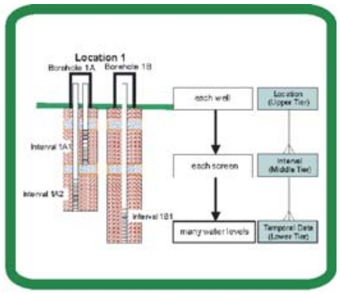
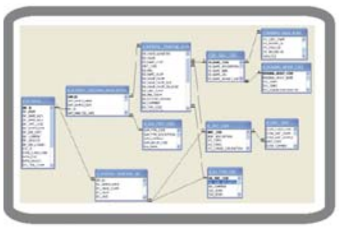
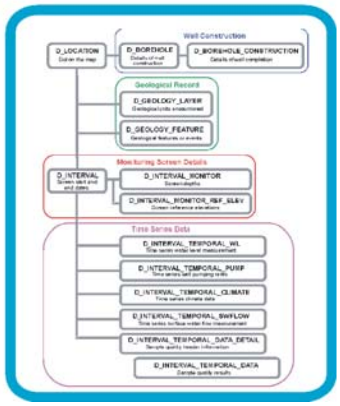






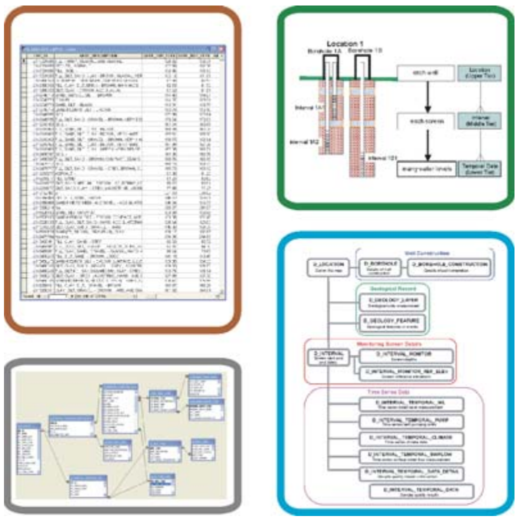
EarthFX Data Model

User's Guide

Version 1.1 Last Revised: November 2005

Last Reviewed: November 2005

Earth*fx* Incorporated



EarthFX Data Model

User's Guide

Technical Support:

EarthFX Inc.,
3363 Yonge St.
Toronto, ON
M4N 2M6

e-mail: support@earthfx.com

www.earthfx.com

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Databases

Introduction

Structure of Report

The objective of this report is to explain the rationale of the EarthFX data model in order to remove the intimidation barriers often preventing broader use of the database.

These goals are addressed through the following main discussion points:

- **Databases**: Databases were commercialized following data management challenges encountered during the Apollo Missions. Those challenges are not unlike the challenges facing us today, and the examples presented enforce the value and purpose of databases.
- **Data Models**: Poor data model design will permanently undermine the value and effectiveness of the database.
- **Using the Data Model**: This section presents the facts about using the data model, effectively where to find what, and a little background on syntax and naming conventions.
- **Data Model Applications**: This is a useful discussion of the application of the data model to real world objects (such as nested wells and pumping stations) and is intended to reinforce *why it is the way it is.*

Background on Databases

To generate the relevant information efficiently, the user needs quick access to the data (raw facts) from which the required information is produced. Data management systems, and their underlying databases, that address the data collection, management and retrieval are thus a core activity of any organization.

Good databases are carefully designed, developed, tested and optimized, with the objective of comfortably accommodating the projected *last piece of data to be added*, rather then simply a collection of tables designed to accommodate the data on hand. Often, the best test of an effective database is the amount of redundant data, which, if present, all too often leads to data conflicts that are difficult to trace.

Database Terminology

Like any computer-based system, a new language is developed to explain how the system works. The elements of a database are listed here

Field: Can be equated to a 'column heading' in spreadsheet terminology. It is typically includes no spaces and is follows and consistent naming strategy throughout the database. Most databases can use user friendly captions so that it is easier to understand by end users.

Record: A logically connected sequence of fields that collectively describe a person, place, object or event.

Table: A collection of related records.

Relationships: Establish how tables are related to each other.

Index: A constraint on a field used to promote data integrity (e.g. a well id field can only contain unique ID's)

Database: A collection of tables and their relationships.

Data Management System Elements

Hardware: The computer.

Software: Operating system, database software and database search engine.

Users: The people responsible for the database and the people who use the database in their job functions.

Procedures: The established sequence of steps required to perform operations on the data.

Data: The collection of facts stored in the database

Types of Databases

Hierarchical Databases

As the name implies, hierarchical databases are defined by data that is arranged hierarchically. Data arranged in this manner can be visualized as a tree structure, with a single table as the root, with other tables branching out. This type of relationship is how windows-based directory management systems (e.g., Windows Explorer) function. Links between tables in hierarchical databases make use of the Parent-Child relationship, with each child table having only a single parent table, while each parent can have multiple child tables. Parent and child tables are linked by "pointers". Although very rigid in the latitude given to users, especially with regards to changes to the database structure, hierarchical databases are computationally very efficient.

The main limitation of the hierarchical database structure is that is unable to accommodate many-to-many relationships, a common form of relationship in today's world (e.g., in a university setting, many students take many classes, such that there is a many to many relationship between students and the classes offered). In other words, these databases can only handle a single tree, and they cannot handle links between branches or over multiple layers. This can result in significant duplication of data within the database.

Relational Databases

In 1970, Dr. E.F. Codd, an IBM researcher, developed the relational database in an attempt to remedy several weaknesses of the hierarchal database. The relational database was hailed as the 'automatic' version of the historical 'standard' database. Unfortunately, the relational database was impractical because they required considerably more computational power than was available at the time. However, increased computational power and reduced costs have resulted in increased application of the relational database. Oracle. SQL Server, Access, Approach, Ingress and Informix are current examples of relational databases.

The power of relational databases lies in the relational database management system (RDBMS). This is the software, built into the database program that provides the user with a comfortable environment within which the data is controlled. The software interprets the user's commands to enact data commands. This allows the user to 'see' the database as a series of simple data tables, and manage the data using a simple 'quasi English' programming language called SQL (structured query language). **Using SQL, users can ignore the physical storage structure of the data, and create tables to mimic our logical data storage vision.** SQL allows the users to tell the database what to do, without worrying about how to do it.

Relational databases suffer from two simple disadvantages. First, they require considerably more computational power to operate (a diminishing concern) and, second, they offer so much latitude to the user that novice users can inadvertently create instability in the database or generate inaccurate SQL queries.

Chapter 2

Data Models

Background – Data Models

Data models and data modelling are the first step in the database design process, serving as a bridge between the functionality of the database software and the real-world objects that we see and work with. Traditionally one of the challenges of data modelling is that database programmers and database users see data in different terms. Data modelling is therefore an iterative process that involves both parties in order to maximize the compatibility between the database and the user preferences.

The goal or purpose of the data model is to visually display the meaning of the data as well as the relationships that exist between the different elements of the data. The data model is the design of the database, and it includes the definition of tables and their fields (what information goes where) and how these tables are linked. Embedded within the table and field descriptions are the definitions of field types (text, date, number etc.), field size (number of characters), relationship types (one-to-one, one-to-many etc.), and indexes.

> *Data modeling, as it applies to database systems, is the process of defining the grammar, vocabulary, and content to be used to represent information in a database system. The grammar defines the relationships between individual elements, or objects, in the system; the vocabulary defines the terminology to be used to describe, or attribute, individual elements; and the content defines what is to be included in the system and what is not. Note that the modeling process is independent of specific hardware and software which only become important at a subsequent implementation stage.*
>
> *To be useful, the data modeling process must be carried out in an open, inclusive manner, so that the community of eventual users has adequate input into the design process. The whole reason for the data modeling process is to build systems that make our information more* useable and effective.
>
> Bruce Johnson, USGS

In addition, data models should be scalable, which will facilitate migration to other, enterprise-scale software platforms.

A data model can be evaluated by several criteria:

- **Completeness:** The data model needs to support all of the necessary data.
- **Non-Redundancy:** This ensures that data is only recorded in one location within the data model. Data redundancy increases storage costs and can create data inconsistencies.

- **Enforcement of Business Rules:** The data model needs to reflect the rules governing how data is collected (e.g., a chemical analysis belongs to only one groundwater sample).
- **Data Reusability:** The data model needs to allow data to be used for purposes beyond those that the data was initially collected for.
- **Stability and Flexibility:** The data model has to be generic and flexible enough to cope with change. A model is *stable* if it does not need to be modified in the event of a change in requirements. A model is *flexible* if it can be readily extended to accommodate new requirements.
- **Simplicity and Elegance:** The data model should provide a reasonably natural classification of data. Elegant models are inherently simple, consistent and easily described and summarized.
- **Communication Effectiveness:** The data model has to be an effective communication tool for data management.

Geological Data Models

Storing and managing geological data, from structural features on a continental scale, to water-level and chemistry measurements, is a challenge due to the inherent descriptive nature of our geological world – much of the data is interpretive and subject to debate, and most data varies in time and space on scales of many orders of magnitude. In addition to this, maps are used for both cartography and analysis, which in the digital world, places different demands on the data model.

A fundamental aspect of geological data modelling is whether the objects being modeled are singular objects or compound objects. Singular geologic objects are directly observed at a single location, and include soil sample descriptions, water levels, chemical analyses etc. In many cases, singular geologic objects are also single map entities (a well, for example), but may also belong to a more general, compound object (the top of bedrock contact in a borehole is a singular object, but belongs to the broader bedrock-surface object, which is a compound object). Compound geologic objects typically arise from the interpretive grouping, or classification of many singular observations from many locations. Examples include map units made up of many observations of outcrops, top of bedrock surfaces, stream networks and lake shorelines, for example. Compound objects effectively become map objects, and introduce legends and symbol libraries.

Initially, databases were used to store the singular objects (wells and geological contacts) and a Geographical Information System (GIS) was used to store compound spatial objects (e.g. rivers, lakes, geological features). This led to what we refer to as: *First Generation* Geological Data Models. The evolution of GIS towards more holistic data management systems is creating a *Second Generation* Geological Data Models, as GIS developers and others seek to store all data in a single relational database that is linked to the mapping interface. The ESRI ArcHYDRO data model is a good example of a second-generation data model, because it accommodates both time-series data (stream discharge) and spatial objects (watershed boundaries).

Several second-generation data models are in use today, including the North American Data Model (NADM) (http://cgkn.net/2002/projects/bedrock_e.html) developed by the Geological Survey of Canada (GSC) and the United States Geological Survey (USGS) for map object data management, and ArcHydro. (www.esri.com), developed by ESRI with support from the USGS and the British Geological Survey. Such models try to deal with issues of empirical versus interpreted data (i.e. was it actually noted in the field, or is it interpolated from other points?). This issue resulted in coding systems such as Defined, Approximate and Assumed to qualify many of the database entries. In addition to uncertainly about the objects, the data models must also cope with processes and uncertainly in their interpretation. Geological map symbols must be built into the library systems to portray interpreted processes on the maps.

The design objectives of the EarthFX data model are limited to singular data, and fall within the bounds of first-generation data models. Minor object-grouping functionality is built into the data model to address municipal pumping-house scenarios (one or more production wells, one or more monitoring wells, chlorinators, high lift pumps etc).

The Evolution of the EarthFX Data Model

The evolution of the EarthFX data model has been driven by the need of hydrogeologists to store large amounts of groundwater-related data and the need to rapidly retrieve and analyze these data. Through several iterations, we have made an effort to balance database theory (normalization) and user preferences (spreadsheets) to foster usability yet accommodate larger volumes of data while using JET/Access as the database engine and interface.

The first version of the data model, developed for a landfill in 1991, was biased towards simple groundwater monitoring applications, and was built on two key tables: BOREHOLE, which included details of the position, name, depth and date of the well, and SCREEN, which was added as an afterthought to accommodate boreholes with multiple screens. Time-series data was separated into various other tables, and the data model did not include reference or look-up tables.

This was undoubtedly a simple database, with eight tables, all of which looked like spreadsheets. Look-Up (or reference) tables were not used. This meant that all of the information in the tables was stored as the text strings, rather than codes. On the upside, novice users were quick to learn how to use the database; whereas on the downside, it was not scalable, and simply would not function under the load of the current applications, such as watershed or regional databases.

The data model underwent significant changes with its application to the Smithville PCB Site in 1997. The Smithville site is a small property of several acres located south of Grimsby, Ontario, just outside the village of Smithville. Used transformers were stored and partially recycled on the property. Consequently, PCB waste oil entered the fractured rock groundwater regime. The Ontario Ministry of the Environment took over the site and managed several years of investigative work to further define groundwater contamination characteristics. This lead to massive amount of time-series data, extracted from a variety of inclined monitoring wells completed with WestBay equipment.

This project led to scalability within the data model. Nested wells were no longer vertical, and were tested with packers or WestBay equipment. This required greater flexibility in the database fields in order to store the physical configuration of the wells. Frequent water-levels measurements, pumping tests and water-quality sampling led to a re-evaluation of the time-series data tables, and the creation of the D_INTERVAL series of tables in the current EarthFX data model. Look-Up tables were introduced and used to encode repetitive information such as water-quality parameter names. These look-ups enforced strict QA/QC on the families of volatile organic compound parameters (often with multiple, yet similar names) being tested.

In 1999, EarthFX was retained by a petroleum-engineering firm to support their data management and analysis needs for oil-well exploration data. This led to an important maturing of the data model as EarthFX gained experience with the global-scale petroleum data models of POSC (Petrotechnical Open Software Corporation) and PPDM (Public Petroleum Data Model). These data models are immense, and include fields and tables for all tabular data collected from all aspects of the petroleum business cycle. This exposure provided EarthFX with a ‘final vision’ of a comprehensive groundwater data model. This vision provides guidance in the development of the data model process and reinforced the notion that for large applications, data models are complex.

The final step in the evolution of the EarthFX data model was its application at Ontario Power Generation’s (OPG) Pickering Nuclear Generation Station to manage their environmental monitoring program. OPG was collecting monitoring data from wells, surface water stations, sumps, sewers, trenches, manholes, weather stations and rabbit holes and thus required a data management system to consolidate the data that was collected. This expanded the EarthFX data model from being largely 'well based', to being 'hydraulic based', a process achieved by the incorporation of more generic name conventions (such as *Location* and *Interval* rather then *Borehole* and *Screen*). In addition, this led to the removal of well-specific fields, which were replaced by generic fields applicable to groundwater, surface water and meteorological data sets.

Other Relevant Geological Data Modelling Projects

Ministry of Northern Development and Mines

In 2002, the Ontario Ministry of Northern Development and Mines (MNDM) undertook a data modelling exercise to develop a database for future provincial 3D geological and aquifer mapping projects. The objective was to develop a data model that captured the necessary borehole data (effectively the current EarthFX data model), plus the compound objects that are derived from the borehole data (hydrostratigraphic surfaces, 3D objects etc.). The outcome of this process served as verification for the EarthFX data model…the assembled group built a conceptual data model from first principles and they arrived at a model that was generally consistent with the current EarthFX data model.

Australian Groundwater Data Model

The Bureau of Rural Sciences in the Australian Ministry of Land and Water has developed a singular-object data model for groundwater applications, called The Australian National Groundwater Data Transfer Standard. The data model addresses a wide range of groundwater features, from wells to springs to tunnels and mines, and accommodates positional and construction information, and temporal data. The data model is based on a 'one Location to many Interval' relationship, and employs well construction and well sample structure that is virtually identical to the EarthFX data model.

The model exceeds the EarthFX model in the management of position information, such as coordinates, map information, driving instructions, local land use and others. The Entity Relationship for the data model is shown below. The *groundwater_feature* table is equivalent to *location*, and *groundwater_source* is equivalent to *interval* (note that this is strictly a groundwater data model, and not a broader hydraulic data model and therefore does not accommodate surface water or meteorological data).

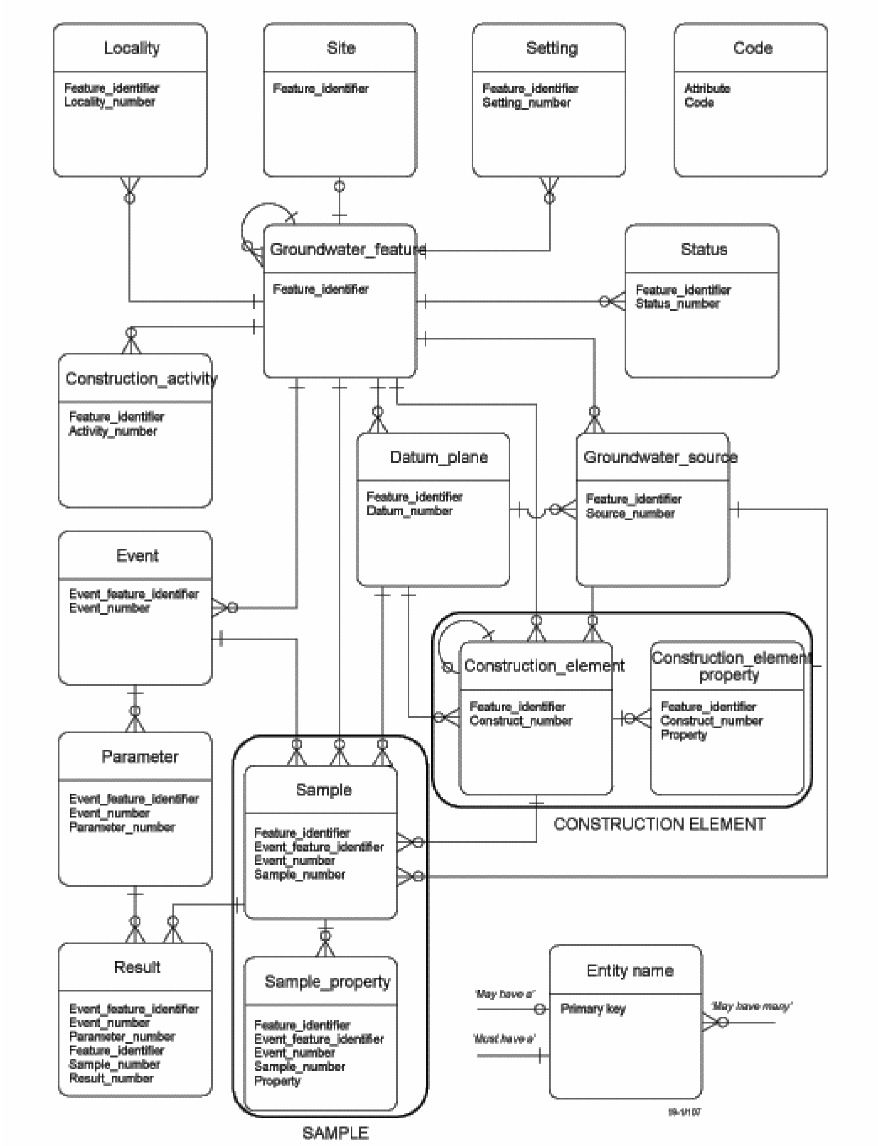


Figure 1   Australian Groundwater Data Model

Chapter 3

The EarthFX Data Model

Introduction

The EarthFX data model has been designed with several criteria in mind, including the ability to accommodate current and foreseeable data. Moreover, it has been designed such that data can be intuitively queried in a hydrogeological context. Criteria include:

- Construction of tables, queries, forms, and macros to effectively store and retrieve the disparate data from the database
- Incorporation of data from the data model into various modelling software packages
- Flexibility in terms of incorporating new data types that might be collected at some future point.

Based on the data needs, a three-tier data model was adapted to include the fields for the required data. In the three-tier data model, the upper tier carries information about the well (or climate station), the lower tier carries all temporal data, and the middle tier accommodates multiple data sources (such as multiple monitoring screens in a well, or multiple radiation meters at a weather station) for temporal data at each well.

This structure is critical to a hydraulic database because it allows wells to have multiple screens, and screens to have multiple temporal water-level readings. The structure also easily accommodates meteorological data, surface water flow data and others, because the locations can be any ‘dot on the map’ from which data can be collected. By classifying locations by their description (e.g., Well, Surface or Water Station) and classifying intervals by the type of data they produce (water levels, water quality etc.), we can easily isolate specific data types from specific locations.

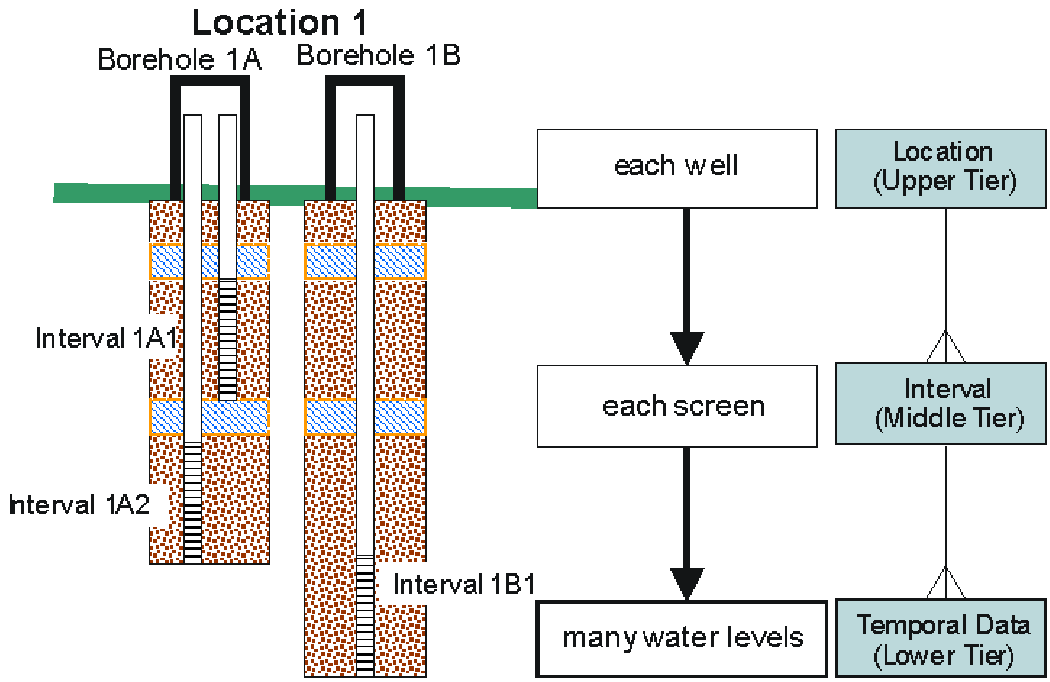


Figure 2 Data Model Objects

Normalization

Data models under go a process or normalization with the goal of minimizing the number of blank fields in the final database (blank fields cause in-efficiencies and are the source of possible uncertainly in the database). Normalization can be a complex science, as it involves creating more tables, each with less and less data, thereby removing the opportunity for blank fields and incorporating greater data integrity into the database. This, however, can burden the database with slow response times, as lengthy queries are necessary even for the simplest requests. A compromise is necessary relating data integrity, database response times and usability of the database.

Database Fields

Within each table, four types of fields are usually present.

Linking Fields

Linking fields are the internal ID numbers assigned to each major entity in the database. Each location, borehole, interval has a unique ID number, with the field called LOC_ID, BH_ID and INT_ID. The value of these fields should not be changed, as it will destabilize the database. The following figure shows how the main tables (D_LOCATION, D_BOREHOLE and D_INTERVAL) are related.

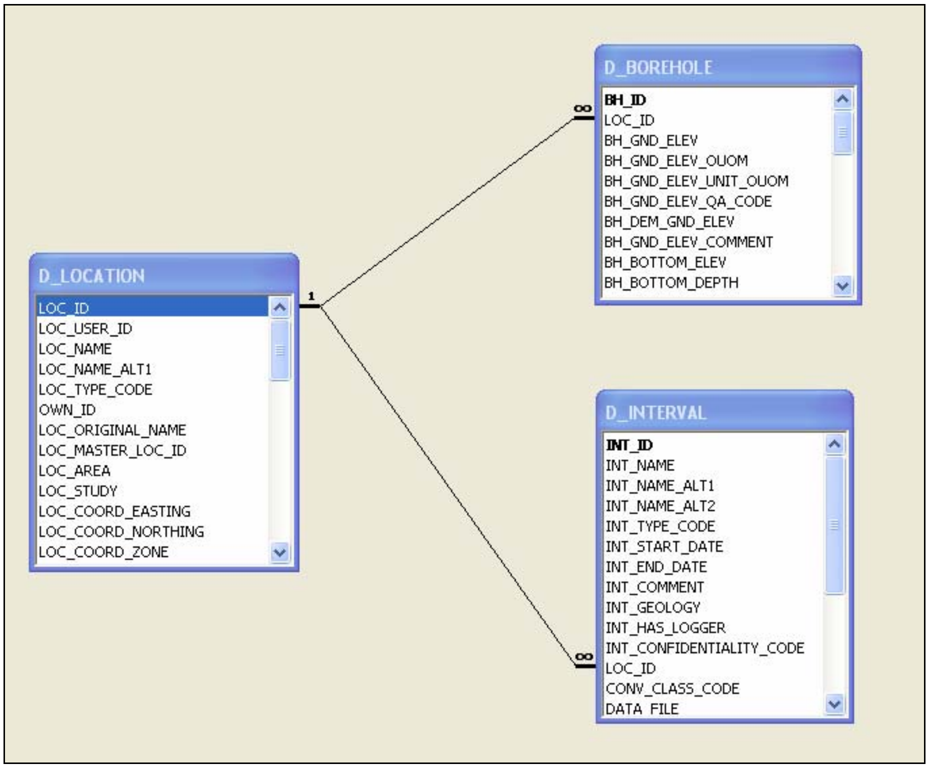


Figure 3 Relationships between D_LOCATION, D_BOREHOLE, D_INTERVAL

Data Fields

These fields are the primary purpose of the table and contain the information stored by the database and required by the queries.

Original Data Fields

The data model is based on the principle that the units of raw data may not be consistent with the 'system/reporting/analysis' units (e.g., water levels may be measured in feet below top of pipe, yet our analysis will require elevations as meters above sea level). This is accommodated with the use of Original Unit of Measure (OUOM) fields. Most tables therefore have several OUOM fields for storing the original data that is loaded into the database, and a parallel series of fields (often called interpreted or reporting fields), usually with similar names, without the OUOM suffix, for storing the corrected data. Queries within the database can be run periodically to update the system fields from the OUOM fields. If data is to be changed, it should therefore be changed in the OUOM fields. Changes made only to the interpreted fields will be overwritten with the next pass of the update queries. The following figure shows the D_BOREHOLE table; OUOM fields are highlighted:

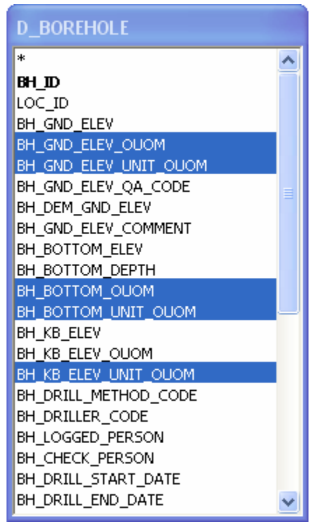


Figure 4   OUOM Fields in the D_BOREHOLE Table.

The importance of the OUOM cannot be understated as they often contain the most accurate data. For example, in the D_GEOLOGY_LAYER table (which contains the geological sequence of layers reported by the driller), the OUOM fields contain the depths to each unit measured in the field by the driller. The interpreted fields contain the corresponding elevations, in masl (or fasl if desired), based on the ground elevation of the well.

System Fields

System fields are found at the right side of most tables and are used by data management software (such as SiteFX and the associated synchronization / partitioning tools) to manage the database. System fields are named with a SYS prefix. Several of the system fields are useful to the user, specifically the SYS_CHANGE_EXPLANATION (used for tracking user correction to the data), SYS_TIMESTAMP (for recording the date and time the record was added, and in some cases updated). The remaining system fields should never be used to store permanent data as they are often overwritten during internal data management tasks.

Naming Conventions

A short list of rules was adopted (mostly from the common Petroleum Data Models) for naming tables and fields.

Tables:

All tables in the data model are prefixed by a single letter as follows:

D: a Data table, populated with data specific to the client or project

L: a Library table, populated with data useful to many projects, such as environmental criteria

R: a Reference (Look-UP) table, populated with data that is used to code information in the data tables

S: a System table (used either by SiteFX or during data validation or synchronization)

U: User tables (currently not in use)

Access also manages a set of tables prefixed with the letters **MSys**. These tables are often hidden from view (depending on the options set in Access).

Table names are (usually) built on the name of the primary table name. For example, D_INTERVAL is the primary table holding details of each interval. The table D_INTERVAL_MONITOR therefore holds information about intervals that are groundwater monitors. This method also benefits from the fact that tables names are often displayed alphabetically.

Fields

Field names were assigned to be as meaningful as possible to the user, to contain no spaces or punctuation (which makes writing queries easier), and to be unique. Usually fields are uppercase but it is not required. Generally the field name is prefixed by several letters that identify the table name, mainly to prevent cases were there may be some confusion between similar field names in different tables (e.g. a 'name' field). One exception is the D_INTERVAL_TEMPORAL tables, where we have made an effort to keep field names consistent (e.g., the reading date field – RD_DATE – is common in all tables. This has been done so that the temporal tables can be 'stacked' (using union queries)).

The Data Model

The primary tables in the three-tiered data model are illustrated in the following figure with their relationships. A more detailed description of these tables is provided below.

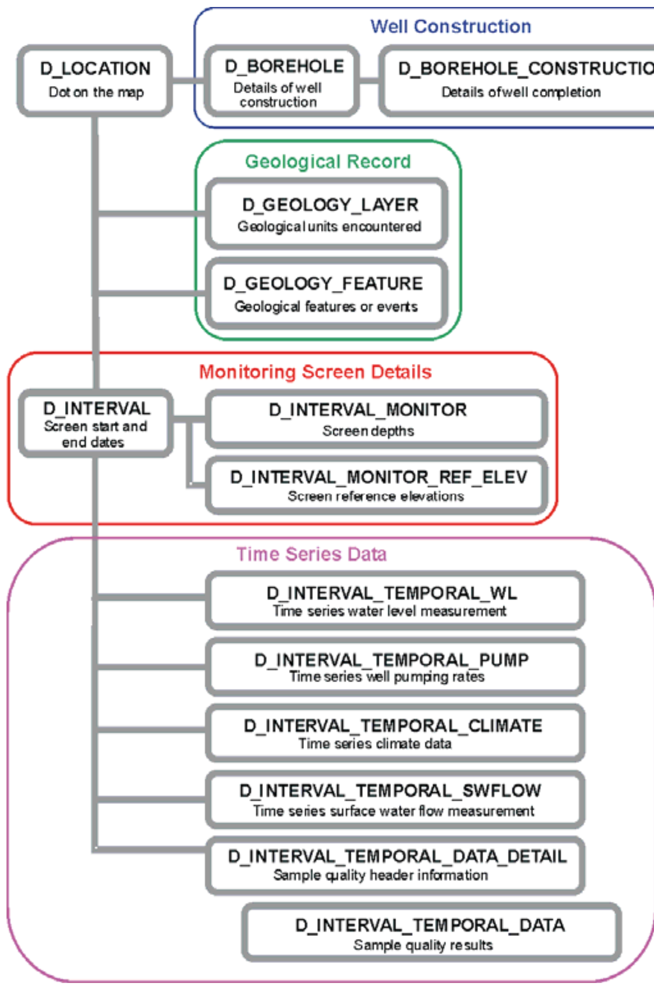


Figure 5 Data Model Skeleton

This next figure shows schematic of the different elements of a typical hydrogeological setting, and how these different elements are stored in the data model.

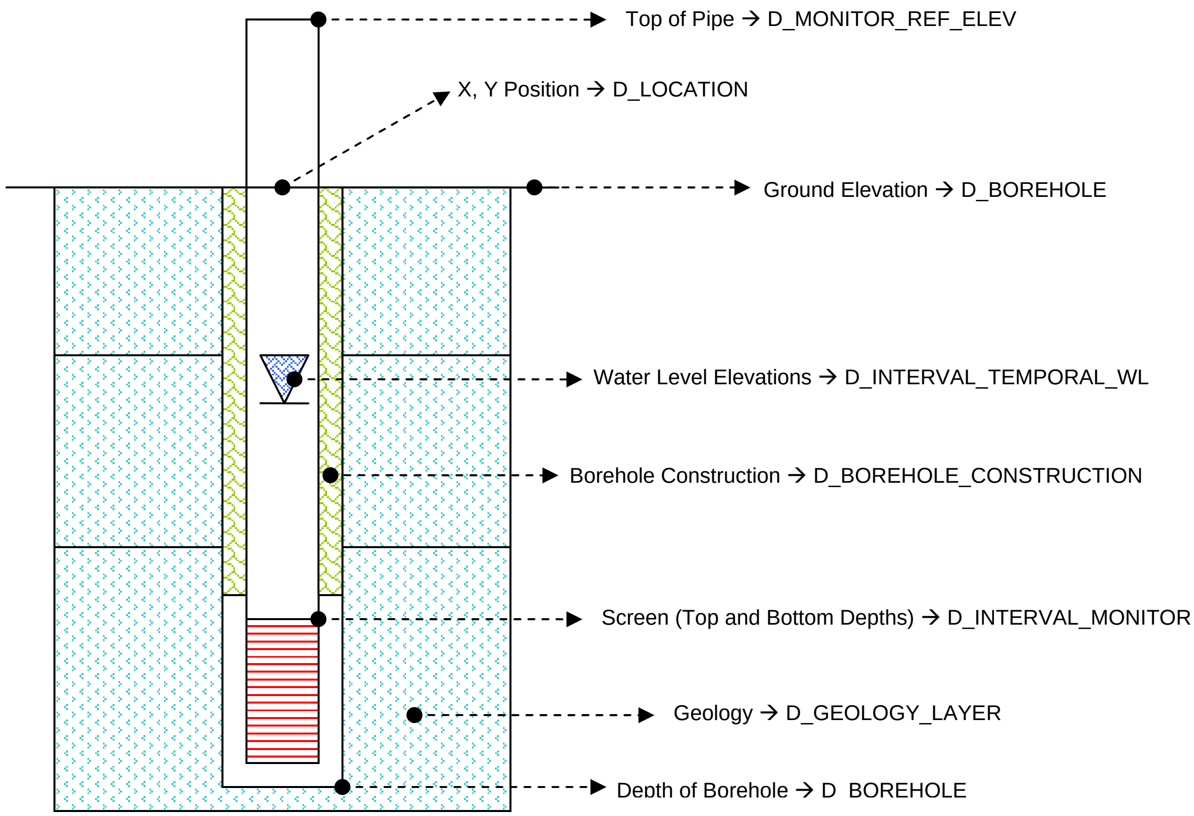


Figure 6 Schematic of a typical Borehole Construction and Associated Database Tables.

D_LOCATION

The D_LOCATION table should be considered the central table and contains positional information about that location.

A *location* is (or has been in the past) capable of producing data about hydrogeological, climate or hydrological conditions (i.e., monitoring points within the hydraulic cycle). Groundwater locations typically include monitoring wells, production wells, test holes, test pits and trenches, but could include tunnels, tile drains, mine shafts, and other structures that intercept the water table. Surface-water locations are usually limited to either a gauging station on a stream or river for measuring discharge, or a gauging station on a lake for measuring elevation. Climate stations measure atmospheric conditions, common precipitation and temperature, as well as solar radiation, wind speed and direction. Other features of locations are listed here:

- Locations can be grouped into families to reflect nested monitoring wells or an amalgamation of locations (such as a pump house) that are thought of as a dot on the map.
- Key properties of a location are:
  - name
  - position coordinates
  - type (e.g., monitoring well versus surface water).
- Locations are never deleted from the database, unless they are duplicated. Should a location cease to be in operation, its status is changed to inactive.

- The LOC_COORD_QA_CODE field is used to provide the user of the data a means of assessing the expected reliability or accuracy of the locations position. The code is assigned a value of 1 through 9, which then links to the R_LOC_COORD_QA_CODE table. One example of data from R_LOC_COORD_QA_CODE is show in the table below. Other QA schemes have been used for specific projects.

LOC_COORD_QA_CODE	**LOC_COORD_QA_DESCRIPTION**
1	Margin of error: <= 3m
2	Margin of error: 3 – 10 m
3	Margin of error: 10 – 30 m
4	Margin of error: 30 – 100 m
5	Margin of error: 100 – 300 m
6	Margin of error: 300 m – 1 km
7	Margin of error: 1 - 3 km
8	Margin of error: 3 – 10 km
9	Margin of error: unknown error

Table 1 The R_LOC_COORD_QA_CODE table.

Table 2 outlines the primary attributes of the location entity. Each location is defined by a unique LOC_ID number that is assigned and managed by the database engine. Note that in applications in which several data models are synchronized, this number may change and therefore it should not be used in external applications without proper precautions.

Name	**Data Type**	**Description**
LOC_ID	Number	Unique identifier for location
LOC_NAME	Text	Name of location
LOC_TYPE_CODE	Code	Type of location (e.g., Monitoring well). Links to R_LOC_TYPE_CODE
LOC_COORD_EASTING	Number	UTM NAD83 position of location
LOC_COORD_NORTHING	Number	UTM NAD83 position of location
LOC_COORD_QA_CODE	Number	Coordinate accuracy code. Links to R_LOC_COORD_QA_CODE
LOC_LOT	Text	Lot
LOC_CON	Text	Concession
LOC_COUNTY_CODE	Number	County in which the location is located. Links to R_LOC_COUNTY_CODE
LOC_TOWNSHIP_CODE	Number	Township or municipality in which the location is located. Links to R_LOC_TOWNSHIP_CODE
LOC_COMMENT	character	Comments about location

Table 2 Key Fields of the D_LOCATION table.

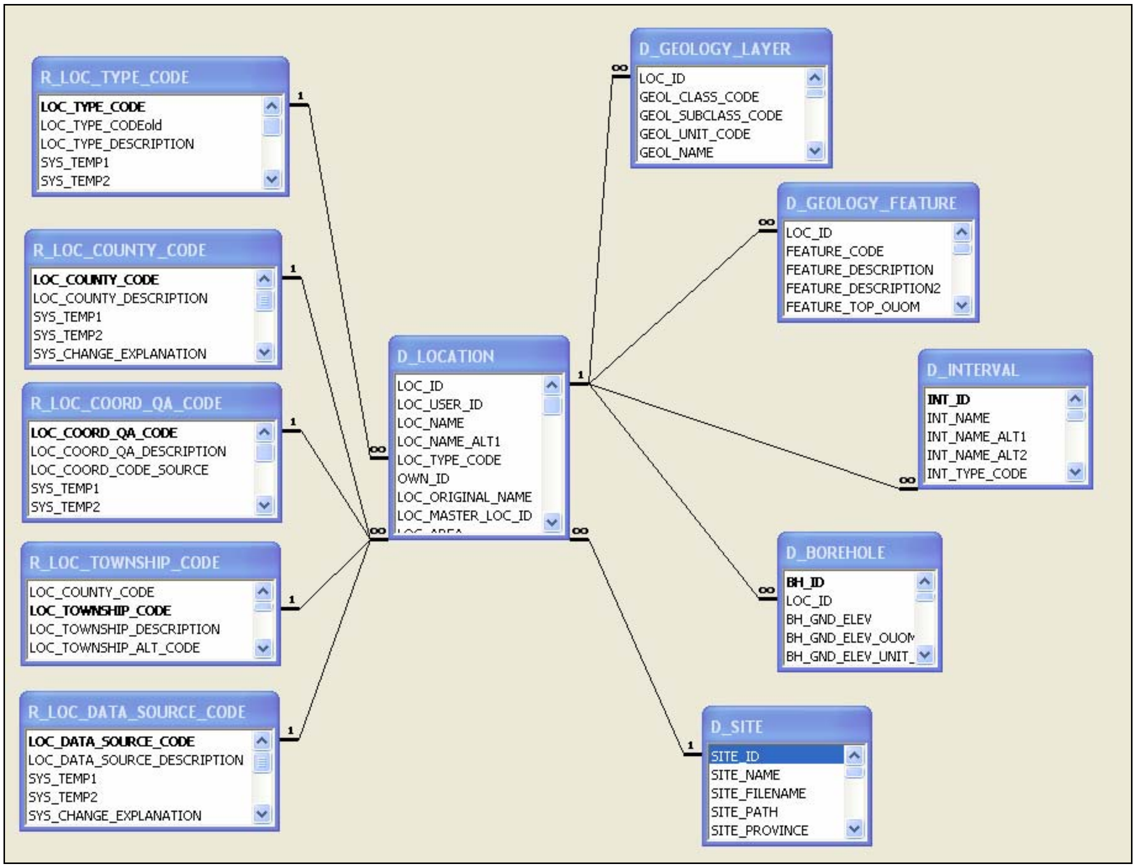


Figure 7   Main Relationships of the D_LOCATION Table.

D_BOREHOLE

D_BOREHOLE is used to store details of the well installation, including elevation, date, depth, driller.

If the *location* is a borehole, details are stored in the D_BOREHOLE table, with a one-to-one relationship between D_LOCATION and D_BOREHOLE. Borehole construction information, such as casing depths, seal etc. are stored in D_BOREHOLE_CONSTRUCTION because a single borehole can have many construction segments. Screen information is stored in a separate table (D_INTERVAL_MONITOR) linked to the D_INTERVAL table because screens can be time-dependent (e.g., a packer interval has only a short life before it is moved) whereas borehole construction details are permanent.

The following table lists the common fields from the D_BOREHOLE table, provides a brief description and identifies their corresponding Look-Up table. Note that by opening the table in design view, a complete description of each field is provided.

Name	Data Type	Look-Up Table	Description
BH_ID	number	N/A	Unique identifier for each borehole
BH_GND_ELEV	number	N/A	Ground elevation at borehole (can be typed in, or calculated via SiteFX). Units are system units (usually mASL)
BH_BOTTOM_ELEV	number	N/A	Elevation of the bottom of the borehole (in system units)
BH_BOTTOM_DEPTH	number	N/A	Total depth of the borehole, in system units (mBGS)
BH_DRILLER_CODE	number	R_BH_DRILLER_CODE	Drilling company code.
BH_DRILL_METHOD_CODE	number	R_BH_DRILL_METHOD_CODE	Drill method code
BH_DRILL_START_DATE	date	N/A	Date drilling activities started
BH_DRILL_END_DATE	date	N/A	Date drilling activities completed
BH_STATUS_CODE	number	R_BH_STATUS_CODE	Borehole status code (e.g., water supply well, observation well, etc.)
BH_COMMENT	test	N/A	Comments
LOC_ID	number	N/A	Links to the D_LOCATION table

Table 3 Key Fields of the D_BOREHOLE Table

The main relationships for the D_BOREHOLE table are shown in the figure below:

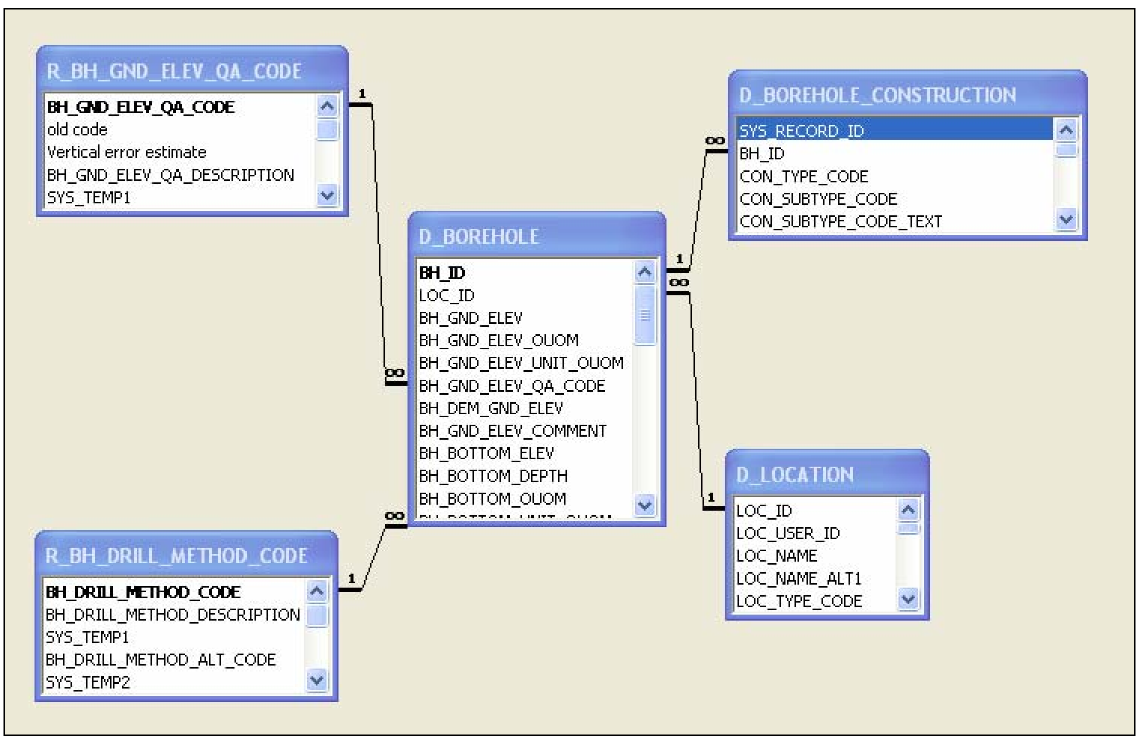


Figure 8 Main Relationships of the D_BOREHOLE Table.

D_BOREHOLE_CONSTRUCTION

D_BOREHOLE_CONSTRUCTION is used to store details of the well completion, including casing diameters and depths, seal position and material and sand packs..

If the *location* is a well or borehole, information about the installation is stored in the D_BOREHOLE table, and information about the construction (sometimes referred to as the completion) of the well is stored in the D_BOREHOLE_CONSTRUCTION table. This table is linked to D_BOREHOLE using the BH_ID field on a one-to-many relationship, which means that there can be many construction records for each borehole record.

Name	Data Type	Look-Up Table	Description
BH_ID	Number	N/A	Links to D_BOREHOLE table
CON_TYPE_CODE	Number	R_CON_TYPE_CODE	Required field. Indicates type of construction
CON_SUBTYPE_CODE	Number	R_CON_SUBTYPE_CODE	Construction material code
CON_TOP_ELEV	Number	N/A	Elevation of the top of the well construction element (in system units)
CON_BOT_ELEV	Number	N/A	Elevation of the bottom of the well construction element (in system units)
CON_COMMENT	Text	N/A	Comment
CON_DIAMETER	Number	N/A	Diameter of the construction material

Table 4 Key Fields in the D_BOREHOLE_CONSTRUCTION Table

Note: *CON_DIAMETER* field is unitless, meaning that the units provided by the user are not converted to system units like elevation.

The main relationships for the D_BOREHOLE_CONSTRUCTION table are shown in the figure below:

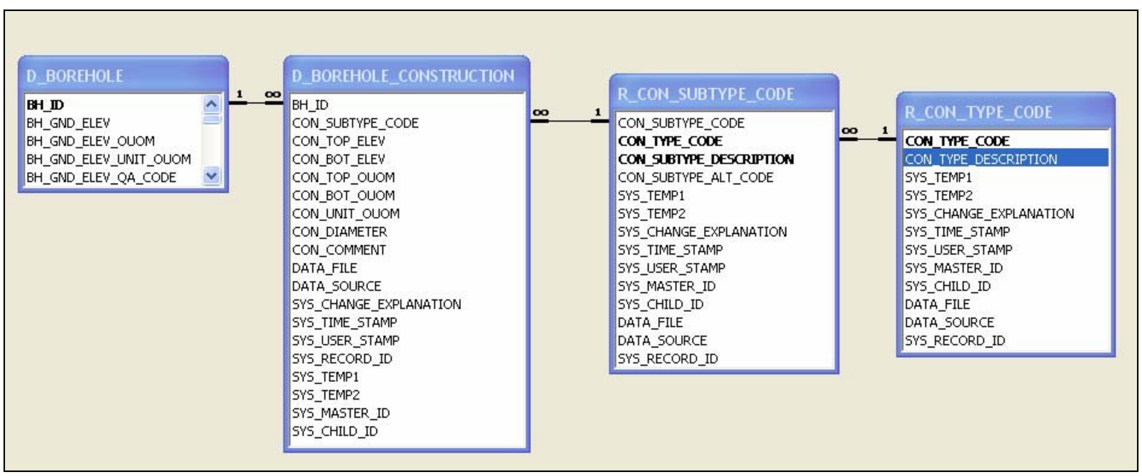


Figure 9 Main Relationships of the D_BOREHOLE_CONSTRUCTION Table.

D_INTERVAL

D_INTERVAL records the monitoring points from which time series data is collected.

The concept of the *interval* in the data model evolved from an early version of the data model that tracked only boreholes and monitors. The interval can be though of as a monitor, or more broadly, the source of all temporal data. All temporal data collected must be tied to an interval, whether the data is a water level or a surface water flow measurement from a weir (in which case, the *location* and the *interval* are effectively the same thing). However, in order to maintain the relationship between temporal data and the *location*, the data model requires that each location have one or more intervals, and that any temporal data be linked to the interval.

The primary interval table is called D_INTERVAL, and lists:

- Interval name: Often the same as the location name, except in cases where there are two or more intervals at the same location.
- Interval type: Often the same as the location type (e.g., location type may be a climate station, and interval type could also be a climate station, but in the future could be expanded to rain gauge, thermometer etc). Alternatively, the interval type can be used to described the "type" of data monitored (e.g. ground water, surface water, soil)
- Interval dates: Because intervals are temporal (packer intervals are the best example), it is necessary to track the start and end date of each interval).

Several tables are linked to the D_INTERVAL table:

- D_INTERVAL_MONITOR: Provides details of the screened interval if the interval is a monitoring screen. Included are top and bottom elevations, screen material, slot size etc. Based on a one-to-one relationship with table D_INTERVAL.
- D_INTERVAL_REF_ELEV: This table lists the reference elevation (often top of pipe (TOP) elevation) for monitoring wells and is used to correct water-level measurements. Each TOP elevation is bracketed by a start date and an end date which allows the monitor to be periodically resurveyed.

A well can have multiple screens, and a weather station can have multiple sensors. The D_INTERVAL table is linked to the D_LOCATION table through the LOC_ID field. D_INTERVAL stores the name of each screen or sensor, and links to the table that stores the actual time-series data.

Like the D_LOCATION table, the D_INTERVAL table allows for three separate names for each interval and a code for the interval type. If packer tests were completed in a cored, open-hole test well, the *LOC_TYPE_CODE* in the D_LOCATION table would be set to 'test well', and the *INT_TYPE_CODE* field in the D_INTERVAL table would be set to 'packer'.

Name	Data Type	Look-Up Table	Description
INT_ID	number	N/A	Unique system number for each interval
LOC_ID	number	N/A	Links to the D_LOCATION table
INT_NAME	text	N/A	Primary name of interval, often the same as the location name
INT_NAME_ALT1	text	N/A	Secondary or alias name of interval
INT_NAME_ALT2	text	N/A	Tertiary name of interval
INT_TYPE_CODE	number	R_INT_TYPE_CODE	Code to indicate type of interval
INT_START_DATE	date	N/A	Start date of interval
INT_END_DATE	date	N/A	End date of interval
INT_COMMENT	text	N/A	Comment

INT_GEOLOGY	text	N/A	Optional. Information about the stratigraphic zone where the interval is installed.
INT_HAS_LOGGER	Yes/No	N/A	Used to indicate if interval is equipped with a data logger

Table 5 Key Fields in the D_INTERVAL Table.

The main relationships for the D_INTERVAL table are shown in the figure below:

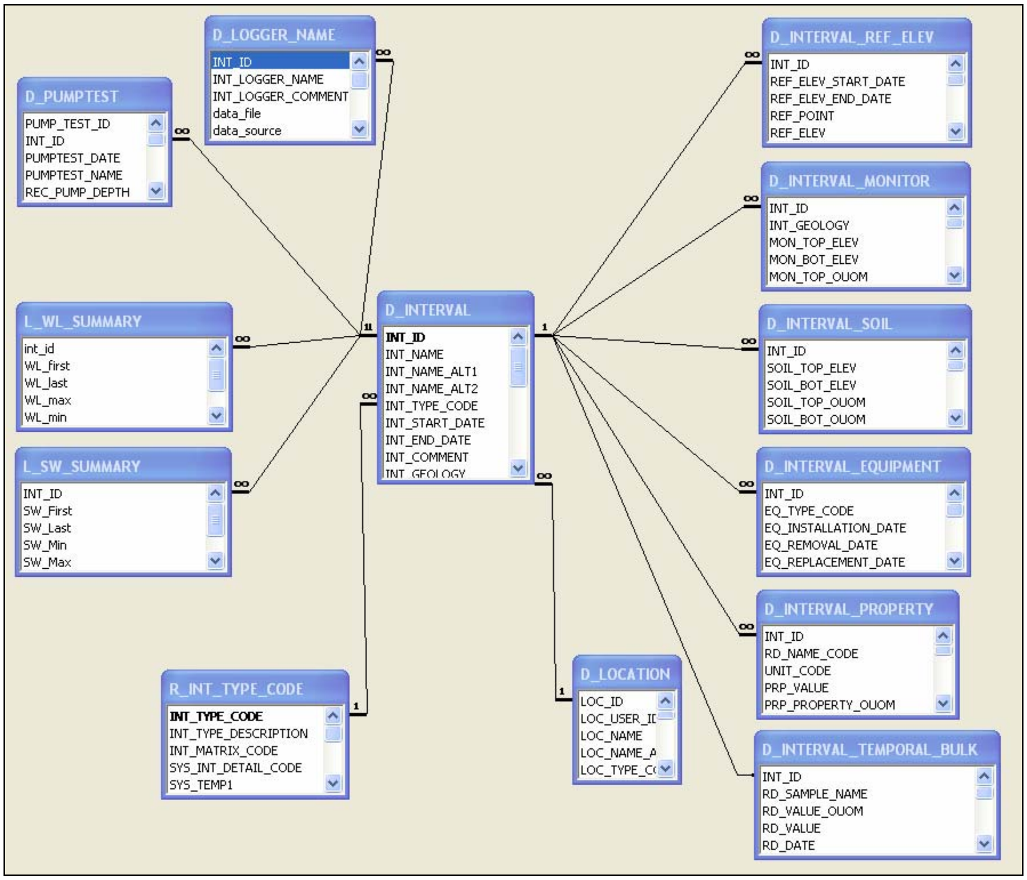


Figure 10 Main Relationships of the D_INTERVAL Table.

D_INTERVAL_MONITOR

D_INTERVAL_MONITOR records the construction details of monitoring well screens.

As a hydraulic database, not all intervals are groundwater monitoring screens – many are stream-flow gauging stations, climate stations etc. However, if the location is a well, and the interval is a screen, the table D_INTERVAL_MONITOR stores details of the screen construction, namely top and bottom elevations, material, slot size and diameter (as with borehole construction diameter, the screen diameter is assumed to be in inches).

The D_INTERVAL_MONITOR table is linked to the D_INTERVAL table in the INT_ID field.

Name	Data Type	Look-Up Table	Description
INT_ID	number	N/A	Links to the D_INTERVAL table
MON_TOP_ELEV	number	N/A	Elevation of the top of the monitor
MON_BOT_ELEV	number	N/A	Elevation of the bottom of the monitor
MON_SCREEN_MATERIAL	text	N/A	Text description of the screen material, such as PVC, stainless steel
MON_SCREEN_SLOT	number	N/A	Slot size of the screen
MON_DIAMETER	number	N/A	Diameter of the screen (inches)
MON_COMMENT	text	N/A	Comment
MON_FLOWING	number	N/A	Used to indicate if the well (screen) was flowing at the time of drilling

Table 6 Key Fields in the D_INTERVAL_MONITOR Table.

D_INTERVAL_REF_ELEV

D_INTERVAL_REF_ELEV records the reference point elevation (TOP) for monitoring wells

Elevation control on monitoring wells is necessary for accurate flow and gradient analyses. This is achieved by surveying the reference point (usually top of pipe, or TOP) for monitoring wells and correcting all ensuing water level measurements to the reference point elevation. Due to changes in the monitor construction, particularly in landfill and quarry settings, monitors are often re-surveyed on a regular basis.

The D_INTERVAL_REF_ELEV table links to the D_INTERVAL table and stores the start and end date of each survey period, plus the corresponding elevation. When water-level depths are corrected into elevations, this table is used to locate the correct elevation based on the water-level measurement date.

Name	Data Type	Look-Up Table	Description
INT_ID	number	N/A	Links to the D_INTERVAL table
REF_ELEV_START_DATE	date	N/A	Start of the period over which the elevation is valid
REF_ELEV_END_DATE	date	N/A	End of the period over which the elevation is valid
REF_POINT	text	N/A	Text description of the reference point
REF_ELEV	number	N/A	Elevation of reference point, in system units

Table 7 Key Fields in the D_INTERVAL_REF_ELEV Table

Note: for wells drilled on an angle, the angle (dip) is can be pulled from the D_BOREHOLE table during the calculation of a water level elevation. The database interface software SiteFX accounts for this correction.

D_GEOLOGY_LAYER

Perhaps the most important data type in the database, from a groundwater perspective, is the geological record from each well. This information is stored in the D_GEOLOGY_LAYER table.

This table is structured for data describing sequential layers of soil and rock (i.e., ‘to’ and ‘from’ depths and a description). Note that a second table, D_GEOLOGY_FEATURE, is structured for planar hydrogeological features such as water-found depth a fracture (the D_GEOLOGY_FEATURE table is typically used for recording the water-found depths reported by the well drillers).

The geological description table is unique from a data management perspective because it stores interpretive descriptions rather then measured numbers. In its simplest form, this can be achieved by allowing large, free text fields where the full geological description, whether it is as simple as 'sand', or a full ASTM soils description is entered. Although effective in storing the words, this format has little value from an analysis perspective as we are unable to classify the descriptions for the purpose of generating geological cross sections. A coding system is therefore often necessary in order to standardize key elements of the description.

The Ontario Ministry of the Environment (MOE) recognized the importance of standardization and offered four fields, one for colour and three for the soils description, and has used this format successfully for encoding the well driller's geological descriptions. The EarthFX data model has expanded on this format, and offers 10 fields to accommodate the elements of a more complete soils description, while including the four field data from the MOE. All fields are optional and this allows the system to accept 'sand' equally well as a full, 10-part, soils description. An example geological description for common granular fill follows:

Brown fine to medium sand fill, with gravel, some silt and trace clay, dense, moist, trace organic material

Using the original MOE classification, this description would be stored as:

- COLOUR: brown
- MATERIAL 1: fine sand
- MATERIAL 2: gravel
- MATERIAL 3: silt

From a soil facies perspective, this description neglects key elements of the description such as FILL, DENSE, and to a lesser extent the moisture and organic content. The 10-field system in the EarthFX data model includes the four MOE fields shown above, with the addition of:

- MATERIAL 4: clay
- TEXTURE: dine
- LAYER TYPE: fill
- CONSISTENCY: dense
- MOISTURE: moist
- ORGANIC: trace

The following table describes the purpose of the key fields in the D_GEOLOGY_LAYER table:

Name	Data Type	Look-Up Table	Description
LOC_ID	number	N/A	Links to D_LOCATION
GEOL_DESCRIPTION	memo	N/A	Free text description of the geological unit
GEOL_TOP_ELEV	number	N/A	Elevation of the top of the geological unit (in system units)
GEOL_BOT_ELEV	number	N/A	Elevation of the bottom of the geological unit (in system units)
GEOL_MAT_COLOUR_CODE	number	R_GEOL_MAT_COLOUR_CODE	Material colour code

GEOL_MAT1_CODE	number	R_GEOL_MAT1_CODE	Primary geological material
GEOL_MAT2_CODE	number	R_GEOL_MAT2_CODE	Secondary geological material – the *with* material
GEOL_MAT3_CODE	number	R_GEOL_MAT3_CODE	Tertiary geological material – the *some* material
GEOL_MAT4_CODE	number	R_GEOL_MAT4_CODE	Trace material
GEOL_COMMENT	memo	N/A	Comment
GEOL_LAYERTYPE_CODE	number	R_GEOL_MAT_LAYERTYPE_CODE	Describes the family of material (e.g., till, fill, topsoil)
GEOL_CONSISTENCY_CODE	number	R_GEOL_MAT_CONSISTENCY_CODE	Describes the consistency of the material (e.g., firm)
GEOL_MOISTURE_CODE	number	R_GEOL_MAT_MOISTURE_CODE	Describes the moisture content of the soil (e.g., dry, damp, most
GEOL_TEXTURE_CODE	number	R_GEOL_MAT_TEXTURE_CODE	Describes the texture of the soil (e.g., stiff, soft)
GEOL_ORGANIC_CODE	number	R_GEOL_MAT_ORGANIC_CODE	Describes the organic content of the soil
GEOL_MAT_GSC_CODE	number	R_GEOL_MAT_GSC_CODE	GSC geomaterial code

Table 8 Key Fields in the D_GEOLOGY_LAYER Table.

The following figure illustrates the relationships between the D_GEOLOGY_LAYER table and the associated Look-Up tables, and serves as a good template for building queries based on the well geology.

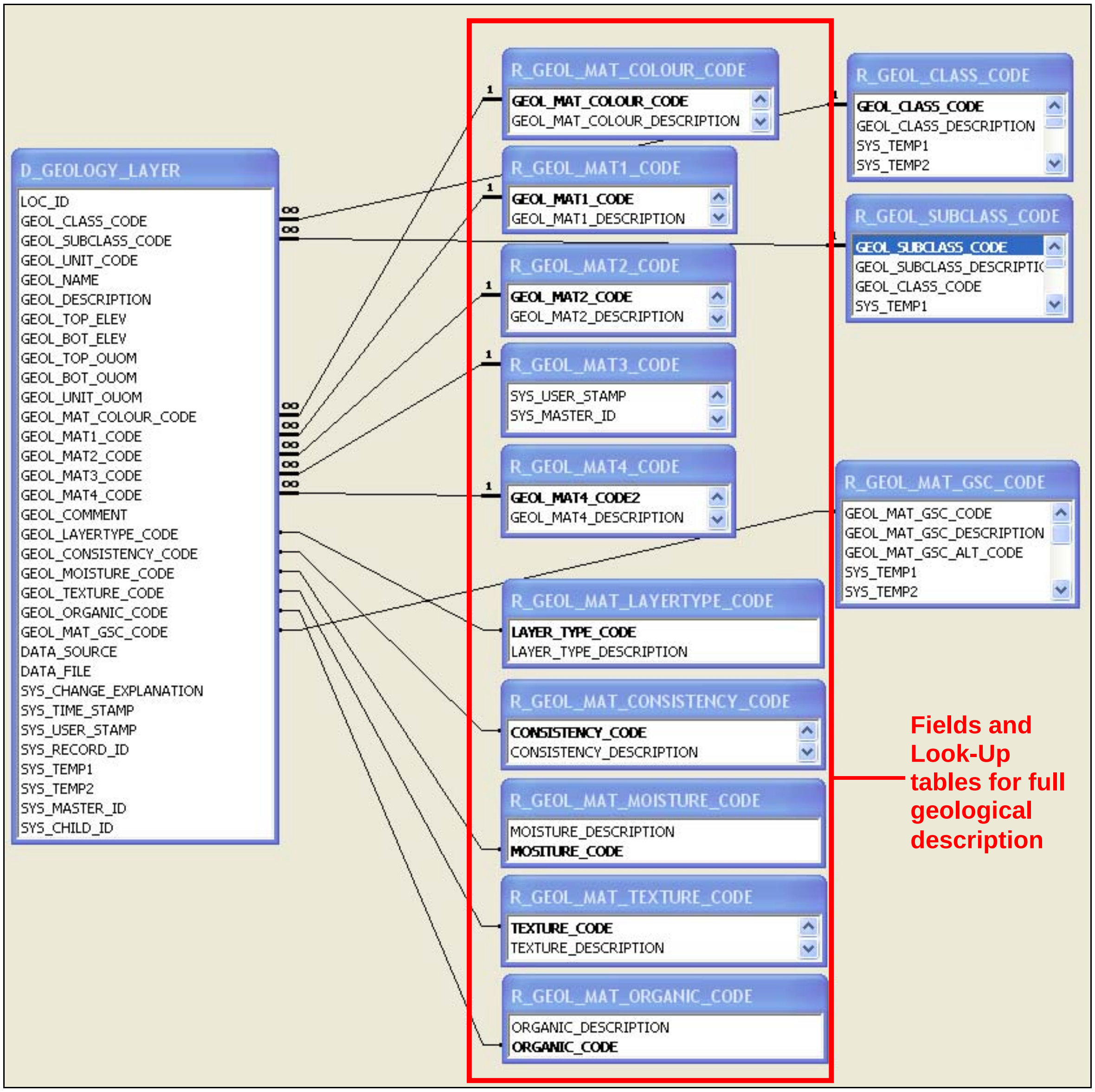


Figure 11 Geology Table Fields and Associated Look-Up tables

Time Series Tables

Water levels, water quality data, climate data, pumping rates and surface water flow rates and other time-varying data constitute the temporal data in the database. In broad terms, the database contains information *about* measuring points (the locations and their properties) and information *from* the measuring points (the temporal data).

The volume of temporal data increases rapidly in comparison with the location data. The temporal data accounts for the majority of the database volume and requires a higher level of attention in the data model design. The following text outlines the current data model structure for temporal data and the logic used for its development.

The temporal data aspects of the data model are developed based on two underlying, themes:

1. First, fundamentally all temporal data is alike, having a station name, date and time, parameter and measured value – whether water levels or air temperature is being measured. In this respect, all temporal data should be stored in a single table (and often is in large enterprise scale databases).
2. Second, because MS Access is the software of choice, the data model must be designed to operate effectively in a Jet environment (Jet is the database management system for MS Access). This means adhering to a physical file size limit of 2 GB (in practice anything over 1 GB suffers from being overweight) by minimizing the number of fields in the temporal data files (adding a field adds approximately 4 MB of file size for every million records). The following lists the actual field requirements of the temporal data sources.

Water Level Data

Water-level data is stored in the D_INTERVAL_TEMPORAL_WL table and key fields in this table are listed below.

Name	Data Type	Description
INT_ID	Number	Interval ID number, linked to the D_INTERVAL table
RD_VALUE	Number	The 'interpreted' value, presented in system units
RD_DATE	Date	Date (and time) of reading
RD_NAME_CODE	Number	Code from Look-Up table R_RD_TYPE_CODE that identifies the type of reading (all set to 'water level' for this table)
UNIT_CODE	Number	Same as above, except for the reading units. Used Look-Up table R_UNIT_CODE
RD_EXCLUDE	Yes/No	Yes/No to indicate that the record should or should not be used for analysis purposes
COMMENT		Optional comments

Table 9 Key Fields in the D_INTERVAL_TEMPORAL_WL Table

Water Quality Data

Water-quality data is stored in two tables:

1. D_INTERVAL_TEMPORAL_DETAIL: This table stores the details (e.g., sample name, date, lab etc.) of the water-quality samples.
2. D_INTERVAL_TEMPORAL_DATA: This table stores the actual water-quality values.

These two tables are linked through a sample identification number (SAM_ID). The relationship between these two tables, and D_INTERVAL is shown in the following figure.

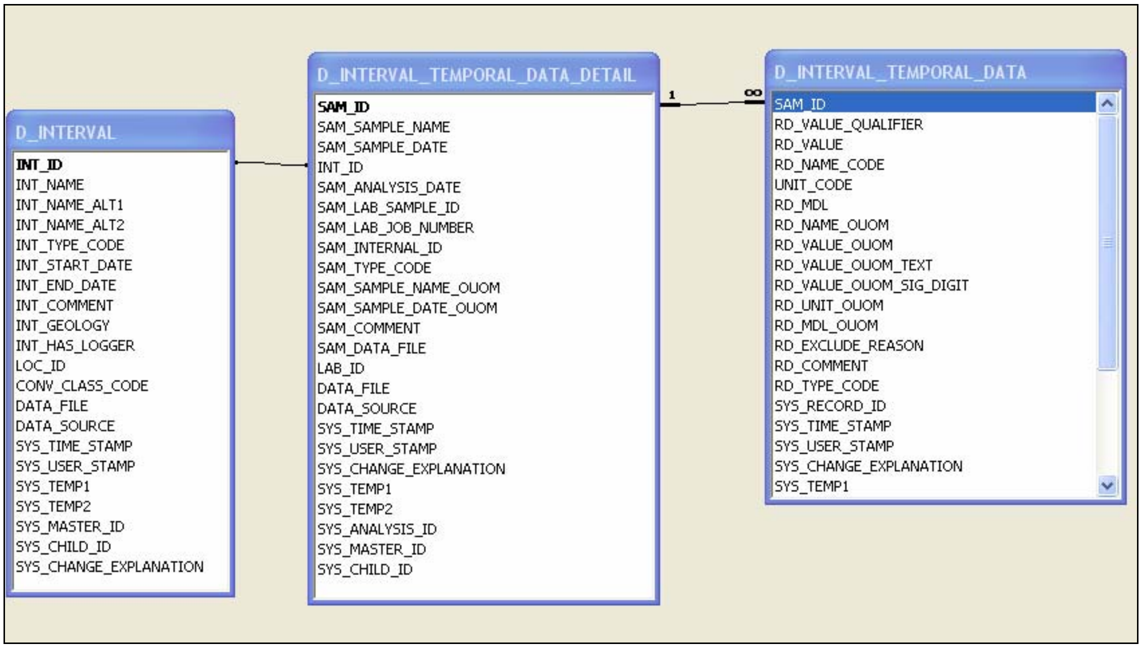


Figure 12 Relationships for Water-Quality Tables.

Details of the key fields in these two tables are shown in below.

Name	Data Type	Description
SAM_ID	number	Sample ID number
SAM_SAMPLE_NAME	Text	Sample name
SAM_SAMPLE_DATE	date	Sample date
INT_ID	number	Interval ID number. Links to D_INTERVAL
SAM_ANALYSIS_DATE	date	Analysis date
SAM_LAB_SAMPLE_ID	text	Laboratory sample ID number
SAM_TYPE_CODE	number	Sample type code. Links to R_SAM_TYPE_CODE
SAM_COMMENT	text	Optional comments

Table 10 Key Fields in the D_INTERVAL_TEMPORAL_DETAILS Table.

Name	Data Type	Description
SAM_ID	number	Sample ID number. Links to D_INTERVAL_TEMPORAL_DETAILS table.
RD_VALUE_QUALIFIER	text	Value modifier (e.g., <)
RD_VALUE	number	Reading value (in system units)
RD_NAME_CODE	number	Reading name code. Links to R_RD_NAME_CODE
UNIT_CODE	number	Unit code. Links to R_UNIT_CODE
RD_MDL	number	Method Detection Limit (in system units)
RD_COMMENT	text	Optional comments

Table 11 Key Fields in the D_INTERVAL_TEMPORAL_DATA Table.

The relationships of these tables are shown in the following figure:

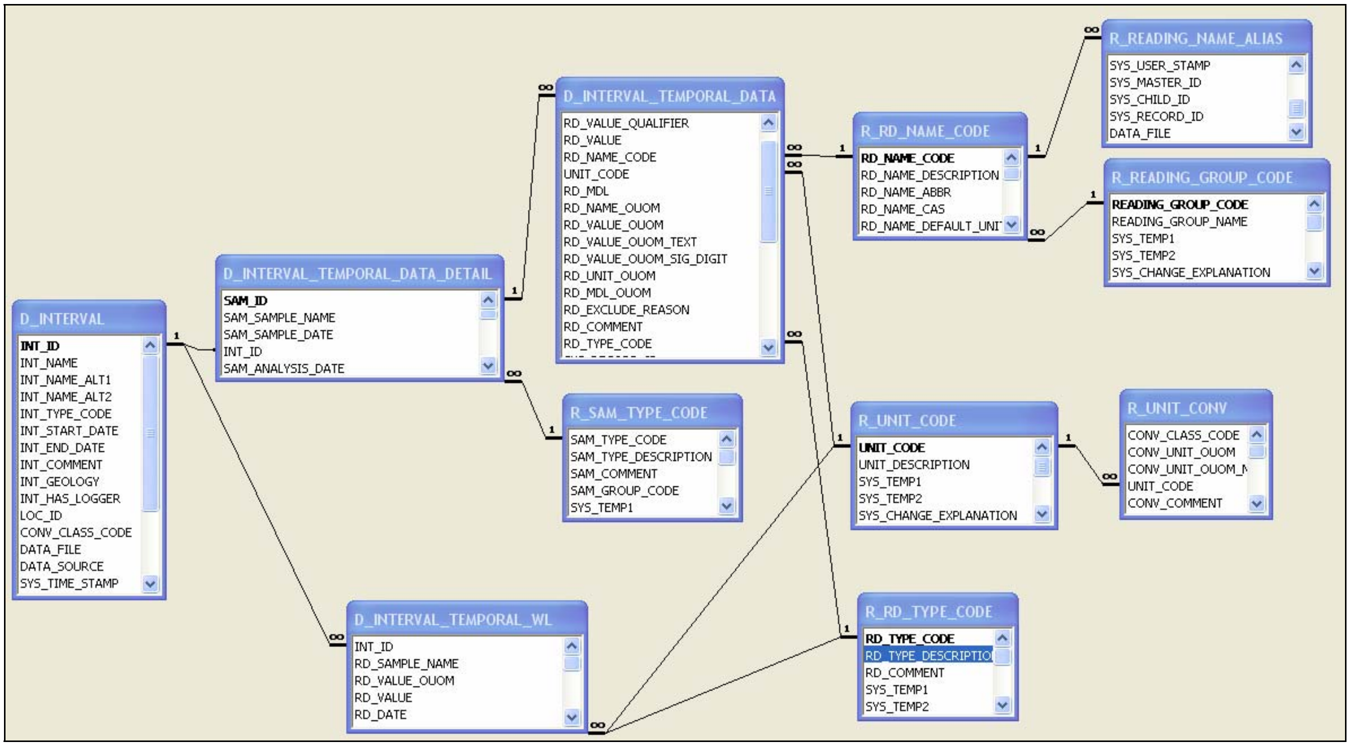


Figure 13 Relationships for D_INTERVAL_TEMPORAL Tables.

Application of the Data Model

To better illustrate the functionality of the data model, this section provides several real world scenarios of data sources, and explains how they are handled in the data model. In practice, most of the details described below are handled by SiteFX and require only minimal user knowledge of the data model details.

Multi Screen Installations

Multi-screen installations are a common groundwater monitoring techniques for hydrogeologists seeking to fully quantify local hydrogeological conditions. They include one or more boreholes, and one or more screens (or monitors) in each well. Three styles of multi-screen wells have been considered in the development of the data model.

Nested wells

Multi-pipe well

WestBay type installations

Nested Wells

Nested wells are comprised of two or more boreholes, each with one or more permanent monitoring wells. They are commonly used in landfill settings where detailed hydrogeological information is required from the near surface geological materials, typically in a multi-aquifer setting.

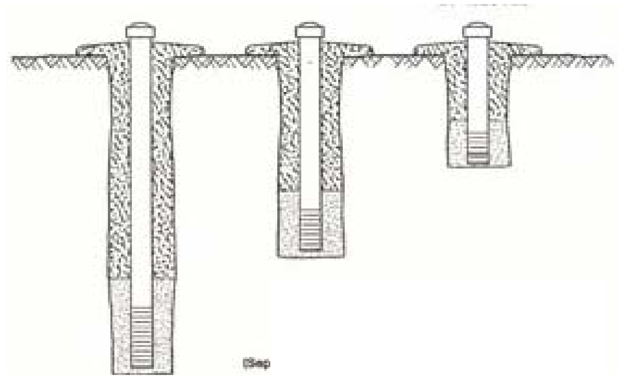


Figure 14   Nested Well Example

From an installation perspective, usually the deepest well is drilled first and the recovered soil logged to establish the geological profile. All subsequent holes are simply drilled to the required depth based on the interpretation of the first borehole. All boreholes are fitted with permanent monitoring wells, usually a single monitor per borehole.

From a data analysis perspective, we think of the nested wells as a single well, with multiple monitors. The monitoring location will be represented as a single dot on the map, for which there is a single geological record, and multiple suits of temporal data from each of the monitors.

From a database perspective, management of the data is controlled by the following concepts.

- o Although there are multiple boreholes, from an analysis perspective, we think of the nest as a single well and a single dot on the map. The database must be able to aggregate all data behind a single location.
- o Each borehole is different, with a unique UTM position and borehole construction details, which must be accurately recorded. A separate record is therefore necessary for each borehole.
- o Each borehole is completed with one or more permanent monitoring wells, with a unique name and reference elevation, and with a suite of temporal monitoring data (such as water levels and water quality data).

The data model handles nested wells as fully independent wells, linked only through the use of the Master Location feature in the D_LOCATION table. Each well therefore has a unique record in the D_LOCATION table and the D_BOREHOLE table, except that the primary (deepest) well is flagged in the D_LOCATION table as being the ‘master location’, and the remaining boreholes, as listed in the D_LOCATION table, are related to the master by inserting the master well ID number into the ‘master well field.

Arguably an alternate approach would be to list a single location, with multiple boreholes and multiple intervals. In fact, this was the structure used by the data model for many years. The shortcoming of this approach is that we can carry only a single set of coordinates for the well (X-Y positional information is a function of the location, and is found in the D_LOCATION table), and we therefore loose the exact position of the other associated wells. While this loss may be tolerable for common multi-well installations where the boreholes are usually within a few metres, it is not acceptable for municipal pump house scenarios where the individual wells may be hundreds of metres apart. Consequently, each well is listed in the D_LOCATION table, with independent coordinates, and linked into families using the LOC_MASTER_LOC_ID field.

Multi-Pipe wells

A multi-pipe well is a single borehole with multiple monitoring wells installed, screened a various depths. These wells are not commonly installed as the seal between the wells screens cannot be assured. Multi-pipe wells are managed in the data model with a single location, a single borehole and multiple intervals.

WestBay type installations

WestBay instrumentation is an inflatable packer string inserted into single, usually cored bedrock, well. Packers are assembled to isolate zones of hydrogeological interest, and specialty-monitoring equipment is lowered into the hole to collect samples from individual packer intervals.

Alternatively, a routine double packer assembly may be installed in a cored well to collect water level and water quality data from a specific interval, and the assembly then shifted to a new interval of interest.

In both scenarios a single location and borehole is used in the data model with multiple intervals. In the case of the second example, the intervals are short-lived; often only long enough to perform a simple slug test. The D_INTERVAL table therefore carries an INT_TYPE_CODE field (foreign key to the INT_TYPE_CODE field in the R_INT_TYPE_CODE table) and start and end date/time fields to track the many possible (and overlapping intervals) that could exist in a single borehole.

Municipal pump houses

For municipal applications, pump houses are a key data collection location. Pump Houses are generally considered as a single dot on the map, but may consist of one or more productions wells (not necessarily within the walls of the pump house) providing water levels, pumping rates and water quality measurements, chlorinators, providing a series of readings to gauge the chlorine application rate and several associated monitoring wells.

From a data model perspective, this is a more complex scenario. Firstly, each item, be it a well or chlorinator, is assigned as a location, with an appropriate location type flag set in the LOC_TYPE_CODE field. Usually one of the production wells is set at the master location, with all other locations referenced to the master location ID number. In the event the location is a borehole (production or monitoring), there will be a corresponding record in the D_BOREHOLE table. All locations have a corresponding interval, and appropriate interval type, and for chlorinators, the D_INTERVAL_EQUIPMENT table is used to track serial numbers etc.

Weather stations with multiple probes

Weather stations with multiple temperature and radiation probes are becoming more common in our ongoing efforts to better quantify groundwater recharge. Assuming all probes are grouped on or immediately around a single tower, this scenario is best handled as a single location, with a corresponding record in the D_CLIMATE table to track general details of the climate station tower. Multiple intervals will then be assigned to the location, one for each probe. The D_INTERVAL_MONITOR and / or the D_INTERVAL_EQUIPMENT tables can be used to track specific details of each monitoring point.

Surface water stations

Surface water stations range from a simple staff gauge to a stilling well with associated mini piezometers and seepage meters to measure base flow. This scenario is managed in the data model with a single location for the stilling well or staff gauge (set as the master Location), and associated locations for the mini-piezometers, linked using the LOC_MASTER_LOC_ID field. The mini-piezometers will be flagged as a borehole, or other groundwater monitoring type location, and likely have a corresponding record in the D_BOREHOLE and

D_GEOLOGY_LAYER tables to track the construction and soil profile. Each location will have an independent interval in the D_INTERVAL table. Flow and water quality data is linked to the interval corresponding to the master location. For staff gauge settings, a single location record and single interval record will suffice.

It is noteworthy that at the time of writing, revisions to the data model are underway to accommodate panel diagram and gauging station calibration cure data.

Index