# Rules: Project Websites Database

## Introduction

There is a fair amount of looseness in the “concrete” side of the project websites schema. It is meant to be flexible and not to impose rules on statics of interaction, so that a diverse array of projects may be represented with it. However, this implies that rules must be imposed on the dynamics of database population, and through db-client software, rather than database restrictions, themselves.

This document is to enumerate some of those rules. Some of them are imposed on the data designer, and others on the

## Rules

### Accession Naming

When creating an accession that is tied to a specific event, name it “event name”-plus the literal “Accession”, as in “Genbank Submission Accession”. This is necessary because otherwise there is no way to know which accession is principal to an event, and because the simple term “Accession” may not be overloaded (due to other rules).

### Lookup Values

There is a “namespace” of sorts, embodied in the Lookup Value table. Lookup Value provides a name, a type, and a data type. The types include (among others) Project Attribute, Sample Attribute, Event Attribute, and Event Type. The name field has a unique index, and there is a unique index on the name+type+id. This has the effect of disallowing the overloading of a name for a different type of lookup value. That is, one cannot make a project attribute and a sample attribute with the same name.

Further, the linking of data type and lookup value type enforce strong typing of lookup values, and the objects that reference them. If one has a lookup value called X with a type of Event Attribute, and a data type of “string”, one may not then create an Event Attribute against a specific project with a data type of “integer”.

### Attributes

When one creates an attribute, its existence must be allowed-for by a meta attribute. The scope of this “allowance” varies by attribute type. For project attributes they are scoped across project. You can have an attribute named X on project P if there is a meta attribute referring to project P’s key, and the lookup value for X. Meta attributes provide the link between attributes and lookup values. This implies that no knowledge of meta attributes is needed in order to read meta attributes (just lookup values).

The scope for sample attributes is over project and sample, so a sample meta attribute will have a reference to a sample, a project, and a name lookup value (of type sample attribute). Once sample meta attributes have been setup, sample attributes may be set on a sample (that is in turn associated with a project). Only after project meta attributes have been established, may project attributes (the concrete values) be set versus the project in question.

The scope for event attributes is over project and event type, so an event meta attributes will have a reference to a project, a name lookup value (of type Event Attribute, for the attribute’s name), and an event type lookup value (for the type of event, itself). Setting up these event meta attributes allows subsequent posting of the entire set of event attributes to an event. Of special note, here: when events are created, they may “contain” (that is, the implementing data may include) all different types of concrete attributes. New values for project attributes and sample attributes may be given. However, based on the event type, only event attributes registered for that type may be provided at that time. Can two different event types (not just temporally-separated events), then have event meta attributes describing the same event attribute? The answer is yes. However, there is a three-part key (project, attribute name, event type) which prohibits having multiple event meta attributes of the same name on the same event type.

Having meta attributes with distinct type (lookup values) makes it possible for users to provide attribute settings by name, without having to worry about where they are bound. The binding by lookup\_value.type lets the code determine at runtime where to place the output.

The Event Attribute table has no reference to a project, but does refer to an event. Event rows have references to projects, but only optional samples. Therefore, sample inclusion is not enforced at the database level.

Since it is permitted by the schema for the same Event Meta Attribute to be associated with multiple different Events, care must be taken by users to avoid having multiple event types have the same Event Meta Attribute, when this could cause confusion. For instance, at setup-time for events (when event meta attributes are applied to events), it is not prohibited to have an “Data Thingy” meta attribute that is associated with a Project Registration Event, as well as a Sample Registration Event, as well as a Genbank Submission event. Were this the case, “Data Thingy” could appear in multiple columns with different values, before the user. So, it is important for the user doing setup to avoid this manually.

### Ranged Attributes

For certain ranged attributes, such as accessions for Genbank, which can span a range of values (ex: ACD000010 through ACD000015), rather than using two attributes such as “start” and “end”, we should use a colon as a range delimiter. Specify this as:

Genbank Accession Other Header Other Header II

ACD000010:ACD000015 other value other value II