

# Efficient Chrono-Stratigraphic Database Structure in GIS

Carlos Gabriel Asato, Servicio Geológico Minero Argentino

Buenos Aires, Argentina.

E-mail [gasat@secind.mecon.gov.ar](mailto:gasat@secind.mecon.gov.ar)

## ***Abstract***

In recent years the Instituto de Geología Regional y Recursos Minerales, has been developing the Regional Geology GIS Data Standards. One of the most important tasks was the development of the chrono-stratigraphic database definition and design.

In order to allow the use and exploitation of digital geological maps, a simple database structure had been made. This case differs from traditional chrono-stratigraphic database designs where there are only direct relationships between geological units and their chrono-stratigraphic definition.

The design concept involves a hierarchical data organization taken from the chrono-stratigraphic column where the relationships between classes and sub-classes were preserved.

In more detail it is used a special code definition that was designed in order to have a direct correspondence with a specific chrono-stratigraphic division, and indirect relation to their own hierarchical position.

There is also a complete chrono-stratigraphic definition, ordered in separate fields, and organized by chrono-stratigraphic level. The data structure formed by geological layer, table relationships, chrono-stratigraphic code and chrono-stratigraphic table design allow to the users indexing, classification, and direct searching and scalable design.

## ***Introduction***

GIS design involves a lot of different analysis techniques based in the analysis of data spatial structures, attribute tables and related database.

Analysis could be made using GIS tools, only if data structures have an appropriate design. Database design could be understood as important GIS part when querying analysis will be performed.

Most of the database design problems are solved using the relational model. Nevertheless, the special case of chrono-stratigraphic data is very complex because ages have hierarchical categories and special relationships, temporal relative meaning, sometimes names change from region to region, etc. Age database design has to consider all of the problems related to the management of age data.

This paper shows a development on chrono-stratigraphic databases taken in GIS context and also explains what is possible to do using different queries and analysis.

## ***Design Ideas and Principles***

### Table Structure

Sometimes the first idea that people have about geology and its age database is any kind of one to one relationship, where each element only have one entry for the age item.

Suppose that is wanted to retrieve Jurassic. In the flat database sense, Jurassic means only Jurassic and not means Lias, Dogger and Malm. Then if you have sub-units defined as Lias, Dogger or Malm you cannot retrieve those because there are not relationships in the database between Jurassic and its subdivisions.

So if we want to fix this bug, we can add an item where will put the age parental relationships. That's look as a better solution because a parental and hierarchical relationship are defined and then is possible to retrieve Jurassic, Lias, Dogger and Malm as the same thing, but other problems will appear. If we want to retrieve "Mesozoic", with two or three categories below, the process becomes complex and computing expensive because is necessary to make recursive querying over the same table.

And now make an exercise, think in the simplest schema for data organization and description. Is possible design a database where each age or component is treated as one thing and where ages have a complete description of its principal properties and dependencies like this:

```
Object name: JURASSIC
Object rank: SYSTEM
Stage       Null
Series      Null
System      JURASSIC
Erathem     MESOZOIC
Eonothem    PHANEROZOIC
Age_index   12020000
```

Or like this:

```
Object name: BAJOCIAN
Object rank: STAGE
Stage       BAJONIAN
Series      DOGGER
System      JURASSIC
Erathem     MESOZOIC
Eonothem    PHANEROZOIC
Age_index   12020202
```

Then this schema kept the parental relationships in a simple manner and also simplest querying procedures could be applied.

Generally speaking, this kind of data organization is quite similar to hierarchical tree relationships, but conceptual differences exist, this special case is not link oriented, is data oriented.

### Index System

Indexing methods optimize the speed of queries and facilitate the ordering of data. The simplest idea about key index definition is to define a numerical

scale beginning at zero or one for the oldest ages and then increment values by one to the newest ages. This kind of index is very simple and is possible to retrieve data in ordered form, but the problem is that doesn't reflect ages hierarchy and relationships.

Special numerical index method was adopted. It is a hierarchical code where ages have different values calculated with this formula:

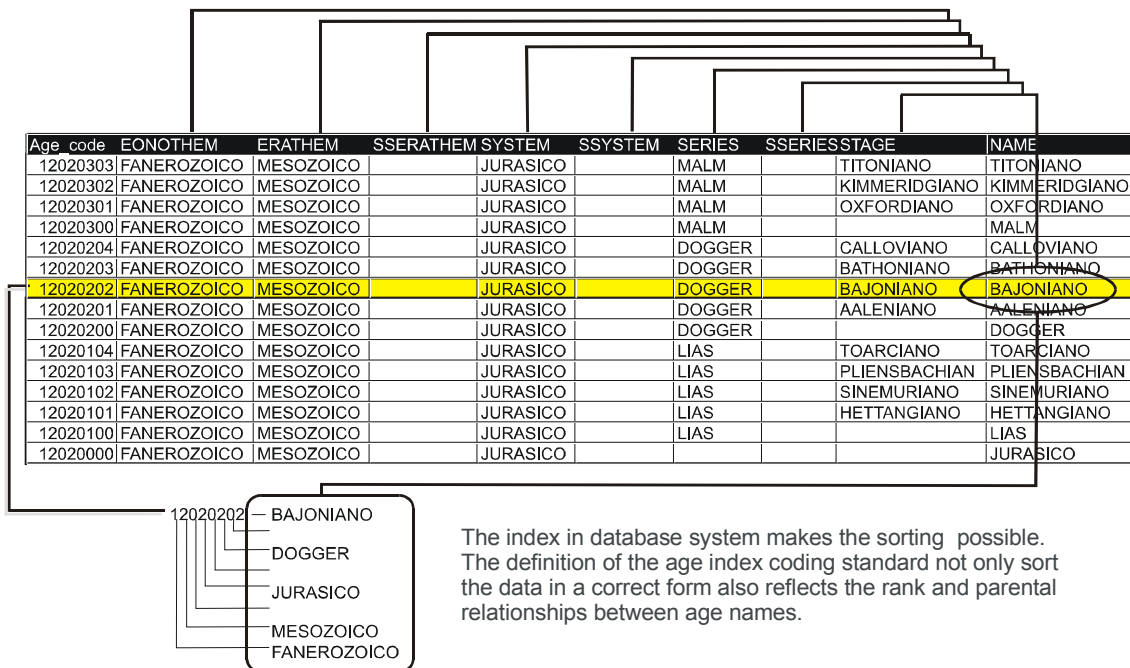
$$\text{INDEX} = \text{Age\_weight} * \text{order} + \text{parent\_age1\_weight} * \text{age\_1\_order} + \dots + \text{parent\_age\_n\_weight} * \text{age\_n\_order}$$

Where weight corresponds to the age hierarchy and order is the numerical position respect the parent age.

Let see how is calculated the Jurassic index:

Jurassic is the second system in the Mesozoic erathem and its weight is 100. Mesozoic is the Jurassic parent and is the second erathem in the Phanerozoic eonothem and its weight is 1000. Phanerozoic is the Mesozic parent and is the second eonothem in the stratigraphic chart and its weight is 10000. Then the result is 22200. See that the latest two values were filled with two zeros, which correspond to series and stages.

The idea about hierarchical index is that this kind of hierarchical code can be manipulated by mathematical procedures and moreover is possible to expand the analysis possibilities.



The index in database system makes the sorting possible. The definition of the age index coding standard not only sort the data in a correct form also reflects the rank and parental relationships between age names.

Other problem that simple scalar index codes doesn't take in account are the different stratigraphic columns used in different regions or countries. So if somebody wants to introduce or modify regional ages then will have to modify the complete stratigraphic schema and index. With the hierarchical index system, is possible to modify the schema introducing or deleting age names

maintaining the general framework and modifying only the smaller category indexes. As the biggest categories were maintained, correlation is possible at least at the level where there are stratigraphic coincidences. See that in first time JURASSIC and Bajonian index were calculated using the Argentina's stratigraphic chart, and the latest Jurassic was calculated using the international chart. Both of these codes are different but is possible to manipulate the indexes in order to make relationships with both chrono-stratigraphic databases.

### Map to Table Relationships

The following diagram shows the database structure where relationships between geological map attribute table, the relationship table and the table with the age definitions are explained.

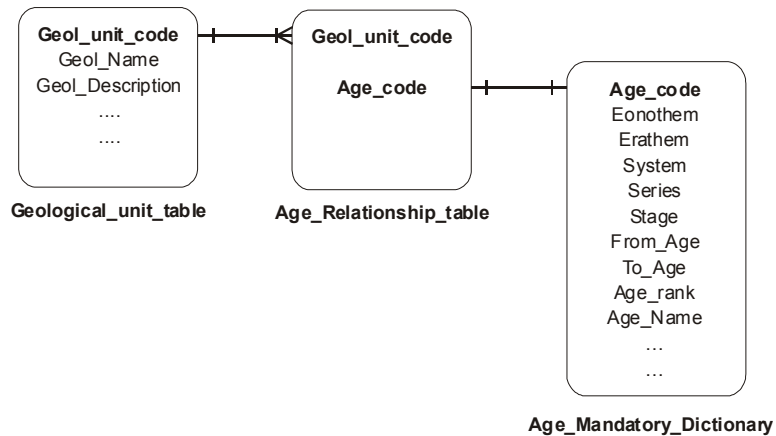


Table Relationships Diagram

### Applications

These are some examples about how this database system can be used in GIS analysis:

#### Simple Querying

Using ordinary geological database it is possible to retrieve all of the samples dated as JURASSIC and Jurassic Series and Stages. But comparing the querying statements you can appreciate how simple is retrieving sample data with the efficient chrono-stratigraphic data structure.

This is the flat database example:

```

select samples.number
  from samples, other_age_dictionary
 where
  samples.age_code = other_age_dictionary.age_code
  other_age_dictionary.age_name = 'JURASSIC' or
  other_age_dictionary.age_name = 'MALM' or
  other_age_dictionary.age_name = 'TITHONIAN' or
  other_age_dictionary.age_name = 'KIMMERIDGIAN' or
  other_age_dictionary.age_name = 'OXFORDIAN' or
  other_age_dictionary.age_name = 'DOGGER' or
  other_age_dictionary.age_name = 'CALLOVIAN' or
  other_age_dictionary.age_name = 'BATHONIAN' or
  other_age_dictionary.age_name = 'BAJONIAN' or
  other_age_dictionary.age_name = 'AALENIAN' or
  other_age_dictionary.age_name = 'TOARCIAN' or
  other_age_dictionary.age_name = 'LIAS' or
  
```

```

other_age_dictionary.age_name = 'PLIENSBACIAN' or
other_age_dictionary.age_name = 'SINEMURIAN' or
other_age_dictionary.age_name = 'HETTANGIAN'

```

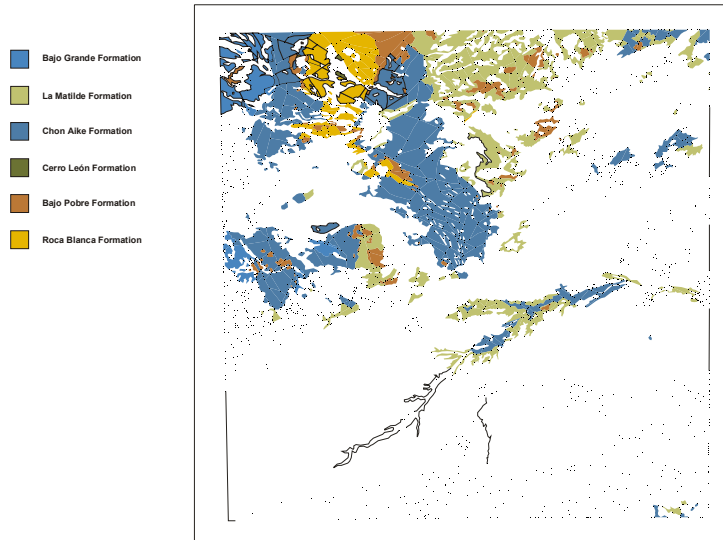
This is the example using the database explained in this work.

```

select samples.number
  from samples, age_r, age_dictionary
 where samples.number = age_r.number and age_r.age_code = age_dictionary and
       age_dictionary.system = "JURASSIC"

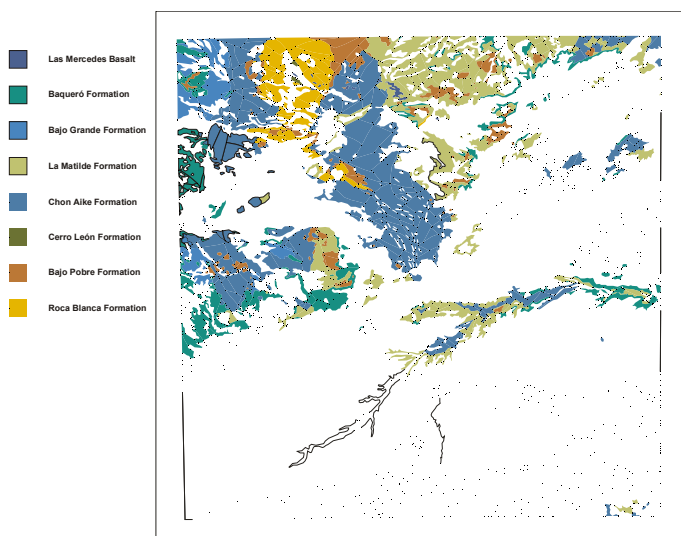
```

The following schema shows the same querying for maps



### Range Querying

Also the database design support range querying. The example shows a Jurassic bottom to Cretaceous top query that was performed using embedded SQL procedure.



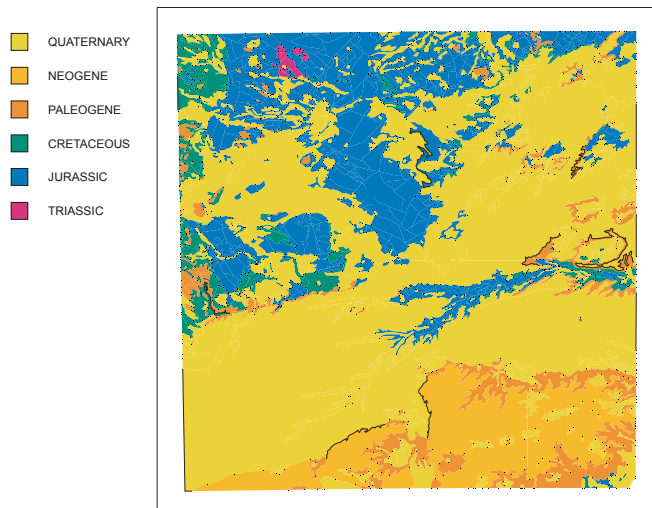
```

SELECT DISTINCT geology.id,
geology.name
FROM geology, age_rb,
ag_dictionary
WHERE
  age_rb.age_code <= (select
max(age_dictionary.age_code) from
age_dictionary where
age_dictionary.system =
'CRETACEOUS;))
  and
  age_rb.edad_code >= (select
min(age_dictionary.age_code) from
age_dictionary where
age_dictionary.system =
'JURASSIC';))
  and
  geology.id=age_rb.id
  and
  age_rb.age_code =
age_dictionary.age_code
ORDER BY geology.id DESC;

```

## Classification

Classification is an analysis method where users can make different maps, grouping object using their attributes values.



Since the proposed age management system kept parental relationships, data can be easily grouped according age hierarchies.

## Conclusion

Data retrieving methods could be enhanced by good data structures, especially if these data structures are simple y easy to understand. The database structure, presented in this paper, helps to the geologists in the querying and displaying of geological maps. This system also is very simple to implement and expand with new developments.

## References

- Asato, Carlos G., 1995. **Condiciones Técnicas y Administrativas para la Digitalización de Mapas Geológicos**. Revised by F.P. Cerdán y G. Marín. Open File. Dirección Nacional del Servicio Geológico, Buenos Aires.
- Asato, C. G., Fernando Perez Cerdán, Graciela Marín, 1996. **SIG Central del Servicio Geológico, La Importancia del Manejo Integrado de Datos Geológicos en Formato Digital**. XIII Congreso Geológico Argentino. III Congreso de Exploración de Hidrocarburos. October 13 –18 1996. Buenos Aires
- Johnson ,B.R., B. Brodaric, G.L. Raines, J.T.Hastings, and R. Wahl, 1999. **Digital Geologic Map Data Model V. 4.3** September 27, 1999.
- Burns R.K. and J. E. Glyn, 1995 **Geological Map Database Definition (Draft-3)**. Computer Technology Section, Geological Survey of Canada.
- Perez Cerdán F. 1993. **Condiciones Técnicas y Administrativas** (draft). Open File. Instituto Tecnológico Minero de España. Madrid, España.
- Remane, J. (Compiler) 1999. **International Stratigraphic Chart**. International Comission on Stratigraphy. Edited by International Union of Geological Sciences - International Comission on Stratigraphy – Commission on the Geological Map of the World, UNESCO.