Geological Data Management for the Small Explorer

20th January, 2012

Laurie Veska Geological Data Services: www.lveska.com

Solid foundations

It is essential for a small, recently listed or low market-cap exploration company to lay a solid foundation for the management of its geological data. The geological database a company uses provides this foundation, and is a key asset, with its value being increased by improving the inherent quality of the data. Improving the quality of this data, which can provide good dividends both now and in the future, is the focus of this article.

Acquisition of data

Typically, a company will have acquired its tenement holdings with an accompanying package of legacy geoscientific and spatial (GIS) data. This data represents the sum total of the previous explorers' investment in the ground that makes up the tenement holdings. It will take many forms; and is usually of widely varying quality and completeness. In its most basic form, the data will be a pile of field notes and associated maps and plans. You maybe a little luckier and find a collection of quizzically named spreadsheets, which when opened display an artistic collage of colourful cells, the exact meaning of which has been lost long ago. At least in this instance much of the data is in digital form. Yet there are many good reasons why a company should be thinking beyond spreadsheets.

Thinking beyond spreadsheets

Why do so many small companies continue to use spreadsheets when a database is the better choice? A spreadsheet has serious drawbacks when used for data storage. It is cumbersome to retrieve selective data from a spreadsheet. Spreadsheets offer little or no data validation and little or no protection against data corruption from well-meaning but poorly trained users (Read: Data sorting). So why do so many small explorers stick with spreadsheets? It is often because of familiarity and a lack of understanding of the alternatives.

Some companies will have discovered they have one or more 'desktop' databases, most commonly MS Access or the equivalent. This is a promising sign because there is a good chance at least some data will be in relational tables. However, before an existing desktop database is used, a company should consider the following questions:

1. Does the database design suit your current business requirements?
2. Does the database have a means of data viewing, entry and editing using well-designed forms?
3. Is there an inbuilt means to create presentation reports for the stored data?

If the answer to any of these questions is no, then either a commercial solution, or some in-house development should be considered. If you are already in possession of a 3D mining package, most provide some means of creating a geological database, however, these databases are often too simplistic and inflexible for other uses. Often in a small company the expense, or indeed the need, for a fully-fledged commercial data-management solution can't be justified. However, there are other solutions available to improve or extend the design and functionality of your current database. There is also the option of creating a new solution from scratch.

There are good reasons to consider your data management needs early, so as not to compound problems over time. Ignoring the health of your geological data will mean that when it comes time for a resource calculation you may have to pay for a consultant and a significant part of their time (and your money) will be spent making sense of your poorly-organised data.

Organising data - some solutions

Drill hole and Samples Database

For the small explorer, the starting point (or scope of the undertaking) is usually a means of managing reconnaissance ‘point’ data e.g. soil samples, stream sediment samples, rock chip samples etc., as well as drill hole (or linear data). A practical and proven solution for the small explorer is to use a desktop database which also has some means of creating data-entry forms, a graphic user-interface (GUI) to aid in creating queries, a method to produce basic reports such as a summary drill hole listing (for exploration announcements) and a detailed drill hole report (annual reporting to the government body).

Microsoft Access is a common choice in industry for both a data entry and reporting tool, especially for those with little programming experience. This is a valid choice for the small operation; there is also a good business case for using MS Access as a front-end CRUD tool, with a separate instance for the back-end database (where only the data will reside). Some suggested back-end options are MS Access again, or for a sturdier solution; SQL Server Express 2008 R2, or Oracle XE which are both freely down-loadable tools and very capable. With respect to CRUD, I’m not talking about a ‘coating or an incrustation of filth or refuse’ (to which the geologist's amongst us are intimately familiar). I am referring to Create, Read, Update and Delete, basic functions of a computer database system.

The separation of front-end from back-end is a good future-proofing idea; the data tables in the back-end are linked to the front-end database containing the forms, queries and reports. This means that work can be done on the front-end without worrying about taking the database off-line or data synchronisation later.

The Database Model

For the beginning geology database modeller there are a number of common pitfalls. Whilst there are important criteria to generally follow in order to create or model a database, there is no single, correct design. Many volumes have been written regarding general database design, however I will discuss only the most problematic pitfalls as they apply to geological and drilling-related data repositories.

It is crucial to understand the importance of storing your geoscientific data in the most logically consistent and efficient manner. It is a common misconception that you store your data according to a specific end use. An example would be the modelling of a database in such a way that collar, sample and perhaps even analysis data are held in the same database table. This may have been designed to eliminate the use of joins in GIS/3D software, but this practice severely limits the flexibility and use of the stored data for other future software or purposes. I have reviewed a database from a medium sized mining company where the ‘Prospect’ field was duplicated in several tables, including the collar table, the geology table and the assay table. Data redundancy such as this can lead to data anomalies and corruption. As an example, if the prospect for a given collar required editing, it would have to be changed in literally thousands of records, instead of one edit in one lookup table.

This leads to the practice of separating the storage from the presentation layers. Ideally you will have one storage layer for your data, but many presentation layers. For example, a GIS presentation layer for each of your GIS software tools of choice, a 3D presentation layer for your modelling software of choice and so on, the key point being that you can easily add presentation layers as the business need arises with the proviso that your storage layer has been properly designed from the start. Presentation layers are commonly created through the use of database Views (saved queries).

To achieve the best protection for your valuable geological data, it is important to understand that the current generation of desktop database engines typically provide inbuilt tools and functionality to ensure the data in your database is accurate and consistent i.e. has integrity.

Whilst no amount of programming can prevent every type of error that could be introduced, it is important to learn how to utilise the provided tools to the best of your ability in order to maximise the validity and quality of your geoscientific data. You will have heard of the old adage: Garbage In – Garbage Out , well let's keep the garbage out!

Ensuring data integrity

There are four primary types of data integrity: entity, domain, referential, and user-defined. In general terms; entity integrity applies at the row level; domain integrity applies at the column (or field) level; and referential integrity applies at the table level.

1. Entity Integrity

AKA: Hey, my drill hole is in there twice!

No duplicate rows. Entity Integrity ensures that the data that you store remains in the proper format as well as remaining comprehensible. Every row has a unique field that can't be null or empty.

In a typical table designed to hold collar information it is very common practice to designate the name of the drill hole or costean as the Primary Key or unique, not null field. Many drill databases generated from 3D modelling software such as Surpac do this. Text fields traditionally make poor primary keys, especially fields that the user manually enters, such as the drill hole name.
This happens both in commercial software, and more than likely in your very own database. For a small scale, simple to maintain system suited to the small explorer, this is understandable particularly given the user-friendliness it offers.

Drill hole 'DDH-97 GREENFIELDS-009' is the same as 'DDH97 GREENFIELDS-009' isn't it? You and I know that, but to your database they couldn’t be more different.
During the data collation stage where you may be initially loading your database tables with data, the practice of using the drill hole name as the Primary Key is going to be at its highest risk in terms of data integrity between tables.
You may find that the geology logs are not displaying for a drill hole you could swear has been logged and the data entered. The end result may be that you enter this log information again and duplication occurs - your entity integrity has been compromised.
One partial solution here is to provide the user a means of easily performing a search for existing data before new rows are created. A search performed in the background to display drill holes with similar names as you enter your new hole, for example. Another, even better solution is to use integer type fields to link tables, such as a drill-id.

2. Domain Integrity

The values in any given column fall within an accepted range. (Validation)

It is important to make sure that the data entered into a table is not only correct, but appropriate for the fields it is entered into.

Some familiar examples may include:

1. A numeric field to hold end of hole (E.O.H) depth.
2. A date field (with date picker in your entry form) for the drill start date, drill end date
3. Drill hole collar azimuth in the range (0-360 degrees)
4. Drill hole collar dip in the range (-90 - +90).
5. Multi-field validation: hole survey depths must be less than or equal to E.O.H depth.

If the range of acceptable values for a given field must conform to a list of values (e.g. a rock code or alteration type), then typically a lookup list would be employed within the application, with or without a referential integrity constraint (see next) applied at database level on a matching lookup table to enforce this referential integrity.

3. Referential Integrity

Foreign key values point to valid rows in the referenced table. In geological database terms, a consequence will be the requirement to have a drill hole collar entered first, before you can enter its geological log. Additionally you will be prevented from deleting a drill hole collar before first deleting all its associated logs, down-hole survey entries, RQD and Recovery values etc.

4. User-defined Integrity

The data complies with applicable business rules. For instance you may have a rule that states all analyses are to be entered in parts per million units rather than percent. You may require that a drill hole name be entered in the form: “DDH“+ Year + Prospect Name. It is almost certain that you will want to prevent lithology intervals for a given hole from overlapping.

Summary

Given that geologists are normal, fallible human beings a database design and its application code should be able to anticipate and prevent many types of errors, there should also be an expectation that every effort is made to enter data correctly. Some errors may be impossible to trap systematically (for example, an incorrectly typed name or rock description), but any business rules which can be defined should be enforced in your database. Avoid the temptation to rely solely on vigilance by end-users to ensure accurate data.

A database and application is ultimately worthless if the data is of poor quality and cannot be relied upon. The geological database a company uses is a key asset, with its value being increased by improving the inherent quality of the data - and improving the quality of data can provide good dividends both now and in the future.