GEOC 589 – Advanced Topics: Geology of Critical Minerals

Project Report

The Compilation, Organization, Analysis and Visualization of Geochemical Data Related to Numerous Studies Underway at the New Mexico Bureau of Geology and Mineral Resources - Report of Work in Progress

Submitted by

Mark Leo-Russell
GEOC 589 §3
New Mexico Institute of Mining and Technology
May 4, 2022
corrections: May 5, 2022

Project Report

Table of Contents

1. Introduction

This report outlines the initial stages of an on-going project I am undertaking as the Database Administrator at the New Mexico Bureau of Geology and Mineral Resources and in conjunction with this class and other graduate coursework I will be doing at New Mexico Tech.

2. Project Overview

The purpose of this project is to develop and deploy an integrated, sustainable system for the collection, storage, management, retrieval, analysis, and visualization of geochemical data related to numerous projects at the New Mexico Bureau of Geology.

Geochemical data are a vital component of many of the research projects currently underway at the Bureau. To date, most of our geochemical data are stored, managed and analyzed in multiple ad-hoc spreadsheets with a few desk-top databases (MS Excel and Access, respectively). These have not been developed in a unified, cohesive manner—rather as individual independent files created by the geologists, researchers and other investigators within the Bureau. While some have attempted to utilize “standard” formats, there remains a multitude of file versions, formats, and techniques used within these data sets.

2.1 Goals and Objectives

The aim of this project is to develop a unified, integrated system and procedures for the collection, storage and use of geochemistry data across the Bureau and its partners. A core database system is to be designed and implemented and associated applications for the entry, editing, analyses, visualization, and extraction of those data. In other words, this project will encompass constructing the database itself as well as developing applications, primarily web applications, for working with the data.

While there are numerous contracts currently underway at the Bureau, and several more in the pipeline, this project will focus on developing data systems specifically for the following:

- Gallinas Mountains – Exploration and delineation of critical minerals especially Rare Earth Element (REE) deposits.
- Cornudas Mountains – Exploration and delineation of critical minerals especially Rare Earth Element (REE) deposits.
- REE/Coal – Assessment of coal and coal wastes for REE recovery.

- Laramide Porphyry Copper – Analysis of Copper Porphyry deposits in Arizona and New Mexico (in cooperation with the Arizona Geological Survey).
- Various water quality and water filtration studies, including a project between New Mexico Tech (NMT) and the Navajo Technical University (NTU).

2.2 Project Phases

This endeavor is definitely a "project in progress" with requirements, specifications, schedules, and personnel expected to change over time. However, to get started I am initially dividing the development into five main phases.

1. Background Research
2. Data Inventory
3. Database Design and Deployment
4. Development of Data Entry and Retrieval Tools
5. Data Analysis, Reporting and Visualization

Background research will involve literature reviews and other investigations into previous geochemical database projects (within the Bureau and elsewhere), existing database schemas, industry best practices, design considerations and related topics. This phase of the project is currently underway.

The data inventory phase will be the review, inventory, organization and cataloging of existing geochemical data sets housed within the Bureau. As previously mentioned, most of these data are in various spreadsheet files with a few stored in Access databases and even fewer within our existing SQL Server database environment. This part of the project is already well underway as I work with numerous researchers within the Bureau. Based on my previous experience reviewing and consolidating an organization's data, potentially this part of the project could be the most time-intensive and complicated phase of the work. So far my efforts have generally been well received by Bureau data owners but I anticipate there may be some resistance to moving data into a central repository. Staff perceptions of "my data, my computer" are not an uncommon hurdle to these types of projects and will be addressed as they arise.

Designing and constructing the database is the nuts and bolts part of my job as DBA. Since many of the Bureau's existing databases reside within a MS SQL Server database system, the geochemical data will probably be housed on the same system. SQL Server (SQLS) is a commercial enterprise-level database management system (DBMS) widely used in industry

and academia. Even if our geochemical and other data needs were to grow ten-fold or more over the next few years, SQLS can scale up to handle such requirements.

Another DBMS software being considered for this project are the MySQL/MariaDB platforms. These are open-source systems also widely used in enterprise systems. In fact, the Bureau's website, Geoinfo (geoinfo.nmt.edu), is presently backed by multiple MySQL databases. From the perspective of this project either SQL Server, MySQL, or MariaDB would serve equally well.

Issues related to the overall database design, regardless of the specific database system used, will be addressed later in this report.

Developing the database for geochem data is only part of the equation. Another equally important component are the software applications used to enter, edit and retrieve data from the database. In the past the Bureau has used Access forms for such data entry. Personal experience over the past year has shown these to be resource-intensive to develop, maintain and support especially as the number of users needing to use such systems has increased. For this project, and other data systems I am developing for the Bureau, web-based applications will be the primary way for users to interact with their data. Web applications have the advantage of only requiring an Internet connection and a web browser in order to use the program. These days most people have such connectivity and software on many devices including their desktop computer, laptop, and/or smart phone. Unlike an Access front-end where a copy of the file and associated configuration must be setup and maintained on every device for every user, a web application does not involve any application or configuration on the user's device apart from the aforementioned web browser. Development and deployment of the data applications are done entirely on centralized servers—computers that run the applications for all users utilizing the system.

This web-based applications architecture has been in common use for over 20 years. The Google ecosystem used at Tech for e-mail, documents, and other office tasks is a well-known and understood web platform most staff and students at the Bureau use everyday. The web applications to be developed for the geochemical data will be functionally similar to systems we already have.

The last phase of this project, and potentially the most influential in terms of how users will interact with their data, will be the development of capabilities for analyses, reporting, and visualization of data. Presently, much of our researchers' time and effort with their data are spent with one-off methods for manipulating the data for incorporation into Geographic Information Systems (GIS), extractions for other analyses, and reformulating data into the various formats required as deliverables by our customers. Here the aim is to have a set of

software tools pre-built to stream-line and standardize this aspect of the data life-cycle. Not only will this greatly reduce the time needed for processing data for reports and other aspects of our projects, errors will be reduced and reproducibility greatly improved.

The last phase of the project, designing and building the analysis, reporting and visualization components, is really about turning the raw data in the database into useful information, and ultimately into actionable knowledge and wisdom. This follows the *Data-Information-Knowledge-Wisdom* model first presented using slightly different terminology by Kenneth Boulding in 1955 (Boulding, 1955, pp. 21–32; Wallace, 2007, pp. 1–14) and later reformulated into the D-I-K-W elements we use today by Nicholas Henry in 1974 (Henry, 1974, pp. 189–196). The DIKW hierarchy is explained in more detail in Appendix 1 of this report.

3. Primer on Databases and Spreadsheets

A group of software applications often used to store and manipulate data are spreadsheets. These programs, which first came to prominence with the adaption of personal computers in the 1980s, started life as “electronic ledgers” intended to replace the gridded sheets historically used in accounting and finances. From there spreadsheets have grown in features and complexity to be used for many applications outside their creators’ wildest use cases. Unfortunately, they are too often used as “databases” for a multitude of data include geochemistry. This problem is not unique to the earth sciences. I have seen spreadsheets used as a student information system (SIS) in small school districts, to record and track patients’ medical data, and in a whole host of areas well beyond electronic ledgers.

While programs like MS Excel, LibreOffice/OpenOffice Calc and Google Sheets may look like tabular data from a database, they are not database tables. Spreadsheets freely intermix the data, logic and presentation layers. This is certainly beneficial when used as a ledger but causes multiple problems when used as a database.

Database systems on the other hand, focus solely on the data layer and leave most of the logic and virtually all of the presentation to other programs. Database software are specifically designed to minimize the storage space needed for data while maximizing performance when reading and writing data (often contradictory requirements: space versus speed). Larger database systems also provide data redundancy, simultaneous transactions, distributed storage and many other capabilities designed to improve performance and reduce the chance of data loss or corruption (Daniels, n.d.; EarthSoft, 2018; Ganchev, 2022; Redderson-Lear and Pot, 2020).

3.1 Spreadsheets for Geochemical Data: Issues and Problems

Despite their shortcomings spreadsheets are commonly used for data including geochemistry and other earth-science data. Numerous problems arise. One reoccurring issue is the lack of standard spreadsheet formats for data. As previously mentioned spreadsheets mix the data, logic and presentation together which individual users can adopt as they see fit. This results in varying layouts, formats and formulae. Even within similar work at the same organization users will rearrange and reformat data in order to address an immediate problem encountered during their work. One such seemingly simple issue I found in my research of geochem data are the orientation of rows and columns within different spreadsheets. Authors mix the x and y orientation of the data: some authors list the analytes or chemical species along the x-direction (horizontally across the columns) with samples on the y-axis (vertically down the rows); other authors do the exact opposite. Figure 1 illustrates these cases.

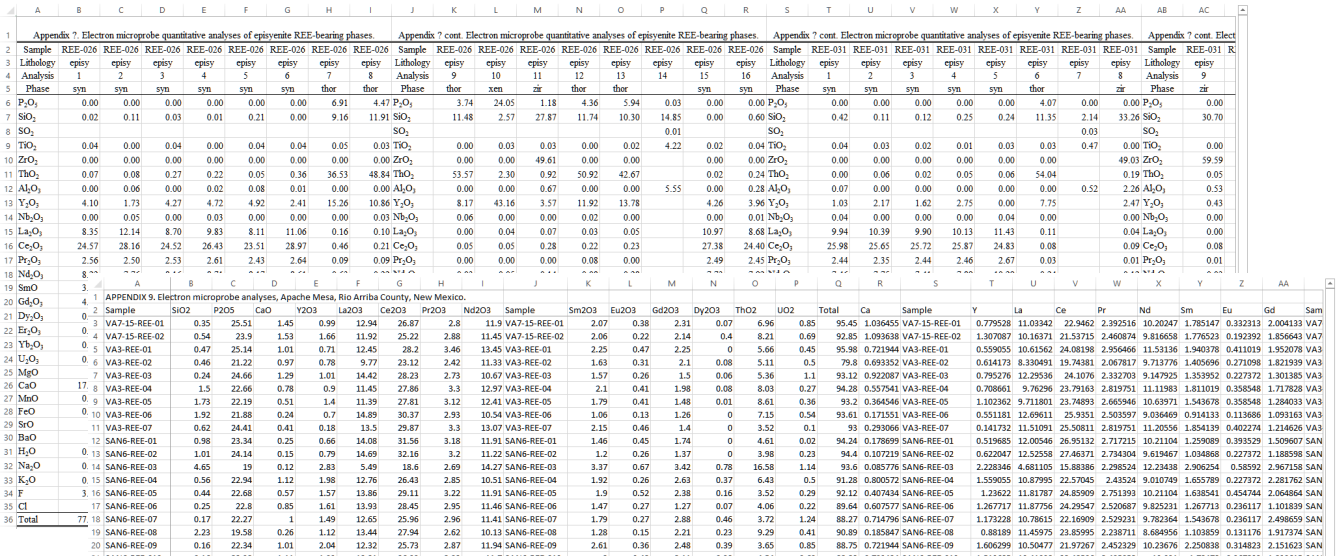

Figure 1: Varying orientation of geochemical data in spreadsheets: Analytes (chemical species) along the columns with Samples down the rows [foreground image] versus Samples along the columns and Analytes down the rows [background image].

Another problem with using spreadsheets as a database is the ability to mix different data in the same column. Typically this is used to create section titles or subtitles to denote groups of data. This is a case of mixing the presentation (i.e., the desire to separate groups of data) with the data (the actual text of the title or subtitle). Extra programming logic is needed to work around the presentation formatting. Figure 2 shows an example where extra programming (SQL code in this case) is needed to treat such a spreadsheet as a data table in order to bring the geochemical data into a Geographic Information System (QGIS software in this example).

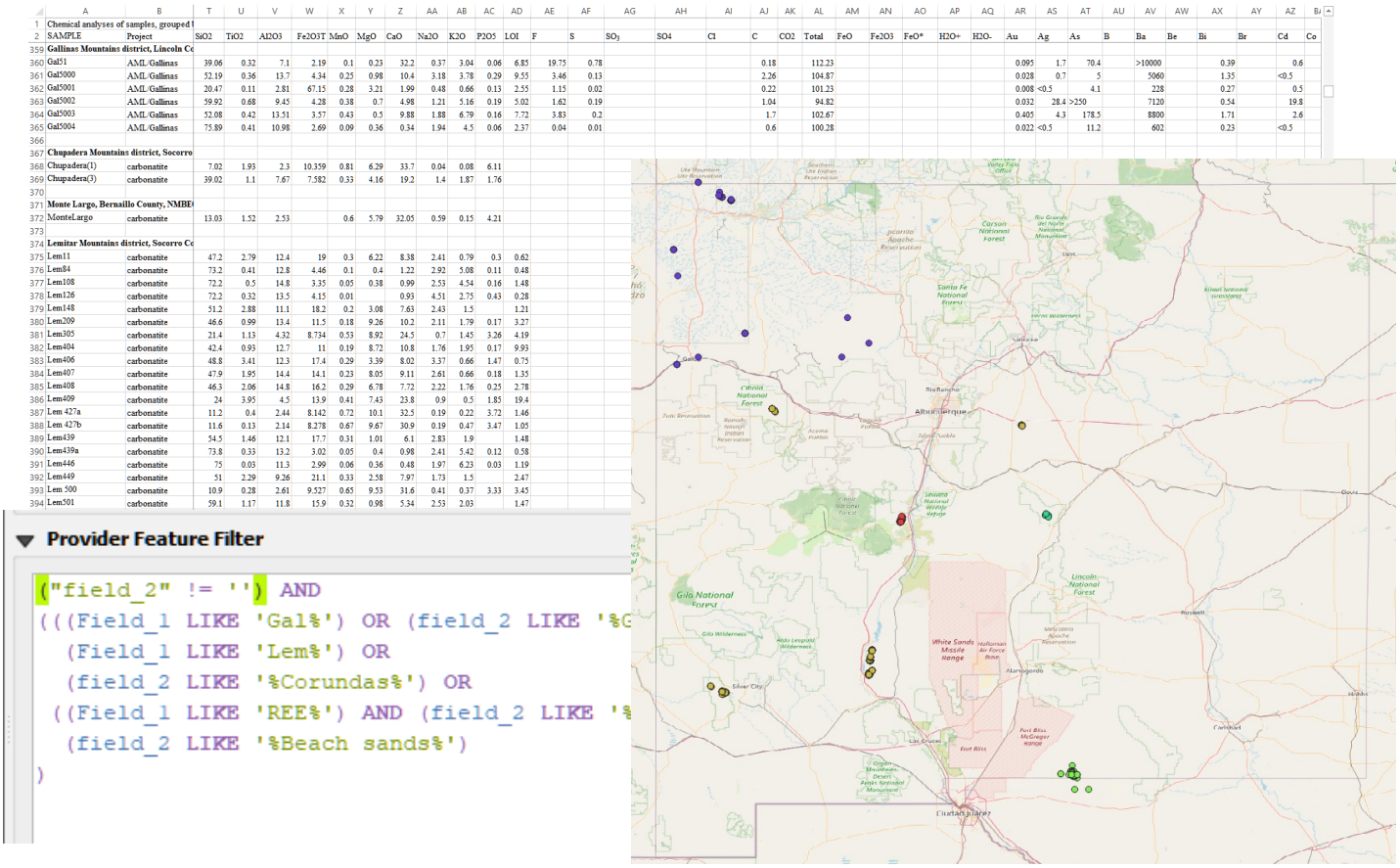

Figure 2: Example of geochemical data in a spreadsheet with intermixed "label rows" [background image] requiring programming to include or exclude various rows [SQL code, lower left] in order to plot locations on a G.I.S. map [right] (data adapted from McLemore, 2020).

3.2 Relational Databases

In contrast to spreadsheets, a database system, specifically a Relational Database Management System (RDMS), divides the data into logical units in order to optimize storage and retrieval. Usually a DB is not concerned with how the data are processed or presented—just the data layer—the logic and presentation layers are handled by different pieces of software. Many relational database systems do include logic facilities for data manipulation in the form of a programming engine. Structured Query Language or "SQL" is the language used within relational databases. Strictly speaking the data storage and retrieval capabilities of a database system are separate from the SQL processing but they are often tightly coupled in order to maximize performance.

The overall structure of a relational database is known as a *Schema*. The process of designing and creating a schema is called *Data Modeling*.

A more thorough discussion of relational databases and their components are included in Appendix 2.

4. Database Structures and Schemas for Geochemical Data

When I started researching and investigating existing geochemistry database designs and previous work done in this area I assumed there would be a "standard" design used for geochemical data, or at very least, a small number of similar schemas. This assumption turned out to be completely wrong!

The literature illustrated various database designs and schemas that depended on when the database was constructed, who and where the design originated, the extent of the data to be handled, the variability of the geochemical analyses to be stored, and many other factors. In short, this was becoming a much bigger and more complicated engineering endeavor than I had anticipated.

My literature review brought to the front two fundamentally different database designs for geochemical data. Authors use a variety of nomenclature and taxonomy but I organize things into two groups: *Sample-Centric* or *Monolithic Record* schema, and *Analyte-Centric* or *Chemical Species-centric* schema.

4.1 Sample-Centric or Monolithic Record Schema

The *Sample-Centric* or *Monolithic Record* schema has all geochemical analysis values for a given sample in a single record. Since there are many analytes or chemical species that can be tested on for any given sample, this makes for many columns or fields for each sample's record, a so-called "wide table." At first look, this table design looks like the spreadsheets of geochemical data previously discussed. Specifically, with a field for each analyte, this table would be very similar to the analytes-across-the-x-axis example shown in the foreground of Figure 1.

Typically, we would not have analysis values for all chemical species for all samples in the table. Thus many of the fields are blank for some records which results in a "sparse table." In this design a position is reserved for each analyte for every record even if that particular analyte does not have a value (a blank or null in database terminology).

A relationship diagram for this schema is shown in Figure 3. In database relationship diagrams each rectangle represents a table in the database and the items listed in each rectangle are the columns or fields in that table. The lines indicate the relationships between tables. These diagrams do not show actual data values in the tables, just the tables, fields and relationships. Note in this example the first table on the left of the diagram and the number of fields listed. These are the fields for each chemical species that can be stored in that table, whether a particular sample record has values for those fields or not.

This design has the advantage of having all analysis values for a given sample stored in a single record. This simplifies data retrieval in that the records do not need to be joined with other tables or records to get results for a given sample or set of samples. On the downside, if an additional analyte or chemical species needs to be included, a new column must be added to the database table. This can result in the table being even more "sparse" than before because all the existing records will be blank for the new field. From an operational perspective, modifying a database schema to add a new column to a database table usually involves the database administrator or someone in the organization with appropriate database skills.

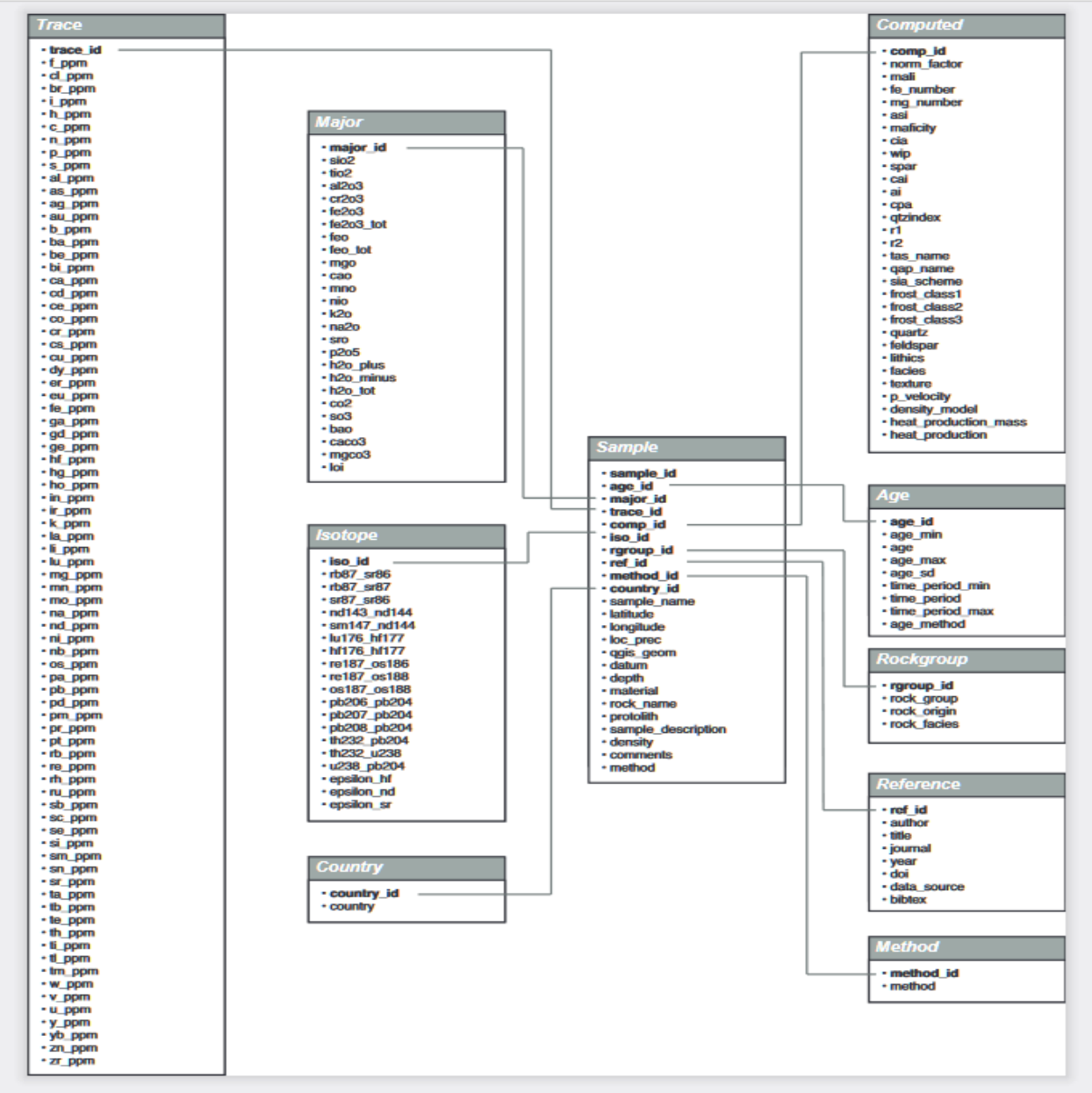

Figure 3: Database diagram showing an example of a *Sample-Centric* or *Monolithic Record* schema for geochemical analysis data. Note the 'Trace' table on the left has fields for each possible analyte or chemical species to be stored in the table (Gard, et al, 2019).

4.2 Analyte-Centric Schema

In the *Analyte-Centric* or *Chemical Species-Centric* schema data for each sample are stored as separate records for each analysis done for the sample. Each record has at least three values: the sample ID, an analyte or chemical species ID, and an analysis value. There may be additional fields for result units and testing metadata such as the test date, vendor, instrument used, etc. Even with additional metadata fields this table design would be considered a “narrow table.”

Unlike the sample-centric schema, the analyte-centric design will not result in sparely populated tables since a record is not created for a given sample’s analyte unless there is an analysis value to be stored. Likewise, if a new analyte or chemical species is included in the analyses the database schema does not need to be modified to accommodate the new values. A new analyte may need to be added to other tables in the database (eg., an analyte lookup table) but this does not involve a schema change. In that regard this schema is much more flexible and adaptable to changes in the chemical analyses performed in the organization.

A disadvantage to this design is it is more complex to extract all analyte values for a given sample or set of samples. Whereas the sample-centric schema would involve extracting a single record for a given sample, the analyte-centric schema involves multiple records to get the same results set. An experienced database programmer can do this with relative ease but to the uninitiated this can be a confusing and error-prone task.

Figure 4 on the next page shows a database relationship diagram for this schema. Note there is not a single table that includes fields for every chemical species. In database parlance, this design is considered more *normalized*.

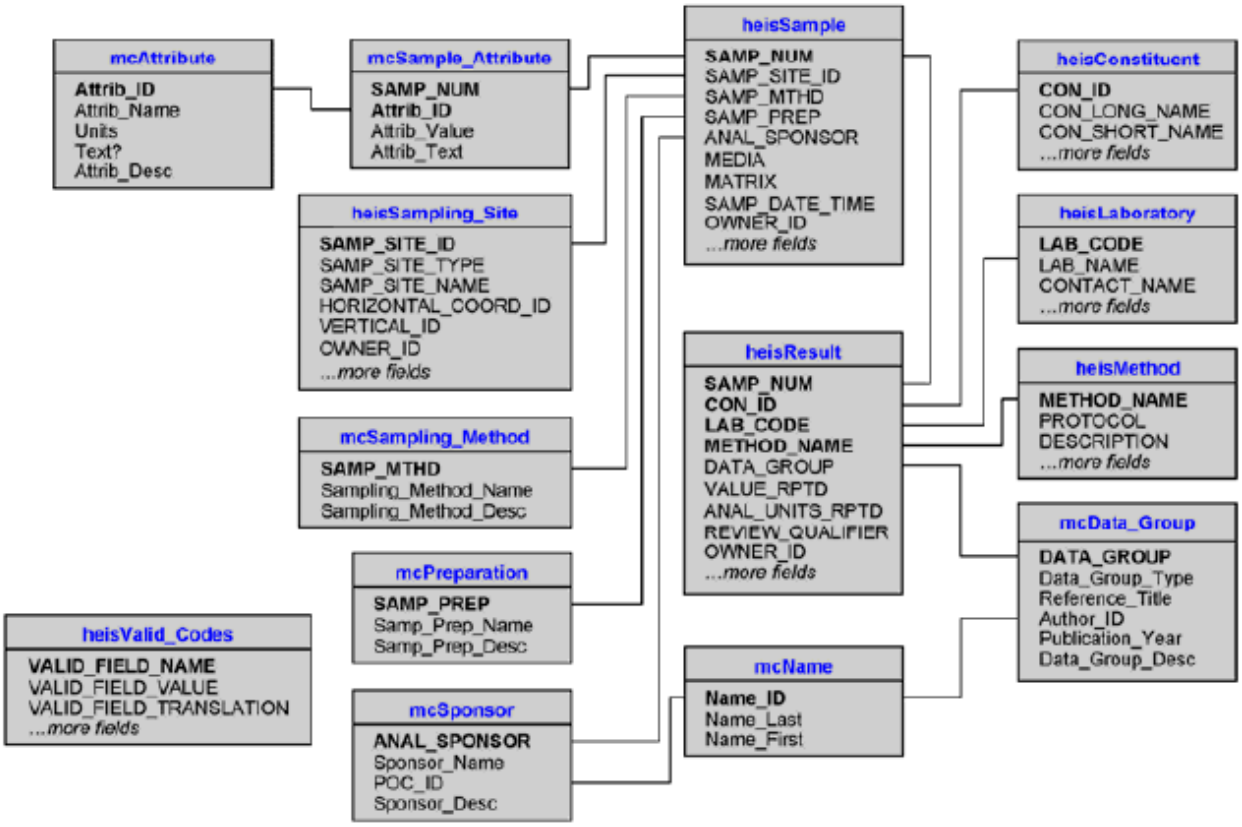

Figure 4: Database diagram showing an example of a *Analyte-Centric* or *Chemical Species-Centric* schema for geochemical analysis data. Note the absence of any table that explicitly includes fields for all possible analyte or chemical species to be stored in the table (Mackley, et al, 2008, p. 4).

5. Project Development and Future Plans

Much work remains to be done on this project. In fact, the work completed this semester is just the tip of the iceberg. While I was definitely expecting to be further along and even have some early prototypes built to demonstrate the project, the research and initial design phases are proving more complicated. I know from a career of software design and engineering, including developing new databases, that these extra design efforts are time well spent. Everyday I learn something new about databases and more applicable, something new about geochemistry, its data, and how we use and wish to use those data in the Bureau and beyond.

5.1 Project Schema

One aspect of the project that has come into focus so far is the overall schema design to be used in developing the Bureau's geochemical database. While the sample-centric schema

has advantages of simplicity and would be more familiar to those accustomed to dealing with data in spreadsheets, it has limitations and lacks the flexibility to accommodate the variety of geochemical data we are sure to encounter in the future.

My current thinking is to utilize the analyte-centric design for the database. This design will involve more complicated data extractions but its flexibility and adaptability outweigh the complexity. The schema avoids sparsely populated database tables, allows for “an indefinite number of attributes to any sample” (Mackley, et al, 2008, p. 4), is a more normalized design by relational database design standards, and allows a more simplified way of including metadata describing the specific sample-analyte tests.

5.2 Future Work

In the coming months, database work on the Critical Minerals, REE/Coal, Copper Porphyry studies and other Bureau projects will all include efforts on this project. As mentioned earlier in this narrative, the work is very much a “project in progress” so tasks, resources, and schedules are sure to change. To summarize future work:

- Construct a prototype database and import existing project data into the system.
    - Gallinas Mountains and REE/Coal data are first candidates.
- Develop and test queries, views, and other programming to manage and otherwise manipulate the data.
- Design and develop web-based applications for users to enter, edit, extract and report on geochemical data.
- Integration with existing databases currently used at the Bureau.
    - Gallinas, REE/Coal, NM Mines.
- Develop tools and techniques for easy extraction of data into various formats needed by researchers and for project deliverables.
    - Geographic Information Systems (GIS), statistical software, visualizations, etc.
- Develop interactive applications for staff, students, and the general public to explore, interact and analyze the vast amounts of geochemical data currently at the Bureau.

6. Summary

The design, development, deployment and support of databases and related systems are clearly my responsibilities as the Bureau of Geology's Database Administrator. As I have always tried to achieve throughout my career, the objective is to develop and deliver systems that simplify user tasks, streamline operations, and chase that elusive goal of providing meaningful information, knowledge, and ultimately wisdom to my customers. This project, like many others I am involved with at the Bureau, gives me the opportunity to meld my first love of geology with my profession love of technology into a cohesive sustainable system. It will be a hell of an adventure!

References

Boulding, Kenneth, 1955. Notes on the Information Concept. *Exploration*. Toronto. 6: 103–112. CP IV.

Daniels, John, n.d. The Advantages of Databases Over Spreadsheets. Tech Support Reviews, https://www.techwalla.com/articles/the-advantages-of-databases-over-spreadsheets.

EarthSoft, 2018. Databases versus Spreadsheets. https://earthsoft.com/2018/06/07/databases-versus-spreadsheets/.

Ganchev, Martin, 2022. Database vs Spreadsheet: What's the Difference? SQL Tutorials, May 3 2022, https://365datascience.com/tutorials/sql-tutorials/database-vs-spreadsheet/.

Gard, Matthew, Hasterok, Derrick and Halpin, Jacqueline A., 2019. Global whole-rock geochemical database compilation. Earth System Science Data, Volume 11, Issue 4, Oct 17 2019, https://essd.copernicus.org/articles/11/1553/2019/.

Henry, Nicholas L., 1974. Knowledge Management: A New Concern for Public Administration. *Public Administration Review*. 34 (3): 189–196.

Mackley, Rob D., Serkowski, John and Last, George V., 2008. Revised Status Report on the Creation of a Preliminary Data Model and Dictionary for a New Petrologic Database. Pacific Northwest National Laboratory, PNNL-17663, Sep 24 2008, https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-17663.pdf.

McLemore, Virginia T., 2020. Chemical analysis of samples grouped by mining district or area. https://geoinfo.nmt.edu/staff/mclemore/projects/mining/REE/McLemoreMasterChem_v5.xlsx

Mongodb.com, n.d. Relational vs. Non-Relational Databases. https://www.mongodb.com/compare/relational-vs-non-relational-databases.

Okrepilov, Egor, 2021. What is the Data, Information, Knowledge, Wisdom (DIKW) Model? weje.io, updated Apr 7 2022, https://weje.io/blog/data-information-knowledge-wisdom.

Ontotext.com, n.d. What is the Data, Information, Knowledge, Wisdom (DIKW) Pyramid? https://www.ontotext.com/knowledgehub/fundamentals/dikw-pyramid/.

Redderson-Lear, Katie and Pot, Justin, 2020. Spreadsheets aren't databases—stop using them like one. App-Tips, October 12, 2020, https://zapier.com/blog/database-vs-spreadsheet/.

Wallace, Danny P., 2007. Knowledge Management: Historical and Cross-Disciplinary Themes. Libraries Unlimited, Westport, Connecticut.

Appendix 1: The Data-Information-Knowledge-Wisdom (DIKW) Model

Although this project is primarily concerned with the design and development of a database system and issues related to the storage and management of geochemical data used in various Bureau projects, data are just one element in a larger scheme. The *Data-Information-Knowledge-Wisdom pyramid*, also known as the *D-I-K-W model* or *DIKW hierarchy*, is a framework commonly used in knowledge management to conceptualize the relationships between data and higher cognitive constructs (Henry, 1974; Okrepilov, 2021; Ontotext.com, n.d.).

The model is typically illustrated as a four-tier triangle as shown in the following figure. It should be noted that although the tiers are usually drawn with hard lines, in practice they most often seamlessly blend from one to another.

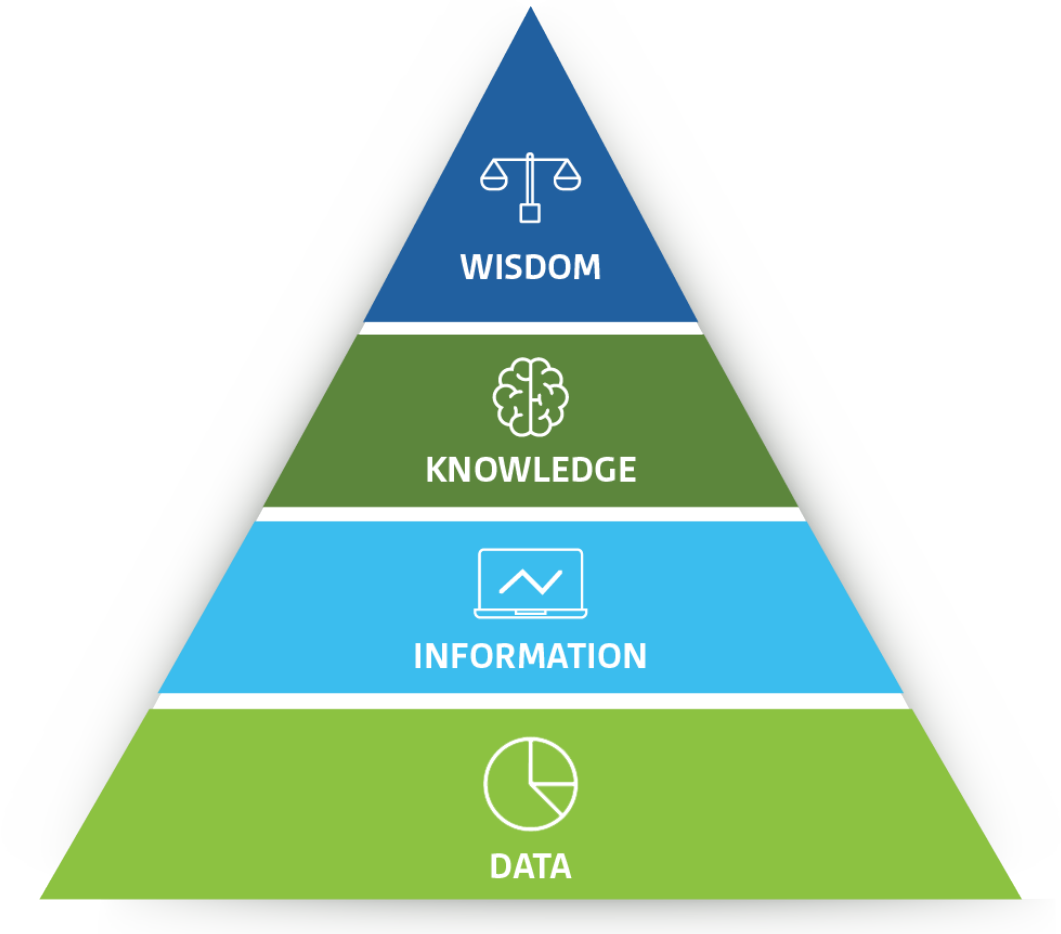

Figure A1: The common four-tier triangular representation of the Data-Information-Knowledge-Wisdom (DIKW) hierarchy model (Okrepilov, 2021).

In the DIKW scheme *data* are the "raw stuff" collected and stored for further use. Some authors say the data are unorganized collections (Ontotext.com, n.d) but in the context of this project, data have some organization and structure within a database system (explained in more detail in Appendix 2). For geochemistry data these are not just laboratory measurements but also include ties to the context of the measurements, known as relational data and metadata. These may include references to specific samples (rock specimens, water samples, ash samples, etc.) which in turn reference location data such as outcrop ID, well number, and the like.

Typically we think of data as being numbers or text but in modern systems data includes imagery, video, audio and other objects of interest. Of course, in a computer all data are reduced to numbers—binary numbers in fact—but we rarely deal with things at such a low level. We interact with the objects created from the numbers.

Moving up the DIKW hierarchy *information* are data that have been processed or otherwise "cleaned" in a manner that provides the user with a more meaningful perspective on the issue at hand. The is the layer where we begin to process data into analyses, graphs and visualizations for a specific purpose (Ontotext.com, n.d.; Okrepilov, 2021).

At the *knowledge* level, our information's relevance and its relationships with other information are brought to bear. *How* the information applies to our goals and objectives brings it into the realm of knowledge (Ontotext.com, n.d).

Finally, at the apex of the DIKW hierarchy is *wisdom*. This is the level where we apply the knowledge learned to answer questions such as "why do something" or "what is the best decision" (Ontotext.com, n.d). Wisdom gives us the ability to predict some events' scenarios based on past experience (Okrepilov, 2021).

There are many interpretations of the DIKW model with varying definitions of the components and counter arguments to each. The purpose here is not to dissect the particulars but rather to illustrate that data are not a means unto themselves. Rather, we are trying to build systems and procedures to use data for higher purposes. Building a database system is a means to an end. We do not collect and manage data just for the sake of collecting and manging data; we are trying to achieve other objectives. In the case of the work done at the Bureau of Geology, we aim to provide objective information on earth-science related areas so our customers, often policy makers, can make knowledgeable decisions related to the geological resources, their development, and use within New Mexico and elsewhere. My intention here in describing the DIKW pyramid is to provide a framework for thinking about how we best utilize our data and data systems and the development thereof.

Appendix 2: The Relational Data Model and Relational Databases

Database architecture is a sub-specialty within computer science which goes back to the earliest days of computational design. One of the oldest designs and one still in wide use today is the *Relational model* which underpins *Relational Databases* also known as *Relational Database Management Systems* (RDMS). While other database designs have taken hold in recent years such as the NoSQL, Key-Value DBs, and Graph DBs (Mongodb.com, n.d.), RDMSs still rule the roust in terms of the number of databases and database systems in use worldwide.

The basic unit of a relational database is the table. While a table looks like the columns and rows of a spreadsheet, a DB table is much different. Within a table data are divided into rows, also known as *records*, and columns, also known as *fields*. A row or record is an individual entity within the table while the columns or fields are the separate distinct values that comprise that entity. Each field is defined to hold a specific type of data such as integer numbers (eg., 1, 987, 0), real or floating-point numbers (1.0, 99.88, 0.0, 3.14159), or various collections of characters (“ABC”, “My Name”, “1”). In a given column or field, all records will have the same type of data in that particular column. This is one of the key things that differentiates databases from spreadsheets. In a given column in a spreadsheet we can have different types of data from row to row. For example, in the “A” column, the first row could have a text value while the second row in the same column could have a number. A database table would not allow this mixing--the text values and numbers must be stored in separate columns.

It should be noted here that databases, and computers in general, treat the numeral one (“1”), the integer value one (1) and the real number one (1.0) as completely different things. In common use we humans tend to mix those together and collectively say “the number one” or “the value of one.” It can be a bit confusing when first encountering how these are handled differently in databases and other computer applications.

The relational part of RDMS comes from how data in different tables are connected or linked together. The field in one table has the same values as those in another table’s column and they are used to relate the records in the two table. For example, we may have a table with various fields describing a study site. We’ll call these *Waypoints*. In the Waypoint table there will be fields for the waypoint identifier (WaypointID) and various fields that record specifics of that point (latitude, longitude, elevation, land ownership, type of property, etc.). In another table we have data on samples collected at each waypoint. Fields for each sample will include SampleID, type of sample, date collected, etc. Most importantly,

each sample record will have a field that records the waypoint location where that sample was collected. Since the Waypoint table and Sample table both have values for WaypointID, we can connect or relate the two tables together. This type of connection is known as a *foreign key relationship* and is at the heart of how relational databases function. Relationship diagrams for two different types of geochemistry database designs are shown in Figures 3 and 4.