Database Design Doc

Design rational:

The database is to be designed with a focus on accuracy over completeness. As such each entry is given a chain of custody in the form of having submission and approval information. Though this adds bloat to the database, it has the benefit of maintaining a full audit trail. Entries should NOT be deleted, but rather have a status set an enumerated type (pending, rejected, active, historical). Historical is for submissions that were previously approved or pending approval but have been superseded. New entries shall default to a pending status.

Each meteorite is a discrete object, about which all information is related. As such, the Body is to maintain the nomenclature. Every element attributed to the body will reference it by body\_id. This necessitates that changes be propagated in a chain. For example, if body\_id 004 originally refers to “steevee” but the name is then changed to “steve” with an id of 009, this shall be a NEW insertion, and all active and pending attributes linked to steevee must have the body\_id updated as NEW insertions. Below is a simplified example.

Body

|  |  |  |  |
| --- | --- | --- | --- |
| Body\_id | NOMENCLATURE | STATUS | PREVIOUS ENTRY |
| 004 | steevee | ~~Active~~ historical | null |
| 009 | steve | active | 004 |

Element

|  |  |  |  |
| --- | --- | --- | --- |
| ELEM\_ID | BODY\_ID | STATUS | PREVIOUS ENTRY |
| 101 | 004 | ~~Active~~ historical | null |
| 102 | 009 | active | 101 |

If, for example, only the element information needs to change, then it is not correct to propagate a change to the body table.

Body

|  |  |  |  |
| --- | --- | --- | --- |
| Body\_id | NOMENCLATURE | STATUS | PREVIOUS ENTRY |
| 004 | steevee | active | null |
| ... |  |  |  |

Element

|  |  |  |  |
| --- | --- | --- | --- |
| ELEM\_ID | BODY\_ID | STATUS | PREVIOUS ENTRY |
| 101 | 004 | ~~Active~~ historical | null |
| 102 | 009 | active | 101 |

For the elements, it is indexed with a serial, but uses body\_id FK, element symbol, paper\_id FK, and page\_num as a composite key. The analysis technique is now part of the elements table. The elements are to be ALL recorded as ppb as a standard measurement. The reasons are two-fold, it allows all elements to be stored in a single table which stores the numbers as 4-bit integers, and it ensures that there is no unit confusion. The element symbol is a varchar(3) as element symbols are always 1-3 letters.

There are several columns which are citext, this allows for efficient case-insensitive searches and guarantees uniqueness for attr such as email address (john1@mail.mail == John1@Mail.mAil).