedURLId		
RelatedURL_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **RelatedURL**.RelatedURL_DIF_Id_fkey **1** **DIF**.DIF_pkey	
(RelatedURLID = RelatedURLID)	**1..*** **URL**.URL_RelatedURLId_fkey **1** **RelatedURL**.RelatedURL_pkey	

SensorName

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The SensorName table stores the names of the instruments used to acquire the data.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	SensorNameId	bigint	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	bigint	True	True			Reference to an entry in the DIF table that is describing a data set.
False	Short_Name	varchar	False	True	80		The abbreviated name of the instrument used to acquire the data.
False	Long_Name	varchar	False	False	160		The full name of the instrument used to acquire the data.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
SensorName_pkey	Public	SensorNameId		
SensorName_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **SensorName**.SensorName_DIF_Id_fkey **1** **DIF**.DIF_pkey	

SpatialCoverage

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The SpatialCoverage table specifies the geographic and vertical (altitude, depth) coverage of the data.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	SpatialCoverageId	bigint	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	Southernmost_Latitude	real	False	False			The southernmost geographic latitude covered by the data. From: 0 - 90° North or 0 - 90° South.
False	Northernmost_Latitude	real	False	False			The northernmost geographic latitude covered by the data. From: 0 - 90° North or 0 - 90° South.
False	Westernmost_Longitude	real	False	False			The westernmost geographic longitude covered by the data. From: 0 - 180° east or 0 - 180° west. The Prime Meridian is 0 degrees, measured positive (+) eastwards of the PM.
False	Easternmost_Longitude	real	False	False			The easternmost geographic longitude covered by the data. From: 0 - 180° east or 0 - 180° west. The Prime Meridian is 0 degrees, measured positive (+) eastwards of the PM.
False	Minimum_Altitude	integer	False	False			The altitude level, which represents the lower limit of data coverage (meters).
False	Maximum_Altitude	integer	False	False			The altitude level, which represents the higher limit of data coverage (meters).
False	Minimum_Depth	integer	False	False			The depth level, which represents the upper-most depth of data coverage (meters).
False	Maximum_Depth	integer	False	False			The depth level, which represents the lowest depth of data coverage (meters).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
SpatialCoverage_pkey	Public	SpatialCoverageId		
SpatialCoverage_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*** **SpatialCoverage**.SpatialCoverage_DIF_Id_fkey **1** **DIF**.DIF_pkey	

TemporalCoverage

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The TemporalCoverage table specifies the start and stop dates during which the data was collected.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	TemporalCoverageId	integer	True	False			Unique identifier for each possible record in the table.
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	Start_Date	timestamp	False	False			The starting date of the data collection.
False	Stop_Date	timestamp	False	False			The ending date of the data collection.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
TemporalCoverage_pkey	Public	TemporalCoverageId		
TemporalCoverage_DIF_Id_fkey	Public	DIF_Id		

Relationships

Columns	Association	Notes
(DIF_Id = DIF_Id)	**0..*TemporalCoverage**.TemporalCoverage_DIF_Id_fkey **1** **DIF**.DIF_pkey	

URL

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* DIF
Detail: *Created on* 27-10-2010. *Last modified on* 28-2-2011.
Notes: The RelatedURL table specifies links to Internet sites that contain information related to the data, as well as related Internet sites such as project home pages, related data archives/servers, metadata extensions, online software packages, web mapping services, and calibration/validation data.
The URL table lists the URL's associated with a particular URL content (sub) type entry in the RelatedURL table.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	URLId	integer	True	False			Unique identifier for each possible record in the table.
False	RelatedURLId	integer	False	False			Reference to an entry in the RelatedURL table that is describing a URL content type (and URL content subtype).
False	URL	text	False	True			The URL to the resource associated with the data set.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
URL_pkey	Public	URLId		
URL_RelatedURLId_fkey	Public	RelatedURLId		

Relationships

Columns	Association	Notes
(RelatedURLID = RelatedURLID)	**1..*** **URL**.URL_RelatedURLId_fkey **1** **RelatedURL**.RelatedURL_pkey	

Location

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 13-12-2010. Last modified on 28-3-2011.*
Notes: The relations in the *Location* schema essentially contain authoritative names for country groups, countries, and 1st level administrative units within countries. As with the Contact schema it is the only entry point to authoritative geographic names (notably of countries) in the core database system. Again, this is to prevent the use of different names or spellings for the same geographic location in various parts of the database where this information is required (e.g. Great Brittain, United Kingdom, England, Royaume Unie). Names comply with ISO-3166, a standard published by the International Organization for Standardization (ISO), which defines codes for the names of countries, dependent territories, and special areas of geographical interest, and their principal subdivisions (e.g., provinces or states). The official name of the standard is *Codes for the representation of names of countries and their subdivisions*. Data have been provided by RIVM, the Dutch National Institute for Public Health and the Environment.
Future access methods to the data in the core database that are location-based may use web services like GeoNames to enable the use of aliases, synonyms, etc.
Relation Region enables ad-hoc grouping of countries, e.g. 'Former Yugoslavia' and 'ASEAN'. Country groups may be used to query the database for these groupings.
Future releases will contain additions spatial information like country polygons.

The current *Location* schema contains 4 relations. See below.

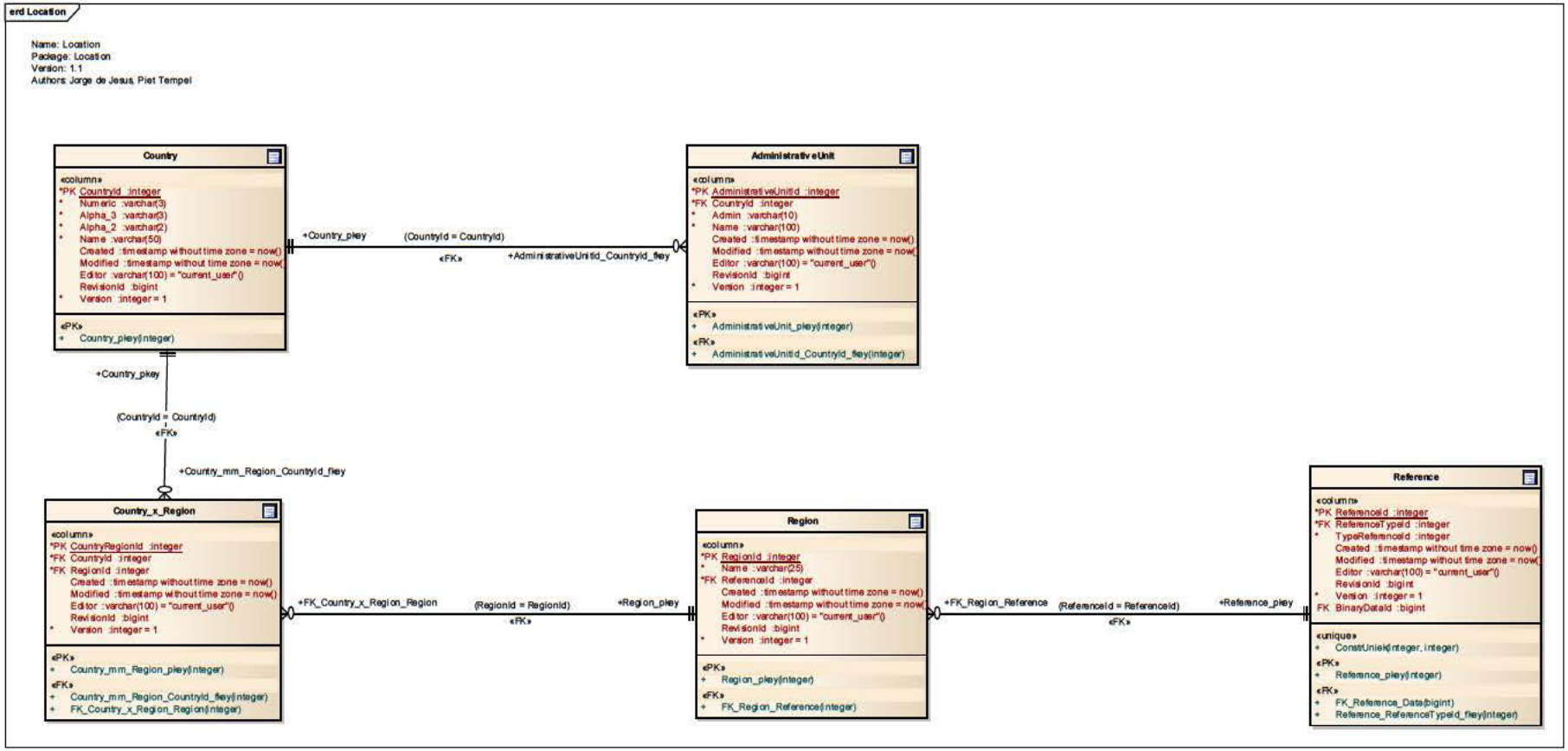

Figure 21
Location schema.

AdministrativeUnit

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Location
Detail: *Created on* 13-12-2010. *Last modified on* 25-3-2011.
Notes: ISO 3166 is the International Standard for country codes. The purpose of ISO 3166 is to establish codes for the representation of names of countries, territories or areas of geographical interest, and their subdivisions.
Table *AdministrativeDivision* is based on ISO 3166-2:2007; Codes for the representation of names of countries and their subdivisions – Part 2: Country subdivision codes, which gives codes for the names of the principal subdivisions (e.g. provinces or states) of all countries coded in ISO 3166-1. This code is based on the two-letter code element from ISO 3166-1 followed by a separator and a further string of up to three alphanumeric characters, which is usually obtained from national sources and stems from coding systems already in use in the country concerned.
A *CountryId* links the administrative divisions to the appropriate countries in table *Country*.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	AdministrativeUnitId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	CountryId	integer	True	True			Reference to a country in the *Location.Country* table.
False	Admin	varchar	True	True	10		A complete ISO 3166-2 subdivision code uniquely identifying a country subdivision in a global context.
False	Name	varchar	True	False	100		ISO 3166-2 name of the administrative subdivision.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
AdministrativeUnit_pkey	Public	AdministrativeDivisionId		
AdministrativeUnitId_CountryId_fkey	Public	CountryId		

Relationships

Columns	Association	Notes
(CountryId = CountryId)	**0..*** **AdministrativeUnit**.AdministrativeUnitId_CountryId_fkey **1** **Country**.Country_pkey	

Country

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Location
Detail: *Created on* 13-12-2010. *Last modified on* 24-3-2011.
Notes: ISO 3166 is the International Standard for country codes. The purpose of ISO 3166 is to establish codes for the representation of names of countries, territories or areas of geographical interest, and their subdivisions.

Table *Country* is based on ISO 3166-1:2006; Codes for the representation of names of countries and their subdivisions – Part 1: Country codes. This is what most users know as ISO's country codes. First published in 1974, it has since then become one of the worlds most popular and most widely used standard solution for coding country names. It contains a two-letter code which is recommended as the general purpose code, a three-letter code which has better mnemonic properties and a numeric-3 code which can be useful if script independence of the codes is important.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	CountryId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Numeric	char	False	False	3		ISO 3166-1 three-digit code; identifies a physical territory. Is identical to that defined by the United Nations Statistical Division. The field is char and not numerical since we have to include zeros to the left of the significant digit so that code is always with 3 numbers
False	Alpha_3	char	False	False	3		ISO 3166-3 three-letter code; identifies a country / region name.
False	Alpha_2	char	True	False	2		ISO 3166-2 three-letter code; identifies a country / region name.
False	Name	varchar	True	False			SO 3166-2 country name.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator, or last modifier, of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Country_pkey	Public	CountryId		

Relationships

Columns	Association	Notes
(CountryId=CountryId)	**0..*** **MapUnit**.Terrain_CountryId_fkey **1** **Country**.Country_pkey	
	0..* **Organization**.Organization_CountryId_fkey 1 **Country**. Country_pkey	
(NationalOf = CountryId)	**0..*** **Contact**.FK_ContactNationalOf_Country **1** **Country**.PK_Country	

Country_x_Region

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Location
Detail: *Created on* 13-12-2010. *Last modified on* 11-4-2011.
Notes: The *Country_x_region* table links countries, listed in the *Country* table, to regions in country groups in the Region table. This link table thus lists all regions that are part of a particular country.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	CountryRegionId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	CountryId	integer	True	True			Foreign key to a country in the Location.Country table.
False	RegionId	integer	True	True			Foreign key to a country group in the Location.Regiontable.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Country_x_Region_pkey	Public	CountryRegionId		
Country_x_Region_CountryId_fkey	Public	CountryId		
FK_Country_x_Region_Region	Public	RegionId		

Relationships

Columns	Association	Notes
(CountryId = CountryId)	**0..*** **Country_x_Region**.Country_x_Region_CountryId_fkey **1** **Country**.Country_pkey	
(RegionId = RegionId)	**1..*** **Country_x_Region**.FK_Country_x_Region_Region **1..*** Location.Region_pkey	

Region

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Location
Detail: *Created on* 13-12-2010. *Last modified on* 24-3-2011.
Notes: Table enables ad-hoc grouping of countries, e.g. 'Former Yugoslavia', 'ASEAN', etc. Country groups may be used to query the database for these groupings.
A region may refer to a map, publication, URL, etc. (in the *Reference.Reference* table) that defines itself.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	"RegionId"	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Name	text	True	False	100		Country group's name.
False	ReferenceId	integer	True	False			Reference to a map, publication, URL, etc. (in the *Reference.Reference* table) that defines the country group.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Region_pkey	Public	RegionId		
FK_Region_Reference	Public	ReferenceId		

Relationships

Columns	Association	Notes
(RegionId = RegionId)	**1..*** **Region**.FK_Region_Reference **1** Country_x_Region.FK_Country_x_Region_Region	
	0..* **Region**.FK_Region_Reference **1** **Reference**.Reference_pkey	

MapUnit

Type: **Package**

Package: ISRIC Soil Data Repository

Notes: A map unit is a collection of areas defined and named the same in terms of their differentiating criteria, or miscellaneous areas[16], or both. Each map unit differs in some respect from all others in the area of interest and is uniquely identified on a map. Each individual area, or polygon, on the map is a delineation of a particular map unit. Map units are represented by cartographic units with a single colour or pattern.
The database will store each map unit as a single point set union of their polygon geometries (i.e. borders). All polygon maps, and therefore their mapping units, are stored in a single table. Reference datum for any point on the earth's surface that is referenced in the database is WGS84.
Applying differentiating criteria in a step-by-step manner may eventually lead to overly complex and /or fragmented map units that cannot be mapped at the scale under consideration. The information is then stored as a non-mappable map unit component without a geometry. Thus, a map unit may consist of one or more map unit components, each covering a certain percentage of the map unit.
The sub model represented by this schema has to some extent been based by the SOTER database schema.

[16] Miscellaneous areas are areas that fall outside the differentiating criteria, e.g. water bodies under terrestrial criteria.

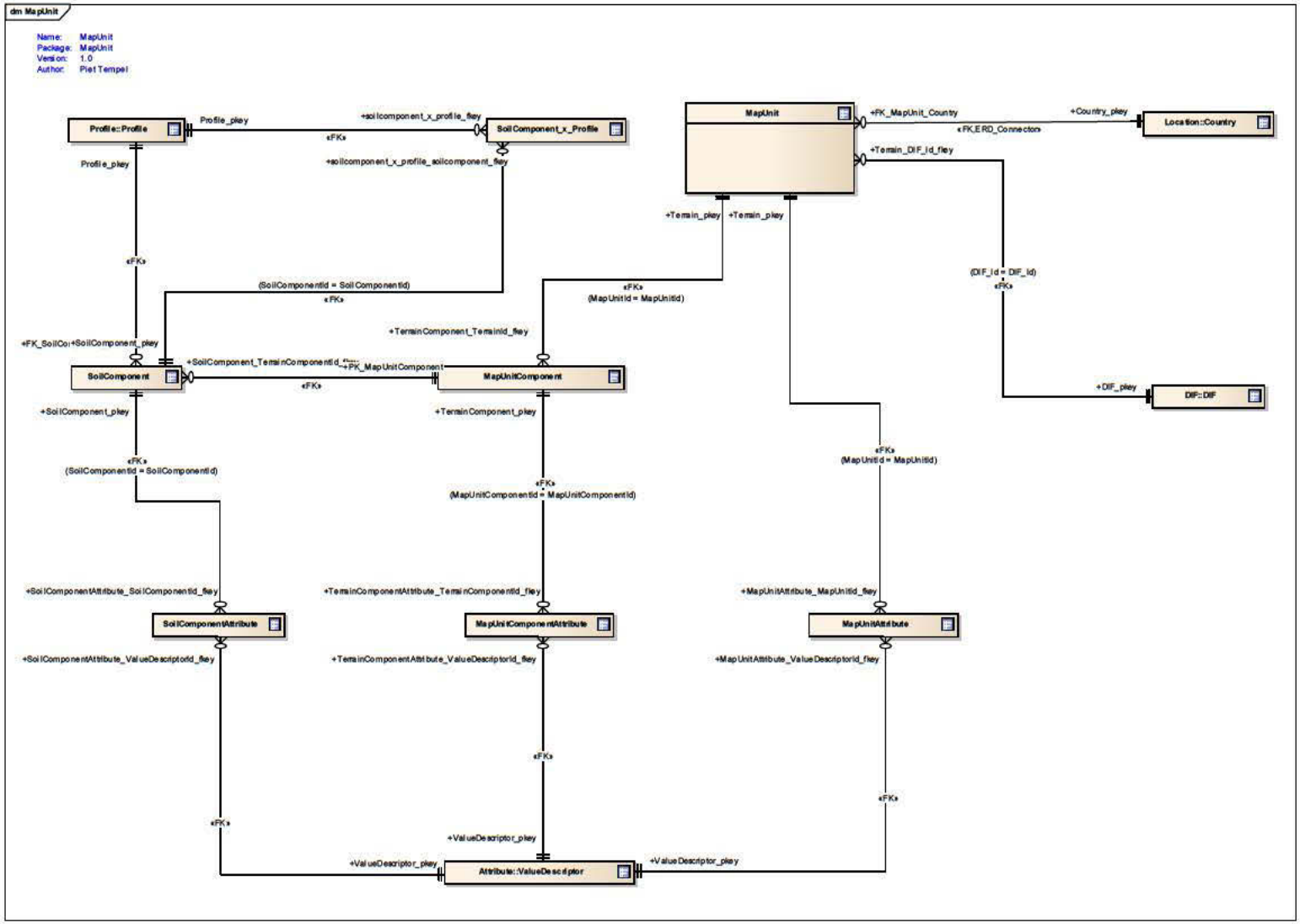

Figure 22
MapUnit schema.

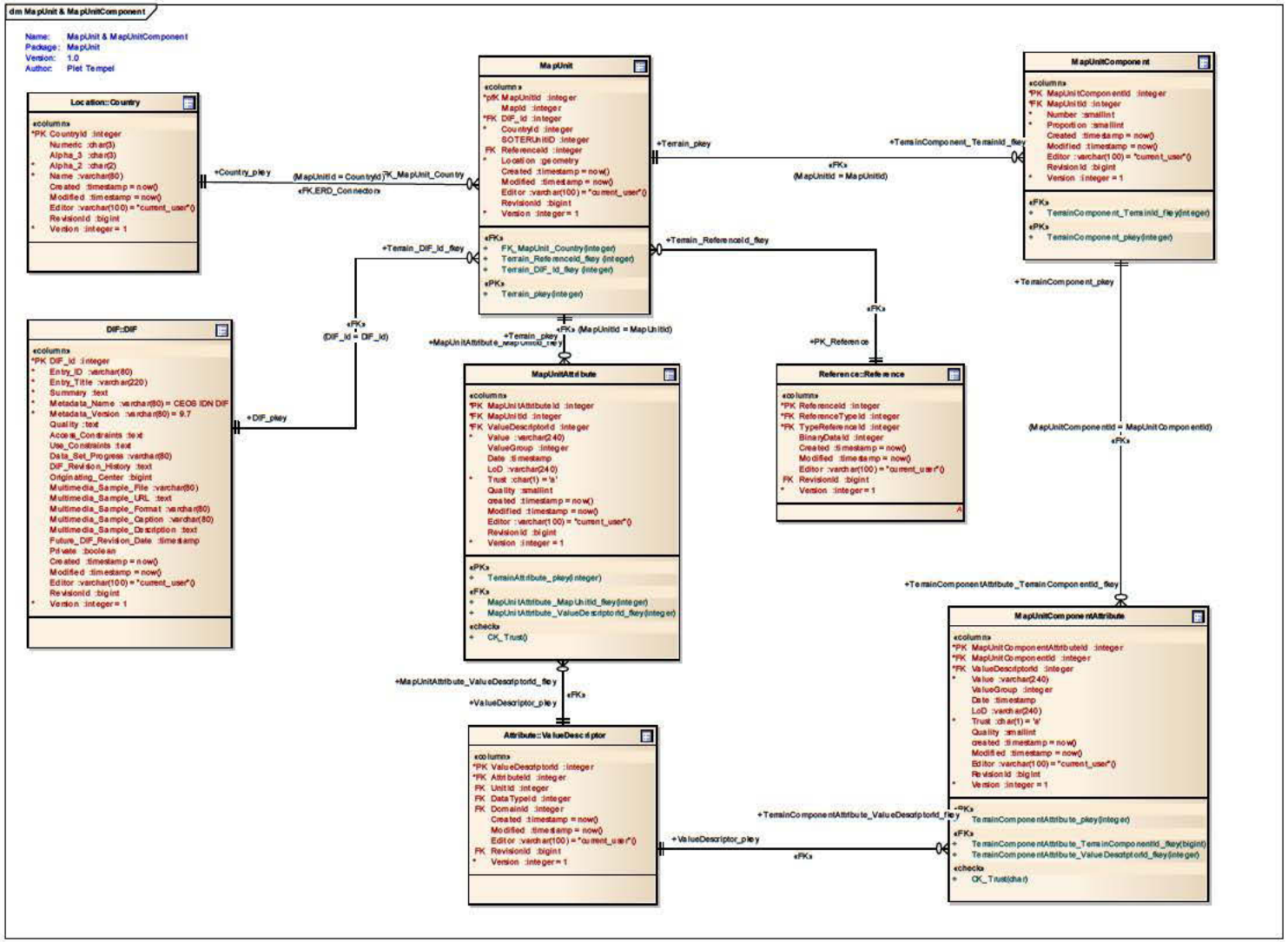

Figure 23

MapUnit & MapUnitComponent diagram.

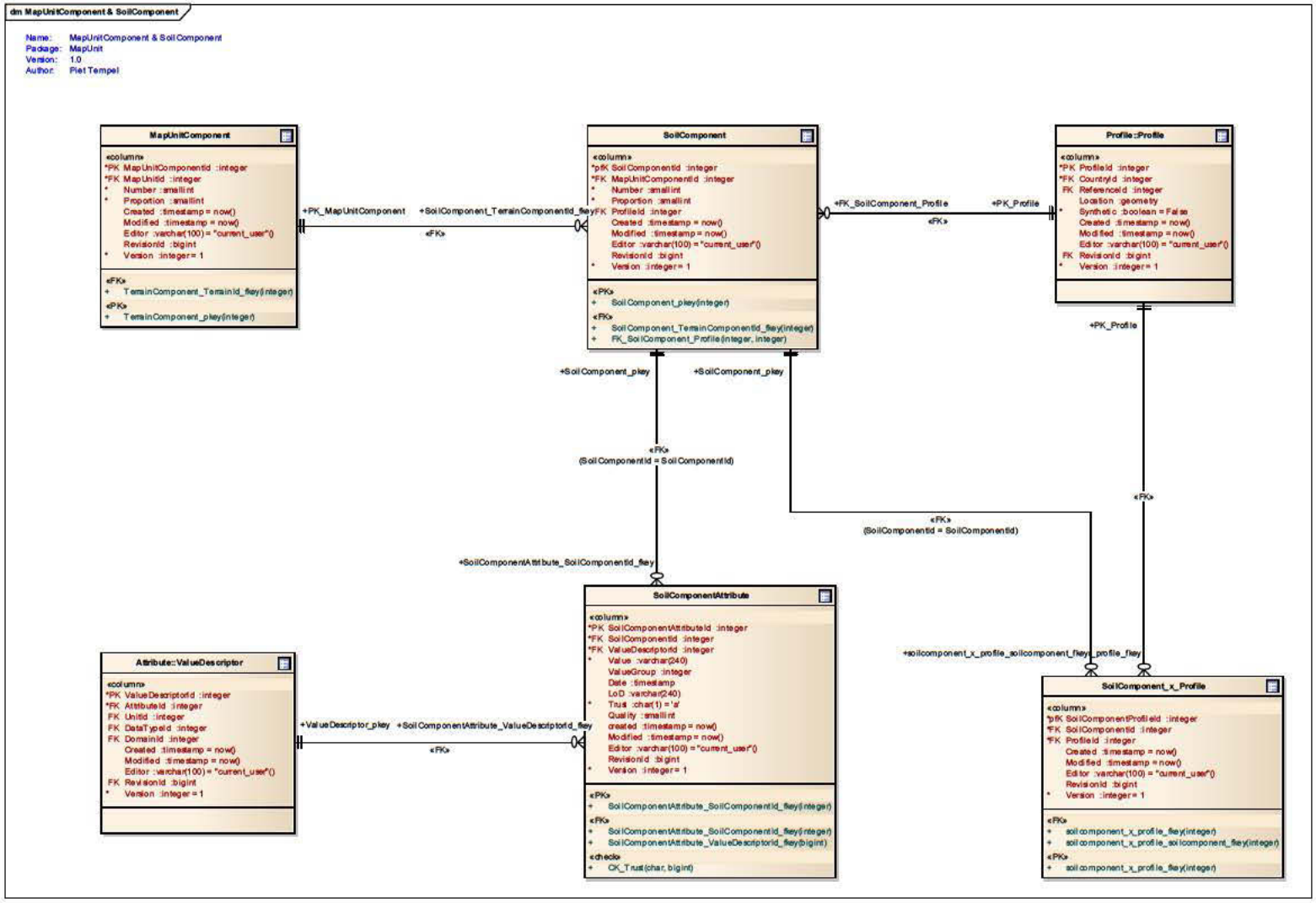

Figure 24

MapUnitComponent & SoilComponent diagram.

MapUnit

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* MapUnit
Detail: *Created on* 23-2-2011. *Last modified on* 29-3-2011.
Notes: Table *MapUnit* stores the polygon geometry (borders) of homogeneous features as well as metadata associated with those features. Homogeneous features (represented on a map by a single colour or pattern) are stored as a single point set union of their polygon geometries.
MapId identifies individual maps within the table, and may be used to link to a catalog application like GeoNetwork. *ReferenceId* links to the source - in table *Reference.Reference* - from which the data were derived for the compilation of the mapping unit. *DIF_Id* enables the linkage of a map to a dataset (in table *DIF.DIF*).
CountryId and *SOTERUnitID* are SOTER map-specific attributes.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	MapUnitId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	DIF_Id	integer	True	False			Reference to an entry in the DIF table that is describing a data set.
False	CountryId	integer	True	False			Reference to an entry in the Country table that represents a country (according to ISO 3166-1:2004).
False	SOTERUnitID	smallint	True	False			The SOTERUnitID is the identification code of a SOTER unit on a SOTER map and in a SOTER database. It links the mapped area to the attributes in the database and in particular, it identifies which terrain belongs to a SOTER unit. SOTER units which have identical attributes carry the same SOTERUnitID. In other words the SOTERUnitID is similar to a code for a mapping unit on a conventional soil map. For each SOTER map, a unique code (up to 4 digits) is assigned to every SOTER unit that has been distinguished. On most SOTER maps 2 or 3 digits will suffice.
False	ReferenceId	integer	False	False			Reference to the source - in table Reference.Reference - from which the data were derived for the compilation of the mapping unit.
False	Location	geometry	True	False			Geographic location of the mapping unit (WGS84).
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Terrain_pkey	Public	MapUnitId		
Terrain_CountryId_fkey	Public	CountryId		
Terrain_ReferenceId_fkey	Public	ReferenceId		
Terrain_DIF_Id_fkey	Public	DIF_Id		
enforce_dims_the_geom	Public	Location	CHECK (st_ndims("Location") = 2)	Only 2D coordinate system allowed
enforce_geotype_the_geom	Public	Location	CHECK (geometrytype("Location") = 'MULTIPOLYGON'::text OR "Location" IS NULL)	Only polygons allowed
enforce_srid_the_geom	Public	Location	CHECK (st_srid("Location") = 4326)	EPSG:4326, geographic coordinates (lat/long) (WSG84)

Relationships

Columns	Association	Notes
(ReferenceId = ReferenceId)	**0..*** **MapUnit**.Terrain_ReferenceId_fkey **1** **Reference**.Reference_pkey	
(MapUnitId = MapUnitId)	**0..*** **MapUnitComponent**.TerrainComponent_TerrainId_fkey **1** **MapUnit**.Terrain_pkey	
(MapUnitId = MapUnitId)	**0..*** **MapUnitAttribute**.MapUnitAttribute_MapUnitId_fkey **1** **MapUnit**.Terrain_pkey	
(CountryId = CountryId)	**0..*** **MapUnit**.Terrain_CountryId_fkey **1** **Country**.Country_pkey	
(DIF_Id=DIF_Id)	**0..*** **MapUnit**.Terrain_DIF_Id_fkey **1** **DIF**.DIF_pkey	

MapUnitAttribute

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* MapUnit
Detail: *Created on* 23-2-2011. *Last modified on* 25-3-2011.
Notes: Table *MapUnitAttribute* stores the values of characteristics that are associated with the homogeneous map features ("mapping units"), in table *MapUnit*. Its *ValueDescriptorId* links each *Value* to a terrain characteristic in table *Atttribute.Attribute*, and - if applicable to a values domain in *Attribute.Domain* (and a unit in table *Attribute.Unit*).
Valuegroup is a numeric to group associated property values for a mapping unit (e.g. degree, and extent for two or more types of erosion in a single mapping unit).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	MapUnitAttributeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	MapUnitId	integer	True	False			Reference to a mapping unit in table *MapUnit*.
False	ValueDescriptorId	integer	True	False			Reference to a ValueDescriptor in the *Attribute.ValueDescriptor* table. A ValueDescriptor links a property to a value domain.
False	Value	varchar	True	False	150		A valid property value.
False	ValueGroup	integer	False	False			Numeric to group associated property values for a mapping unit (e.g. type, degree, and extent of erosion).
False	Date	timestamp	False	False			Date of observation or measurement.
False	LoD	varchar	False	False	150		Level of Determination - accuracy of an observation or measurement.
False	Trust	char	True	False	1	'a'	Level of trust in the given value: 'a' is lowest, 'd' is highest.
False	Quality	bytea	False	False			Quality indicator - 0 is lowest quality level, 255 is highest quality level.
False	created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
TerrainAttribute_pkey	Public	MapUnitAttributeId		
MapUnitAttribute_MapUnitId_fkey	Public	MapUnitId		
MapUnitAttribute_ValueDescriptorId_fkey	Public	ValueDescriptorId		
Check_Trust	Public		"Trust" IN ('a','b','c','d')	

Relationships

Columns	Association	Notes
(MapUnitId = MapUnitId)	**0..*** **MapUnitAttribute**.MapUnitAttribute_MapUnitId_fkey **1** **MapUnit**.Terrain_pkeyt	
	0..* **MapUnitAttribute**.MapUnitAttribute_ValueDescriptorId_fkey **1** **ValueDescriptor**.ValueDescriptor_pkey	

MapUnitComponent

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* MapUnit
Detail: *Created on* 23-2-2011. *Last modified on* 25-3-2011.
Notes: A mapping unit component is a non-mappable constituent of a mapping unit as stored in table *MapUnit*. It has therefore no geometry. It has, however, an extent: a percentage coverage (or *Proportion*) of the MapUnit it is part of. Mapping unit components are sequentially numbered within their enclosing mapping unit, the larger one numbered first.

Obviously, a mapping unit component is mappable if it covers 100% of a mapping unit. Its borders then coincide with borders of its encompassing mapping unit.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	MapUnitComponentId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	MapUnitId	integer	True	False			Reference to a mapping unit in table *MapUnit*.
False	Number	smallint	True	False			Sequence number of the mapping unit component in the mapping unit: the largest mapping unit component comes first, followed by the second in size, and so on.
False	Proportion	smallint	True	False			The proportion that the mapping unit component occupies within the mapping unit. The sum of all mapping unit components within a mapping unit should be 100%.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
TerrainComponent_pkey	Public	MapUnitComponentId		
TerrainComponent_TerrainId_fkey	Public	MapUnitId		
TerrainComponent_Number_check	PublicNumber	"Number" > 0		
TerrainComponent_Proportion_check	Public	Proportion	"Proportion" > 0 AND "Proportion" <= 100	

Relationships

Columns	Association		Notes
(MapUnitId = MapUnitId)	**0..***	**MapUnitComponent**.TerrainComponent_TerrainId_fkey	
	1	**MapUnit**.Terrain_pkey	
(MapUnitComponentId = MapUnitComponentId)	**0..***	**MapUnitComponentAttribute**.TerrainComponentAttribute_TerrainComponentId_fke	
	1	**MapUnitComponent**.TerrainComponent_pkey	
(MapUnitComponentId = MapUnitComponentId)	**0..***	**SoilComponent**.SoilComponent_TerrainComponentId_fkey	
	1	**MapUnitComponent**.TerrainComponent_pkey	

MapUnitComponentAttribute

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* MapUnit
Detail: *Created on* 23-2-2011. *Last modified on* 25-3-2011.
Notes: Table *MapUnitComponentAttribute* stores the values of characteristics that are associated with a mapping unit component in table *MapUnitComponent*. Its *ValueDescriptorId* links each *Value* to a terrain characteristic in table *Atttribute.Attribute*, and - if applicable to a values domain in *Attribute.Domain* (and a unit in table *Attribute.Unit*).
Valuegroup is a numeric to group associated property values for a mapping unit (e.g. duration, and extent for two or more periods of flooding in a single mapping unit).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	MapUnitComponentAttributeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	MapUnitComponentId	integer	True	False			Reference to a mapping unit component in table *MapUnitComponent*.
False	ValueDescriptorId	integer	True	False			Reference to a ValueDescriptor in the *Attribute.ValueDescriptor* table. A ValueDescriptor links a property to a value domain.
False	Value	varchar	True	False	150		A valid property value.
False	ValueGroup	integer	False	False			Numeric to group associated property values for a mapping unit component (e.g. type, degree, and extent of erosion).
False	Date	timestamp	False	False			Date of observation or measurement.
False	LoD	varchar	False	False	150		Level of Determination - accuracy of an observation or measurement.
False	Trust	char	True	False	1	'a'	Level of trust in the given value: 'a' is lowest, 'd' is highest.
False	Quality	bytea	False	False			Quality indicator - 0 is lowest quality level, 255 is highest quality level.
False	created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
TerrainComponentAttribute_pkeyP	Public	MapUnitComponentAttributeId		
TerrainComponentAttribute_TerrainComponentId_fkey	Public	MapUnitComponentId		
TerrainComponentAttribute_ValueDescriptorId_fkeyFK_MapUnitComponentAttribute_ValueDescriptor	Public	ValueDescriptorId		
FK_MapUnitComponentAttribute_Revision	Public	RevisionId		
Trust_check	Public	Trust	"Trust" IN ('a','b','c','d')	

Relationships

Columns	Association	Notes
(MapUnitComponentId = MapUnitComponentId)	**0..*** **MapUnitComponentAttribute**.TerrainComponentAttribute_TerrainComponentId_fkey **1** **MapUnitComponent**.TerrainComponent_pkey	
	0..* **MapUnitComponentAttribute**.TerrainComponentAttribute_ValueDescriptorId_fkey **1** **ValueDescriptor**.ValueDescriptor_pkey	

SoilComponent

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* MapUnit
Detail: *Created on* 23-2-2011. *Last modified on* 25-3-2011.
Notes: A soil component is a constituent of a mapping unit component (as stored in table *MapUnitComponent*), that represents a single soil. Since a mapping unit component is a non-mappable constituent of a mapping unit, a soil component does not have a geometry of its own either. It has, however, an extent: the percentage coverage (or *Proportion*) of the mapping unit it is part of (via its mapping unit component). Soil components are sequentially numbered within their enclosing mapping unit, the larger one numbered first. The percentage coverage of all soil components within a mapping unit component should add up exactly to the percentage coverage of that mapping unit component within its mapping unit.
Obviously, a mapping unit component is mappable if it covers 100% of a mapping unit. Its borders then coincide with borders of its encompassing mapping unit
For every soil component at least one, but preferably more, fully described and analysed reference profiles should be available from existing soil information sources. Following judicious selection, one of these reference profiles will be designated as the representative profile for the soil component. It is referenced by *ProfileId* in table *Profile.Profile.*

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	SoilComponentId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	MapUnitComponentId	integer	True	False			Reference to a mapping unit component in table *MapUnitComponent*.
False	Number	smallint	True	False			Sequence number of the soil component in the mapping unit component: the largest component comes first, followed by the second in size, and so on.
False	Proportion	smallint	True	False			The proportion that the soil component occupies within the mapping unit. The sum of all soil components within a mapping unit component should be equal to the proportion of the mapping unit component in its mapping unit. The sum of all soil components within a mapping unit should be 100%.
False	ProfileId	bigint	True	False			Reference to a profile in table *Profile.Profile*, which is designated as the representative profile for the soil component.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
SoilComponent_pkey	Public	SoilComponentId		
SoilComponent_TerrainComponentId_fkey	Public	MapUnitComponentId		
FK_SoilComponent_Profile	Public	ProfileId		
SoilComponent_Number_check	Public	Number	"Number" > 0	
SoilComponent_Proportion_check	Public	Proportion	“Proportion” >0 AND “Proportion” <= 100	

Relationships

Columns	Association		Notes
(SoilComponentId= SoilComponentId)	**0..***	**SoilComponentAttribute**.SoilComponentAttribute_SoilComponentId_fkey	
	1	**SoilComponent**.SoilComponent_pkey	
(ProfileId = ProfileId)	**0..***	**SoilComponent**.FK_SoilComponent_Profile	
	1	**Profile**.Profile_pkey	
(MapUnitComponentId = MapUnitComponentId)	**0..***	**SoilComponent**.SoilComponent_TerrainComponentId_fkey	
	1	**MapUnitComponent**.TerrainComponent_pkey	

SoilComponentAttribute

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* MapUnit
Detail: *Created on* 23-2-2011. *Last modified on* 25-3-2011.
Notes: Table *SoilComponentAttribute* stores the values of characteristics that are associated with a soil component in table *SoilComponent*. Its *ValueDescriptorId* links each *Value* to a terrain characteristic in table *Atttribute.Attribute*, and - if applicable to a values domain in *Attribute.Domain* (and a unit in table *Attribute.Unit*).
Valuegroup is a numeric to group associated property values for a mapping unit (e.g. size and surface cover for two types of surface rockiness in a single soil component).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	SoilComponentAttributeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	SoilComponentId	integer	True	False			Reference to a soil component in table *SoilComponent*.
False	ValueDescriptorId	integer	True	False			Reference to a ValueDescriptor in the *Attribute.ValueDescriptor* table. A ValueDescriptor links a property to a value domain.
False	Value	varchar	True	False	150		A valid property value.
False	Date	timestamp	False	False			Date of observation or measurement.
False	LoD	varchar	False	False	150		Level of Determination - accuracy of an observation or measurement.
False	Trust	bpchar	True	False	1	'a'	Level of trust in the given value: 'a' is lowest, 'd' is highest.
False	Quality	bytea	False	False			Quality indicator - 0 is lowest quality level, 255 is highest quality level.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
SoilComponentAttribute_SoilComponentId_fkey	Public	SoilComponentId		
SoilComponentAttribute_ValueDescriptorId_fkey	Public	ValueDescriptorId		
Check_Trust	Public	Trust	"Trust" IN ('a','b','c','d')	

Relationships

Columns	Association		Notes
(SoilComponentId = SoilComponentId)	**0..***	**SoilComponentAttribute**.SoilComponentAttribute_SoilComponentId_fkey	
	1	**SoilComponent**.SoilComponent_pkey	
	0..*	**SoilComponentAttribute**.SoilComponentAttribute_ValueDescriptorId_fkey	
	1	**ValueDescriptor**.ValueDescriptor_pkey	

SoilComponent_x_Profile

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* MapUnit
Detail: *Created on* 16-3-2011. *Last modified on* 28-4-2011.
Notes: The *SoilComponent_x_Profile* table links soil components, listed in the *SoilComponent* table, to one or more reference profiles in the *Profile.Profile* table. This link table thus shows a certain range in soil variability within a soil component.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	SoilComponentProfileId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	SoilComponentId	bigint	True	False			Reference to a soil component in table *SoilComponent*.
False	ProfileId	integer	True	False			Reference to a profile in table *Profile.Profile*.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
soilcomponent_x_profile_fkey	Public	ProfileId		
soilcomponent_x_profile_soilcomponent_fkey	Public	SoilComponentId		

Relationships

Columns	Association	Notes
(SoilComponentId = SoilComponentId)	**0..*** **SoilComponent_x_Profile**.soilcomponent_x_profile_soilcomponent_fkey **1** **SoilComponent**.SoilComponent_pkey	
(ProfileId = ProfileId)	**0..*** **SoilComponent_x_Profile**.soilcomponent_x_profile_fkey **1** **Profile**.Profile_pkey	

Profile

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 13-12-2010. Last modified on 28-3-2011.*
Notes: The Profile schema is fundamental to the core database. Its relations describe two basic entities from the domain of discourse underlying the database: a *profile* ("pedon") and its properties (land use, position in the terrain, signs of erosion, etc.), and its constituent *horizons*, each horizon with its own properties (structure, colour, texture, etc.). This yields a rather simple schema of relations.

Profile descriptions may be included in more than one data set, using different identifiers (and subsets of properties). Using the Profile schema, data sets can be reconstructed, using the original identifiers.

Since available soil moisture plays a prevalent role in many applications, it has been made a database object in its own right. Otherwise, soil moisture tensions must be derived from their analytical methods, requiring arbitrary string parsing in queries.

See Figure 25-27.

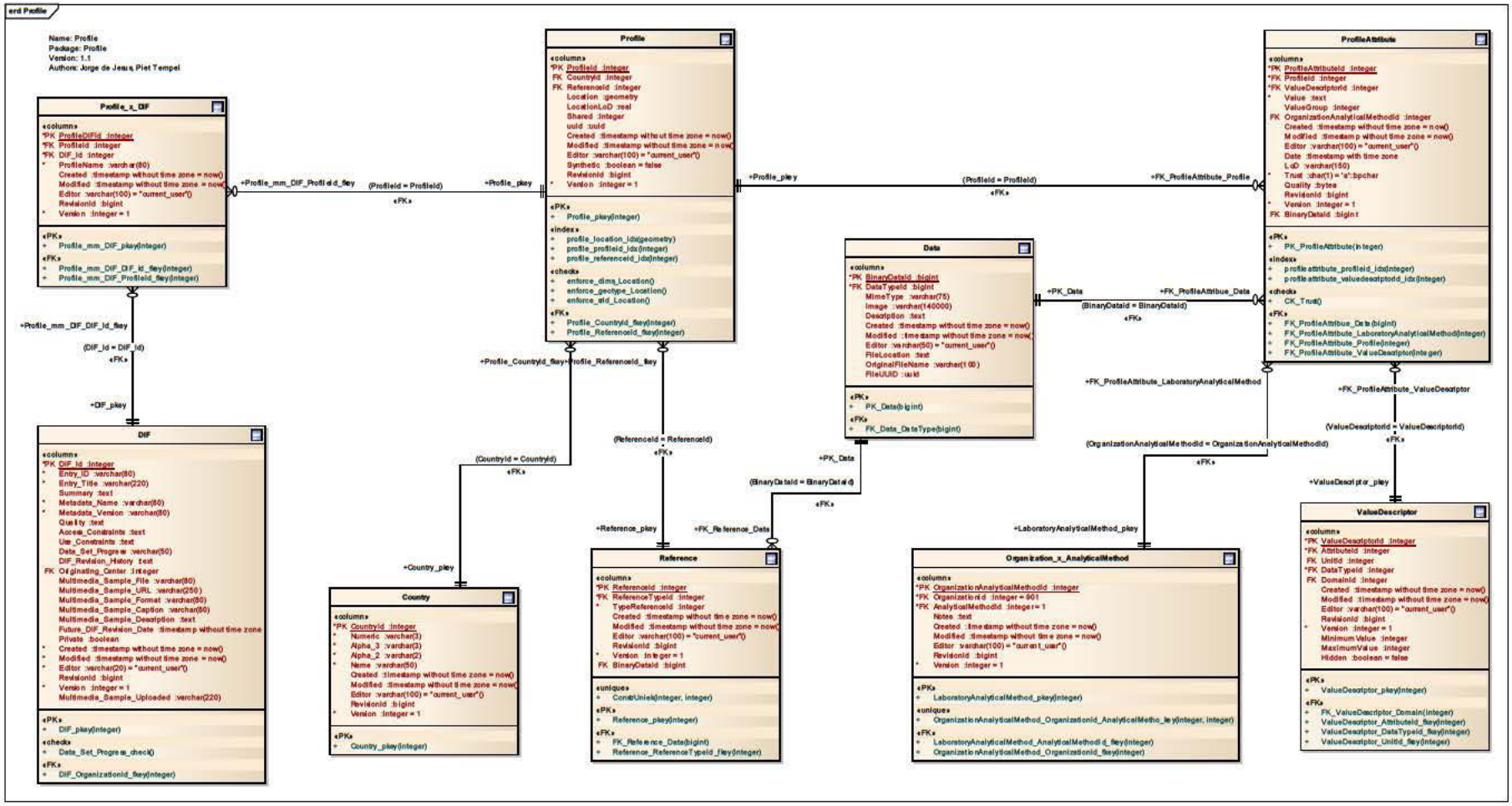

Figure 25

Profile schema.

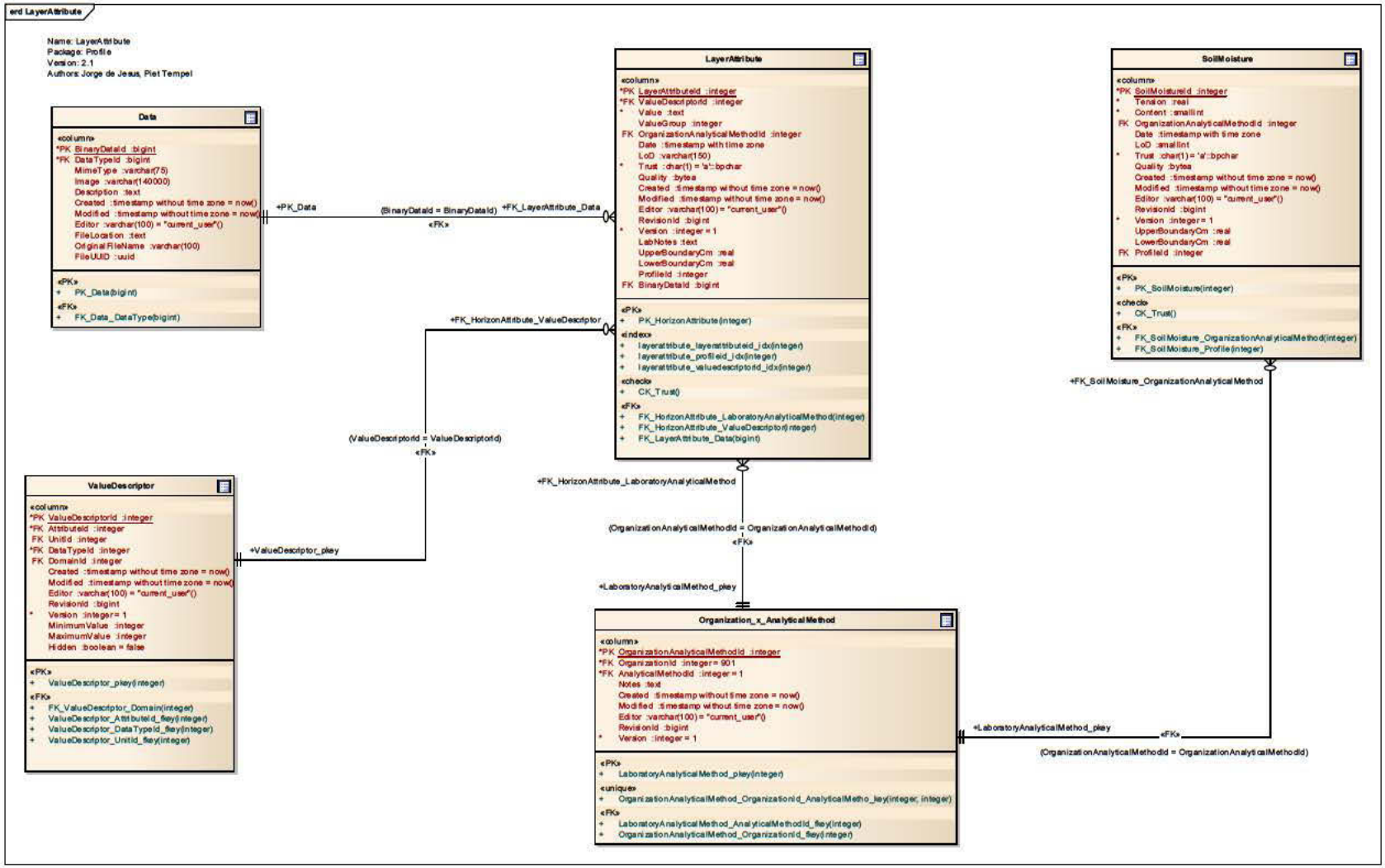

Figure 26
LayerAttribute diagram.

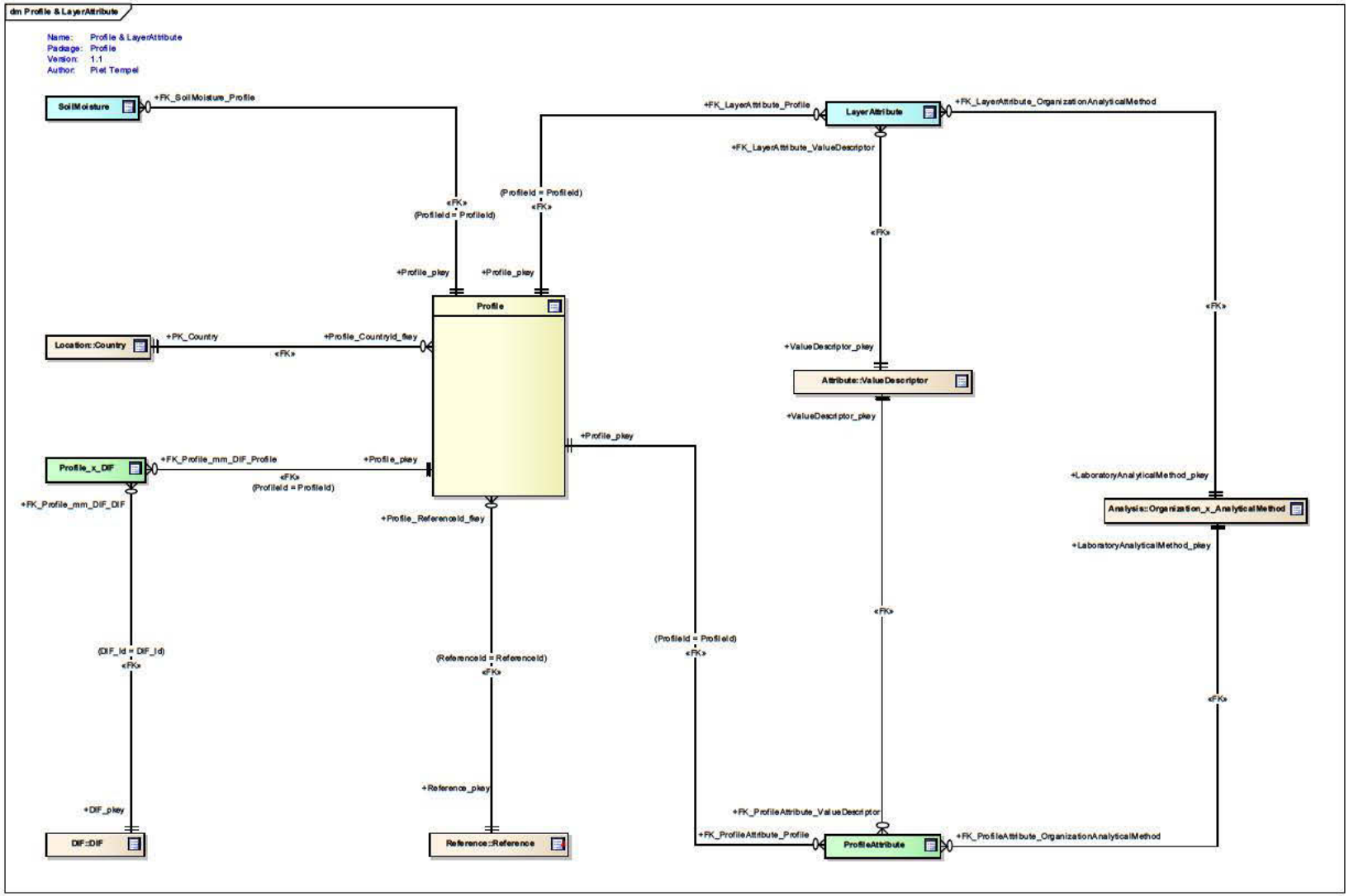

Figure 27
Profile and LayerAttribute diagram.

LayerAttribute

Database: *PostgreSQL*, *Stereotype:* «table», *Package: Profile*
Detail: *Created on* 13-12-2010. *Last modified on* 26-5-2011.
Notes: Table *LayerAttribute* stores the values of any characteristic associated with a single profile layer.
Its *ValueDescriptorId* links each property's Value to a layer characteristic in table *Atttribute.Attribute*, and - if applicable - to a values domain in *Attribute.Domain* (and a unit in table *Attribute.Unit*).
Valuegroup is a numeric to group associated property values for a layer (e.g. colour, size, and distinctness for two or more types of mottles).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	LayerAttributeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileId	integer	True	False			Reference to a profile in the *Profile.Profile* table.
False	BinaryDataId	integer	False	False			Reference to Binary.Data
False	UpperBoundaryCm	real	False	False			Depth to upper layer boundary (centimetres).
False	LowerBoundaryCm	real	False	False			Depth to lower layer boundary (centimetres).
False	ValueDescriptorId	integer	True	True			Reference to a ValueDescriptor in the *Attribute.ValueDescriptor* table. A ValueDescriptor links a property to a value domain.
False	Value	text	True	True			A valid property value.
False	ValueGroup	integer	False	True			Numeric to group associated property values for a horizon (e.g. colour, size, and distinctness of mottles).
False	OrganizationAnalyticalMethodId	integer	False	False			Reference to an entry in the *Organization_x_AnalyticalMethod* table: that is, a laboratory and an analytical procedure.
False	Date	timestamp	False	False			Date of observation or measurement.
False	LoD	varchar	False	False150			Level of Determination - accuracy of an observation or measurement.
False	Trust	char	False	False	1	'a'	Level of trust in the given value: 'a' is lowest, 'd' is highest.
False	Quality	bytea	False	False			Quality indicator - 0 is lowest quality level, 255 is highest quality level.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_LayerAttribute	Public	LayerAttributeId		
FK_LayerAttribute_Profile	Public	ProfileId		
FK_LayerAttribute_ValueDescriptor	Public	ValueDescriptorId		
FK_LayerAttribute_LaboratoryAnalyticalMethod	Public	OrganizationAnalyticalMethodId		
FK_LayerAttribute_Data	Public	BinaryDataId		
CK_Trust	Public		"Trust" IN ('a','b','c','d')	

Relationships

Columns	Association	Notes
(ProfileId = ProfileId)	**0..*** **LayerAttribute**.FK_LayerAttribute_Profile **1** **Profile**.Profile_pkey	
(ValueDescriptorId=ValueDescriptorId)	**0..*** **LayerAttribute**.FK_LayerAttribute_ValueDescriptor **1** **ValueDescriptor**.ValueDescriptor_pkey	
(OrganizationAnalyticalMethodId=OrganizationAnalyticalMethodId)	**0..*** **LayerAttribute**.FK_LayerAttribute_LaboratoryAnalyticalMethod **1** **Organization_x_AnalyticalMethod**.LaboratoryAnalyticalMethod_pkey	
(BinaryId=BinaryId)	**0..*** **LayerAttribute**.BinaryDataId **1** **Binary**.PK_Data	

Profile

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Profile
Detail: *Created on* 13-12-2010. *Last modified on* 28-3-2011.
Notes: The *Profile* table stores soil profiles, either synthetic or real-world, along with their location. The *Profile* table is a pivotal table in the Core database.
Information stored in this table is rather summary, values for profile characteristics are stored in the *ProfileAttribute* table.
To associate synthetic profiles, or profiles with an unknown location, with a minimal georeference, a country reference has been included.
Each profile is also associated with a data source - publication, map, web site, etc.
The *Synthetic* column denotes whether a profile is synthetic or not.
The *Shared* column denotes whether a profile is to be shared or not.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ProfileId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	CountryId	integer	True	False			Reference to an entry in the Country table that represents a country (according to ISO 3166-1:2004).
False	ReferenceId	integer	False	False			Reference to the source - in table Reference.Reference - from which the data were derived.
False	Location	geometry	False	False			Geographic location of the profile (WGS84).
False	Synthetic	boolean	True	False		False	If true, the profile is an artificial profile - i.e. non-existent in the real world.
False	LocationLoD	bigint	False	False			Uncertainty related to the Location. Numerical value
False	uuid	uuid	False	False			Universally Unique Identifier. UUID is automatically generated if profile is inserted without one.
False	Shared	bool	False	False			Boolean indicating if profile can be shared. Public available
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Profile_pkey	Public	ProfileId		
Profile_CountryId_fkey	Public	CountryId		
Profile_ReferenceId_fkey	Public	ReferenceId		
enforce_dims_Location	Public	Location	ndims('Location') = 2	PostGIS default check
enforce_srid_Location	Public	Location	srid("Location") = 4326	PostGIS default check for EPSG:4326 (lat/long)
enforce_geotype_Location"	Public	Location	geometrytype('Location') = 'POINT'::text OR "Location" IS NULL	DB only allows for point data to be entered. Other spatial objects are not allowed

Relationships

Columns	Association		Notes
(CountryId = CountryId)	**0..***	**Profile**.Profile_pkey	
	1	**Country**.Country_pkey	
	0..*	**Profile_x_DIF**.Profile_x_DIF_ProfileId_fkey	
	1	**Profile**.profile_pkey	
(ProfileId = ProfileId)	**0..***	**SoilComponent**.FK_SoilComponent_Profile	
	1	**Profile**.PK_Profile	
		<anonymous>.	
		Profile.	
(ProfileId = ProfileId)	**0..***	**ProfileAttribute**.FK_ProfileAttribute_Profile	
	1	**Profile**.PK_Profile	
(ReferenceId=ReferenceId)	**0..***	Reference.Reference_pkey	
	1	Profile.profile_pkey	

ProfileAttribute

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Profile
Detail: *Created on* 13-12-2010. *Last modified on* 28-3-2011.
Notes: Table *ProfileAttribute* stores the values of characteristics that are associated with the profile's site as well as its identification.
Its *ValueDescriptorId* links each property *Value* to a site characteristic in table *Atttribute.Attribute*, and - if applicable - to a values domain in *Attribute.Domain* (and a unit in table *Attribute.Unit*).
Valuegroup is a numeric to group associated property values for a mapping unit (e.g. degree, and extent for two or more types of erosion at the profile site).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ProfileAttributeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileId	integer	True	False			Reference to a profile in the *Profile.Profile* table.
False	ValueDescriptorId	integer	True	True			Reference to a ValueDescriptor in the *Attribute.ValueDescriptor* table. A ValueDescriptor links a property to a value domain.
False	BinaryDataId	integer	False	False			Reference to a binary content in Binary.Data table
False	Value	text	True	True			A valid property value.
False	ValueGroup	integer	False	True			Numeric to group associated property values for a profile (e.g. type and extent of vegetation at the site).
False	OrganizationAnalyticalMethodId	integer	False	False			Reference to an entry in the *Organizationx_AnalyticalMethod* table: that is, a laboratory and an analytical procedure.
False	Date	timestamp	False	False			Date of observation or measurement.
False	LoD	varchar	False	False	240		Level of Determination - accuracy of an observation or measurement.
False	Trust	char	True	False	1	'a'	Level of trust in the given value: 'a' is lowest, 'd' is highest.
False	Quality	bytea	False	False			Quality indicator - 0 is lowest quality level, 255 is highest quality level.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_ProfileAttribute	Public	ProfileAttributeId		
FK_ProfileAttribute_LaboratoryAnalyticalMethod	Public	OrganizationAnalyticalMethodId		
FK_ProfileAttribute_Profile	Public	ProfileId		
FK_ProfileAttribute_ValueDescriptor	Public	ValueDescriptorId		
FK_ProfileAttribute_Revision	Public	RevisionId		
FK_ProfileAttribue_Data	Public	BinaryDataId		
CK_Trust	Public		"Trust" IN ('a','b','c','d')	

Relationships

Columns	Association	Notes
	0..* **ProfileAttribute**.FK_ProfileAttribute_ValueDescriptor **1** **ValueDescriptor**.ValueDescriptor_pkey	
	0..* **ProfileAttribute**.FK_ProfileAttribute_OrganizationAnalyticalMethod **1** **Organization_x_AnalyticalMethod**.LaboratoryAnalyticalMethod_pkey	
(ProfileId = ProfileId)	**0..*** **ProfileAttribute**.FK_ProfileAttribute_Profile **1** **Profile**.PK_Profile	

Profile_x_DIF

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Profile
Detail: *Created on* 8-2-2011. *Last modified on* 21-4-2011.
Notes: The *Profile_x_DIF* table links profiles, listed in the *Profile* table to one or more data sets in the *DIF.DIF* table. Thus, it shows the various data sets a profile is part of.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ProfileDIFId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileId	integer	True	True			Reference to a profile in the *Profile.Profile* table.
False	DIF_Id	integer	True	True			Reference to an entry in the DIF table that is describing a data set.
False	ProfileName	varchar	True	False	80		Dataset identifier ("name") for the profile.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_Profile_x_DIF	Public	ProfileDIFId		
FK_Profile_x_DIF_DIF	Public	DIF_Id		
FK_Profile_x_DIF_Profile	Public	ProfileId		
UQ_Profile_DIF	Public	ProfileId DIF_Id		

Relationships

Columns	Association	Notes
(ProfileId = ProfileId)	**0..*** **LocalClassification.FK_LocalClassification_Profile_x_DIF** **1** **Profile_x_DIF.PK_Profile_x_DIF**	
	0..* **Profile_x_DIF**.FK_Profile_x_DIF_Profile **1** **Profile**.PK_Profile	

SoilMoisture

Database: *PostgreSQL*, *Stereotype:* «table», *Package: Profile*
Detail: *Created on* 14-12-2010. *Last modified on* 26-5-2011.
Notes: Table *SoilMoisture* stores soil moisture tension (in KPa) and the resulting soil moisture content (vol%) for a soil layer. *OrganizationAnalyticalMethodId* links an analytical method and laboratory to each measurement. Since soil moisture tension is part of the observation that is stored in table SoilMoisture, the associated analytical method may be described in generic terms.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	SoilMoistureId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ProfileId	integer	True	False			
False	UpperBoundaryCm	real	False	False			Depth to upper layer boundary (centimetres).
False	LowerBoundaryCm	real	False	False			Depth to lower layer boundary (centimetres).
False	Tension	real	True	False			Soil moisture tension (in KPa, zero or less)
False	Content	smallint	True	False			Soil moisture content (vol%).
False	OrganizationAnalyticalMethodId	integer	False	False			Reference to an entry in the *Organization_x_AnalyticalMethod* table: that is, a laboratory and an analytical procedure.
False	Date	timestamp	False	False			Date of observation or measurement.
False	Lod	smallint	False	False			Level of Determination - accuracy of an observation or measurement.
False	Trust	char	True	False	1	'a'	Level of trust in the given value: 'a' is lowest, 'd' is highest.
False	Quality	bytea	False	False			Quality indicator - 0 is lowest quality level, 255 is highest quality level.
False	Created	timestamp	False	False		now()	Date of creation of the record.
False	Modified	timestamp	False	False		now()	Date of last modification of the record.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
FK_SoilMoisture_Profile	Public	ProfileId		
PK_SoilMoisture	Public	SoilMoistureId		
FK_Moisture_OrgAnalyticalMethod	Public	OrganizationAnalyticalMethodId		
CK_Trust	Public		"Trust" IN ('a','b','c','d')	

Relationships

Columns	Association	Notes
(ProfileId = ProfileId)	**0..*** **SoilMoisture**.FK_SoilMoisture_Profile **1** **Profile**.PK_Profile	

Reference

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 14-12-2010. Last modified on 29-3-2011.*
Notes: Data, definitions, and descriptions may be drawn from a variety of data and information sources. Potential sources are: publications, maps, web sites (URL's), organizations, and digital media. As can be seen, these sources vary widely in nature, and in the way they are described. The *References* schema enables a harmonized way to refer to these heterogeneous sources.

Most other schemas somehow rely on this schema. Schema *Reference* on the other hand, is the only schema that does not depend on any other schema.

See Figure 28.

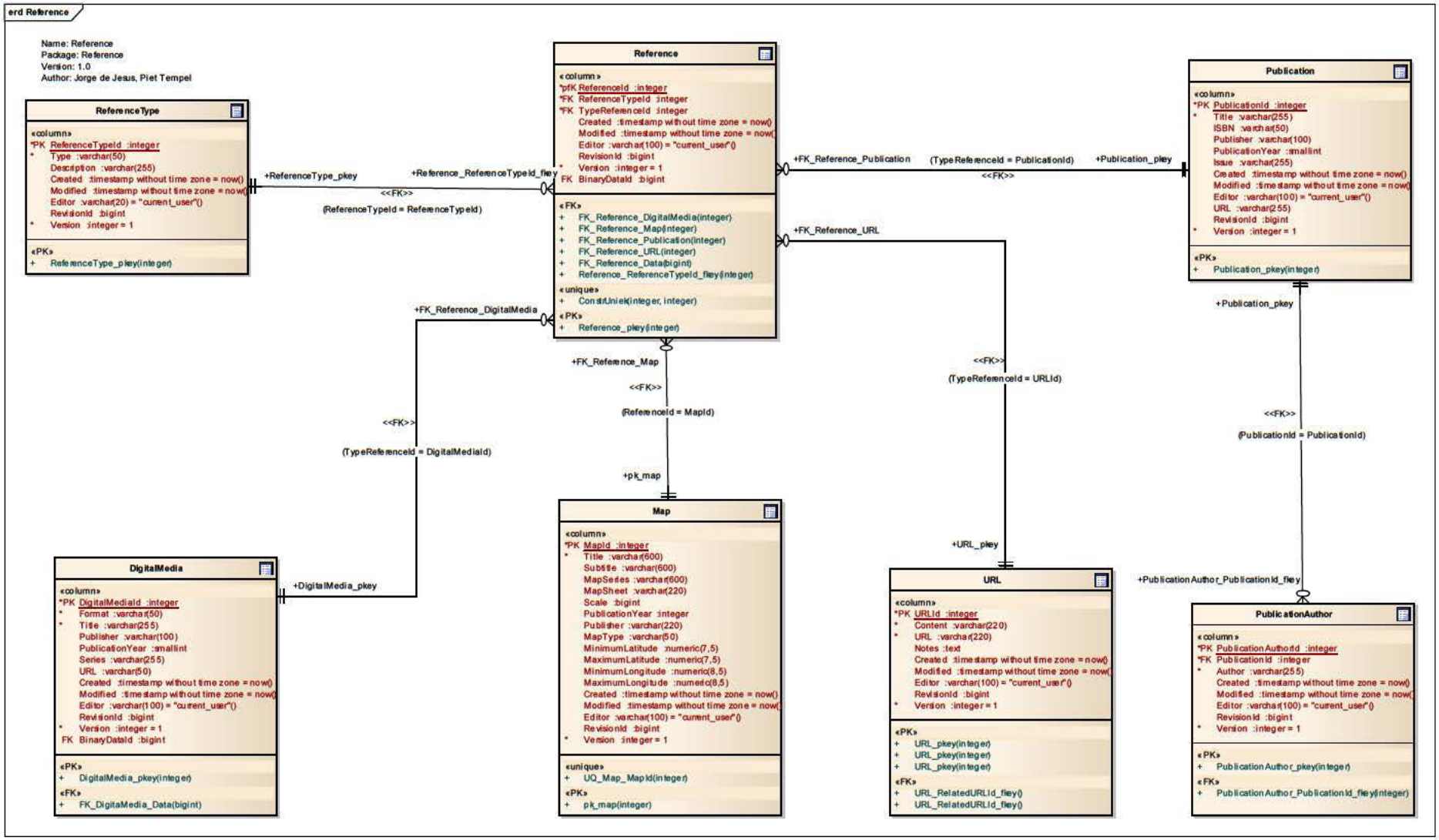

Figure 28
Reference schema.

DigitalMedia

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Reference
Detail: *Created on* 14-12-2010. *Last modified on* 29-3-2011.
Notes: Digital media includes any storage device that holds digital data (magnetic disk, magnetic tape, optical disc (CD, DVD), etc.). That is, any storage device of *ReferenceType* 'Digital media'.
Table *DigitalMedia* stores common information like title, publisher, and publication year of a digital media release.
'Digital media' is one of the reference types listed in table *ReferenceType*.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	DigitalMediaId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	BinaryDataId	integer	False	False			BinaryDataId reference
False	Format	varchar	True	False	50		Shape file, (geographic) database, flat file, PDF, etc.
False	Title	varchar	True	False	255		The title of the digital data set
False	Publisher	varchar	False	False	100		Publisher of the digital data set.
False	PublicationYear	smallint	False	False			Year of publication of the digital data set.
False	Series	varchar	False	False	255		If applicable
False	URL	varchar	False	False	50		
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
DigitalMedia_pkey	Public	DigitalMediaId		
FK_DigitaMedia_Data	Public	RevisionId		

Relationships

Columns	Association	Notes
(TypeReferenceId = DigitalMediaId)	**0..*** **Reference**.Reference_ReferenceTypeId_fkey **1** **DigitalMedia**.DigitalMedia_pkey	
(BinaryDataId=BinaryDataId)	**0..*** **Data**.PK_Data **1** **DigitalMedia**.DigitalMedia_pkey	

Map

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Reference
Detail: *Created on* 14-12-2010. *Last modified on* 29-3-2011.
Notes: Table *Map* describes printed (source) maps. These maps may have been used to derive data for the compilation of map units (stored in *MapUnit.MapUnit*) or profile descriptions.
Table *Map* stores common information like title, publisher, publication year and map extent.
"Map" is one of the reference types listed in table *ReferenceType*.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	MapId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Title	varchar	True	False	600		
False	Subtitle	varchar	False	False	600		
False	MapSeries	varchar	False	False	600		
False	MapSheet	varchar	False	False	220		
False	Scale	bigint	False	False	0		
False	PublicationYear	integer	False	False	0		
False	Publisher	varchar	False	False	220		
False	MapType	varchar	False	False	50		
False	MinimumLatitude	numeric	False	False	(7,5)		
False	MaximumLatitude	numeric	False	False	(7,5)		
False	MinimumLongitude	numeric	False	False	(8,5)		
False	MaximumLongitude	numeric	False	False	(8,5)		
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
pk_map	Public	MapId		

Relationships

Columns	Association	Notes
(TypeReferenceId = MapId)	**0..*** **Reference**.Reference_ReferenceTypeId_fkey **1** **Map**.pk_map	

Publication

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Reference
Detail: *Created on* 14-12-2010. *Last modified on* 29-3-2011.
Notes: Table *Publication* describes any content that has been made available to the public, usually in the form of text, images, or other visual content, and predominantly on paper (books, manuals, journals, articles, etc.). These publications may have been used to derive data for the compilation of profile descriptions, or to define soil and terrain characteristic, and analytical methods.

Table *Publication* stores common information like title, publisher, publication year and ISBN. Publication authors are stored in a separate *PublicationAuthor* table. 'Publication' is one of the reference types listed in table *ReferenceType*.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	PublicationId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Title	varchar	True	False	255		
False	ISBN	varchar	False	False	50		
False	Publisher	varchar	False	False	100		
False	PublicationYear	smallint	False	False	0		
False	Issue	varchar	False	False	255		
False	URL	varchar	False	False	255		
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
Publication_pkey	Public	PublicationId		

Relationships

Columns	Association	Notes
(PublicationId = PublicationId)	**0..*** **PublicationAuthor**.PublicationAuthor_PublicationId_fkey **1** **Publication**.Publication_pkey	
(TypeReferenceId = PublicationId)	**0..*** **Reference**.Reference_pkey **1** **Publication**.PK_Publication	

PublicationAuthor

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Reference
Detail: *Created on* 14-12-2010. *Last modified on* 29-3-2011.
Notes: Table *PublicationAuthor* links author names to publications (in table *Publication*). A single author may be linked to more than one publication, and a publication may have more than one author.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	PublicationAuthorId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	PublicationId	integer	True	False			Reference to a publication in table *Publication*.
False	Author	varchar	True	False	255		Name of the author: *Surname, Initial_1.Initial_2.Initial_3.*
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
PublicationAuthor_pkey	Public	PublicationAuthorId		
PublicationAuthor_PublicationId_fkey	Public	PublicationId		

Relationships

Columns	Association	Notes
(PublicationId = PublicationId)	**0..*** **PublicationAuthor**.PublicationAuthor_pkey **1** **Publication**.Publication_pkey	

Reference

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Reference
Detail: *Created on* 14-12-2010. *Last modified on* 29-3-2011.
Notes: The *Reference* table lists references - sources of data and information. A reference may be either of reference type:

- Publication (from table Publication)
- Organization (from table Contact.Organization)
- Map (from table Map)
- Web site (from table URL)
- Other (Any other data source - still to implemented)
- Digital media (from table DigitalMedia)

A reference consists of a *referenceType* (pointing the reference to the appropriate table) and a *TypeReference* (a *MapId*, *PublicationId*, *URLId*, etc., in accordance with the *ReferenceType*). Table Reference harmonizes reference to a variety of data and information resources.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ReferenceId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	ReferenceTypeId	integer	True	False			Reference to a reference type (digital media, map, publication, URL) in table *ReferenceType*.
False	TypeReferenceId	integer	True	False	0		Reference to a data source / authoritative source in the appropriate type of reference table - *DigitalMedia*, *Map*, *Publication*, *URL* or *Contact.Organization*
False	DataTypeId	integer	True	False			Reference to binary data in *Binary.Data*
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
DigitalMedia_pkey	Public	TypeReferenceId		
pk_map	Public	TypeReferenceId		
Publication_pkey	Public	TypeReferenceId		
URL_pkey	Public	TypeReferenceId		
Reference_pkey	Public	ReferenceId		
Reference_ReferenceTypeId_fkey	Public	ReferenceTypeId		
PK_Data	Public	DataTypeId		

Relationships

Columns	Association	Notes
(ReferenceTypeId = ReferenceTypeId)	**0..*** **Reference**.Reference_ReferenceTypeId_fkey **1** **ReferenceType**.ReferenceType_pkey	
	0..* **AnalyticalMethod**.AnalyticalMethod_ReferenceId_fkey **1** **Reference**.Reference_pkey	
	0..* **Attribute**.Attribute_ReferenceId_fkey **1** **Reference**.Reference_pkey	
	0..* **Domain**.FK_Domain_Reference **1** **Reference**.Reference_pkey	
(TypeReferenceId = URLId)	**0..*** **Reference**.FK_Reference_URL **1** **URL**.URL_pkey	
(TypeReferenceId = MapId)	**0..*** **Reference**.FK_Reference_Map **1** **Map**. pk_map	
(TypeReferenceId = DigitalMediaId)	**0..*** **Reference**.FK_Reference_DigitalMedia **1** **DigitalMedia**.DigitalMedia_pkey	
(TypeReferenceId = PublicationId)	**0..*** **Reference**.FK_Reference_Publication **1** **Publication**.Publication_pkey	
(ReferenceId = ReferenceId)	**0..*** **MapUnit**.FK_MapUnit_Reference **1** **Reference**.PK_Reference	
	0..* **Profile**.FK_Profile_Reference **1** **Reference**.Reference_pkey	

	0..* **BibliographicCitation**.BibliographicCitation_ReferenceId_fkey **1** **Reference**.Reference_pkey	
	0..* **Region.**.FK_Region_Reference **1** **Reference**.Reference_pkey	
(BinaryDataId=BinaryDataId)	**0..*** **Reference.FK_Data** **1** **Data.BinaryDataId**	

ReferenceType

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Reference
Detail: *Created on* 14-12-2010. *Last modified on* 29-3-2011.
Notes: Database tables like *Attribute.Attribute* and *Profile.Profile* may refer to various data and information sources: Publications, Maps, web sites (URL's), Organizations, Digital media, and others. Table *ReferenceType* contains the master data for these sources - a name and a description.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ReferenceTypeId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Type	varchar	True	False	50		Reference type name.
False	Description	varchar	False	False	255		Reference type description.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
ReferenceType_pkey	Public	ReferenceTypeId		

Relationships

Columns	Association	Notes
(ReferenceTypeId = ReferenceTypeId)	**0..*** **Reference**.Reference_pkey **1** **ReferenceType**.ReferenceType_pkey	

URL

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Reference
Detail: *Created on* 16-3-2011. *Last modified on* 29-3-2011.
Notes: Table *URL* lists Universal Resource Locators ("web site addresses") as data and information sources for various database tables. "URL" is one of the reference types listed in table *ReferenceType*.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	URLId	integer	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Content	varchar	True	False	220		Content / information to be found at URL.
False	URL	varchar	True	False	220		Universal Resource Locator ("internet site address") includes protocol: "http://www....."
False	Notes	text	False	False			Additional information
False	Created	timestamp	False	False		now)'	Date of record creation.
False	Modified	timestamp	False	False		now)'	Date of last record modification.
False	Editor	varchar	False	False	100	'current_user()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

	Public	URLId		

Relationships

Columns	Association		Notes
(TypeReferenceId = URLId)	**0..***	**Reference**.FK_Reference_URL	
	1	**URL**.URL_pkey	

Binary

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* Binary
Detail: *Created on* 01-11-2012. *Last modified on* 01-11-2012.
Notes: The *Data* table contains reference to binary data, e.g. images.
Data entrance is done using the function 'Binary'.'importImg'('fileImage'
' text, 'descriptionText' text DEFAULT NULL::text, 'storageLoc' text DEFAULT '/opt/imageStorage'::text). The function returns a BinaryDataId that is used in other tables.
Direct data in the table is not recommended and data introduction should be done in SQL queries in other tables that require a BinaryDataId key value.
Original files are stored in the file system and given a UUID that replaces the file name.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	BinaryDataId	bigint	True	False			Unique identifier for each possible record in the table
False	DataTypeId	bigint	True	False			DataTypeId value defined in Attribute.DataType for binary data (normally it should be 7)
False	MimeType	bigint	False	False	75		MimeType is assigned automatically using Python's FileMagic library (http://pypi.python.org/pypi/filemagic). The python library is based on C/C++ libmagic (Unix systems: file command)
False	Image	character	False	False	140000		For each stored image a small thumbnail is generated (100kb) and stored as base64 encoding (text)
False	Description	text	False	False			Generic image description. Human readable information.
False	FileLocation	text	False	False		'/opt/imageStorage'::text	File system path were original image will be stored. Default value defined in Binary.importImg() list arguments
False	OriginalFileName	character	False	False	100		Original file name before UUID assignment
False	FileUUID	uuid	False	False			UUID used as filename.
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	100	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_Data	Public	BinaryDataId		
FK_Data_DataType	Public	DataTypeId		

Relationships

Columns	Association	Notes
(BinaryDataId = BinaryDataId)	**0..*** **LayerAttribute**.FK_LayerAttribute_Data **1** **Data**.PK_Data	
	0..* **ProfileAttribute**.FK_ProfileAttribue_Data **1** **Data**.PK_Data	
	0..* **DigitalMedia**.FK_DigitalMedia_Data **1** **Data**.PK_Data	
	0..* **Reference**.FK_Reference_Data **1** **Data**.PK_Data	
(DataTypeId = DataTypeId)	**0..1** **Data**.FK_Data_DataType **1** **DataType**.DataTypeId	DataType is binary. DataType table only has 1 type of data for binary

i18n

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* i18n
Detail: *Created on* 01-12-2012. *Last modified on* 01-12-2012.
Notes: *T*able relates between all the schemas/table in WOSIS and the RFC5646 language code of available translations

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	i18nid	integer	True	False			Unique identifier for each possible record in the table
False	schema	character	True	False	80		Schema name containing the translated record
False	table	character	False	False	80		Table name containing the translated record
False	pkeyid	integer	True	False			Primary key of translated record
False	rfc5646	character	False	False	20		RFC 5646 code e.g.: nl-NL, de-DE, pt-PT
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
i18nid_pkey	Public	i18nid		

Relationships

Columns	Association	Notes
(pkeyid = <any>)	0..* i18n.i18ind 1 <any>.<PK_any>	

Translation

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* i18n
Detail: *Created on* 01-12-2012. *Last modified on* 01-12-2012.
Notes: Content translation table. All translations are stored in this table.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	transid	integer	True	False			Primary key
False	i18nid	integer	True	False			Foreign key relating to i18n.i18n (translation metadata)
False	column	character	False	False	80		Column name of translation
False	value	text	False	False			Translation value
False	Created	timestamp	False	False		now()	Date of record creation.
False	Modified	timestamp	False	False		now()	Date of last record modification.
False	Editor	varchar	False	False	80	"current_user"()	Creator or last modifier of the record.
False	RevisionId	bigint	False	False			Reference to the last revision (i.e. update) of the record in the System.Revision table.
False	Version	integer	True	False		1	Sequential record version number - starting with 1 for the initial entry.

Constraints

Name	Type	Columns	Initial Code	Notes
transid_pkey	Public	transid		
i18nid_fkey	Public	i18nid		

Relationships

Columns	Association	Notes
(i18nid = i18nid)	**0..*** **translation**.i18nid_fkey **1** **i18nid**.i18nid_pkey	

System

Type: **Package**
Package: ISRIC Soil Data Repository
Detail: *Created on 22-2-2011. Last modified on 29-3-2011.*
Notes: Schema *System* contains database specific tables that support or enable a number of maintenance-related tasks:

- Keeping track of primary key values to be assigned to newly created records in the database - *Tally*
- Supplying an audit trail for changes to the database - *Revision*, *ChangeItem*, and *ChangeGeometry*

See Figure 29.

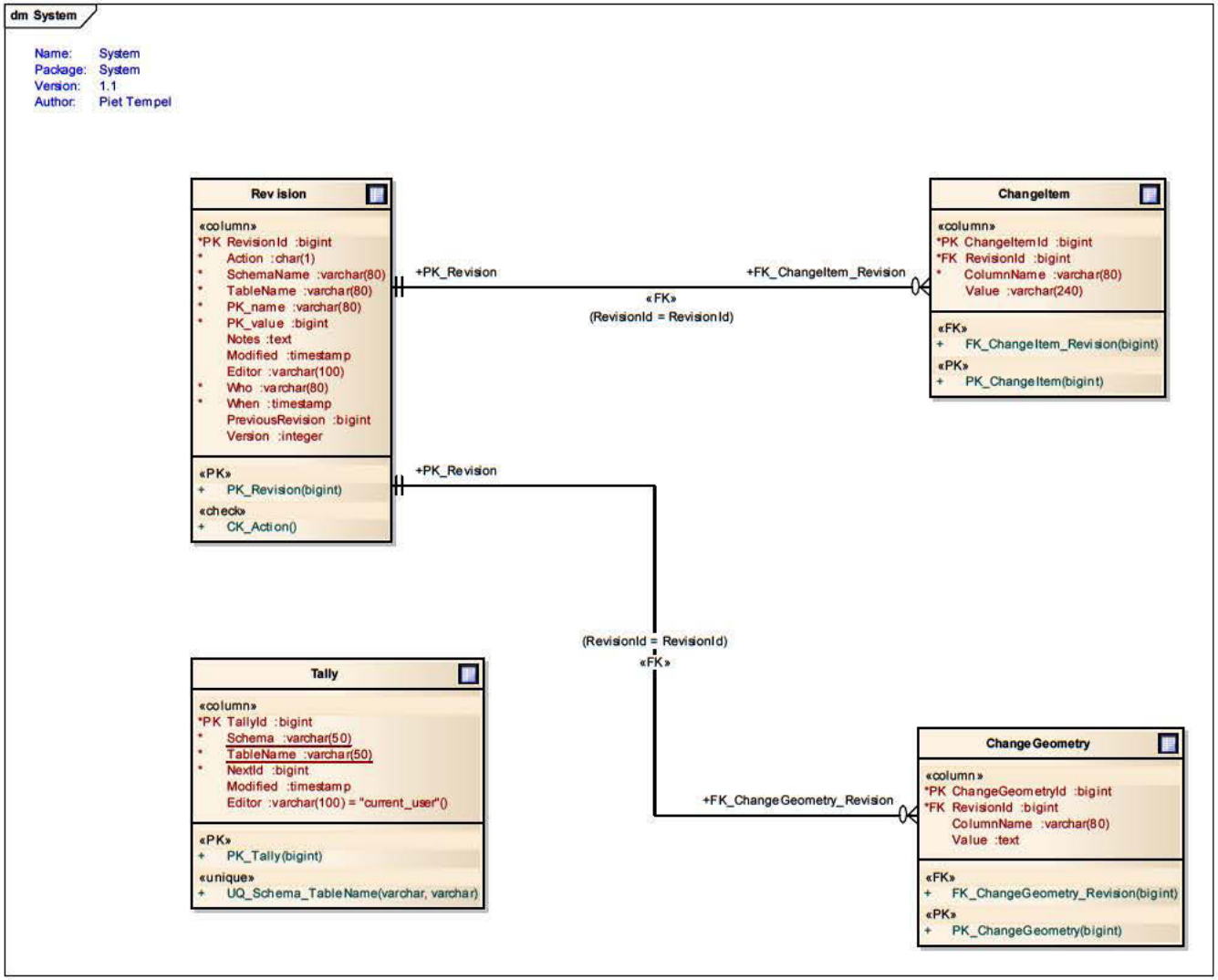

Figure 29
System schema.

ChangeGeometry

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* System
Detail: *Created on* 16-3-2011. *Last modified on* 28-3-2011.
Notes: Table *ChangeGeometry* stores deleted or updated data items of type 'geometry' from the database. All other deleted or updated data items are stored in table *ChangeItem*.
Column *RevisionId* refers to the revision in table *Revision* that the data items are part of.
Columns *ColumnName* and *Value* store the name of the affected column, and its original value (as text), respectively.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ChangeGeometryId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	RevisionId	bigint	True	False			Reference to a revision - either an "UPDATE" or a "DELETE" - in table *Revision*.
False	ColumnName	varchar	False	False	80		Name of the record's column whose value was changed. This column must be of type "geometry".
False	Value	text	False	False			Original value of the record's column whose value was changed.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_ChangeGeometry	Public	ChangeGeometryId		

Relationships

Columns	Association	Notes
(RevisionId = RevisionId)	**0..*** **ChangeGeometry**.ChangeGeometry_RevisionId_fkey **1** **Revision**.PK_Revision	

ChangeItem

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* System
Detail: *Created on* 16-3-2011. *Last modified on* 28-3-2011.
Notes: Table *ChangeItem* stores deleted or updated data items from the database. Deleted or updated geometry items are stored in a separate *ChangeGeometry* table.
Column *RevisionId* refers to the revision in table *Revision* that the data items are part of.
Columns *ColumnName* and *Value* store the name of the affected column, and its original value, respectively.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	ChangeItemId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	RevisionId	bigint	True	False			Reference to a revision - either an "UPDATE" or a "DELETE" - in table *Revision*.
False	ColumnName	varchar	True	False	80		Name of the record's column whose value was changed. Changes to Geometry columns are stored in a separate table.
False	Value	varchar	False	False	240		Original value of the record's column whose value was changed.

Constraints

Name	Type	Columns	Initial Code	Notes
FK_ChangeItem_Revision	Public	RevisionId		
PK_ChangeItem	Public	ChangeItemId		

Relationships

Columns	Association	Notes
(RevisionId = RevisionId)	**0..*** **ChangeItem**.FK_ChangeItem_Revision **1** **Revision**.PK_Revision	

Revision

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* System
Detail: *Created on* 16-3-2011. *Last modified on* 28-3-2011.
Notes: Table Revision stores an audit trail for changes to the database. The *Revision* table traces every UPDATE and DELETE operation to any record in the database (with the exception of *System* tables). The database operation is stored in column *Action*: 'U' for a record update, and 'D' for a deleted record.
Columns *SchemaName*, *TableName*, *PK_name* (name of the primary key column), and *PK_value* (primary key value) pinpoint the record affected by the operation.
Column *Who* registers the login role that executed the operation, *When* registers the timestamp of the operation.
Column PreviousRevision refers, by means of its *RevisionId*, to the previous revision of this record. *Version* contains the version number of the affected record.
The (original) values of the affected records are stored in tables *ChangeGeometry* (for geometry columns) and *ChangeItem* (all other columns).

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	RevisionId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Action	char	True	False	1		DML action; either 'U' for UPDATE, or 'D' for DELETE.
False	SchemaName	varchar	True	False	80		Schema for the table that was revised.
False	TableName	varchar	True	False	80		Name of the table that was revised.
False	PK_name	varchar	True	False	80		Name of the primary key field of the table that was revised. It is assumed that there is a single primary key field (as is the case with ORM surrogate keys) with position 1 in the fields list.
False	PK_value	bigint	True	False			Value of the primary key field of the table that was revised.
False	Modified	timestamp	False	False			Previous modification timestamp of the record that is revised.
False	Editor	varchar	False	False	80		Previous editor ("owner") of the record that is revised.
False	Who	varchar	True	False	80		Editor ("owner", in effect a login role) of the record that is revised.
False	When	timestamp	True	False			Timestamp for the revision of the record.
False	PreviousRevision	bigint	False	False			Points to a previous revision of the same record in the *Revision* table - NULL for a first revision, or a DELETE action.
False	Version	integer	False	False			Previous version of the record that is revised.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_Revision	Public	RevisionId		
CK_Action	Public		'Action' = 'U' OR 'Action' = 'D'	

Relationships

Columns	Association	Notes
(RevisionId = RevisionId)	**0..*** **ChangeItem**.FK_ChangeItem_Revision **1** **Revision**.PK_Revision	
(RevisionId = RevisionId)	**0..*** **ChangeGeometry**.FK_ChangeGeometry_Revision **1** **Revision**.PK_Revision	

Tally

Database: *PostgreSQL*, *Stereotype:* «table», *Package:* System
Detail: *Created on* 22-2-2011. *Last modified on* 28-3-2011.
Notes: Table *Tally* keeps track of the (numeric) primary key values to be assigned to newly created records. Upon insertion of a new record in a table, an INSERT trigger on that table:

- reads the primary key value for that particular table from the Tally table;
- assigns this value to the primary key field of the newly created record;
- increases the primary key value for that particular table in the Tally table with one.

Tally keeps track of the time and the last login role to insert a record in a particular database table.

Columns

PK	Name	Type	Not Null	Unique	Len	Init	Notes
True	TallyId	bigint	True	False			Unique identifier for each possible record in the table (ORM surrogate key).
False	Schema	varchar	True	True	50		Schema for the table.
False	TableName	varchar	True	True	50		Name of the table.
False	NextId	bigint	True	False			Next value for the ORM surrogate key of *schema.table*.
False	Modified	timestamp	False	False			Date of last record insertion in *schema.table*.
False	Editor	varchar	False	False	80	"current_user"()	Last login role to insert a record in *schema.table*.

Constraints

Name	Type	Columns	Initial Code	Notes
PK_Tally	Public	TallyId		

ISRIC – World Soil Information has a mandate to serve the international community as custodian of global soil information and to increase awareness and understanding of soils in major global issues.

More information: www.isric.org

WAGENINGEN UR
For quality of life

ISRIC – World Soil Information has a strategic association with Wageningen UR (University & Research centre)