Crystallography Service Sample Database Administrator's Guide

J.P.Hagon
Computer Systems Support
School of Chemistry

March 11, 2012

Introduction

This guide is intended for staff who administer the Newcastle University Crystallography Service Sample Database. It includes both a general description of the web interface and associated administration procedures along with a more technical description of the software interface and the database itself so that administrators can recover from situations such as a forgotton administrator password.

1 General Description of the System

1.1 Introduction

The system consists of a *front-end* which is used by users and administrators to submit sample requests and upload analysis data. This front-end is implemented using the ruby programming language and version 3 of Ruby on Rails. The front-end is hosted on an *Apache* server running on an *Ubuntu* linux system. Technical details will be described elsewhere.

The *back-end* consists of a set of ruby programming libraries and a SQL database — in this case SQLite3. More technical aspects of the back-end will be described elsewhere.

1.2 Users

The system has a relatively simple user setup with just one basic user type. However, there are three levels of authority that a user can have:

Standard Most users of the system will have a standard account which allows them to submit sample requests and view their own sample data.

Group Leader These users have the additional privilege of being able to see all of the sample data for their own group in addition to their own samples. A group of users may have more than one designated group leader.

Administrator An administrator, in addition to standard user privileges, can do many administration tasks. These include adding/deleting users, changing user privileges, submitting/updating/deleting samples, editing public web pages on the server and uploading files to the server.

Users can either self-register or be added by an administrator. An administrator can also disable a user account without actually deleting it. Only the most basic information about a user is stored in the database, namely first name, last name and email address. the email address serves as a login id. The user can set his own password. If the user forgets his password, the system can email him a secure link to the server via which the password can be reset.

1.3 Groups

All users must be associated with a group. Typically this will be a research group associated with a particular person. When a user self-registers, he must select an appropriate group. If such a group does not exist, an administrator must set one up for him. Usually one or more users will be designated *group leaders* and will have access to information about all the group's samples.

1.4 Samples

The primary purpose of the system is to track and keep a record of samples submitted to the crystallography service. A typical workflow is shown in Figure 1. Emails are sent automatically by the system when a sample status is updated by an administrator.

1.5 Public Pages

Most information on the server can be viewed only by registered users. However, there are some pages which are more generally accessible. Such pages include the home page, general information pages and the sample queue. Public pages can be created and edited by an administrator using tools provided by the server software. rather than write pure HTML, an administrator can use a text-based markup language called Textile which can produce sophisticated web pages with all the usual constructs such as headings, paragraphs, floating elements, tables and images.

2 The Database

The core of the system is the database which holds information about users, samples etc. In this section we describe the whole database structure (or *schema* in database parlance). The easiest way to get an overall view of the database schema is to study Figure 2. This shows all the tables, fields and relationships in a single diagram. We now give a brief description of each table.

Note that all tables except join tables have an autoincremented integer field called `id` which servers as the unique primary key for each record in the table. The `id` field will not be listed explicitly in the description of each table. All non-join tables also have two other fields, `created_at` and `updated_at` in a datetime format. Again, we will not explicitly list these fields in the description of the tables which follows.

2.1 The Samples Table

The samples and users tables are the key parts of the database as is evident from Figure 2. They are related to each other via a *one-to-many* relationship, i.e. *one* user can have *many* samples. The samples table consists of the following fields:

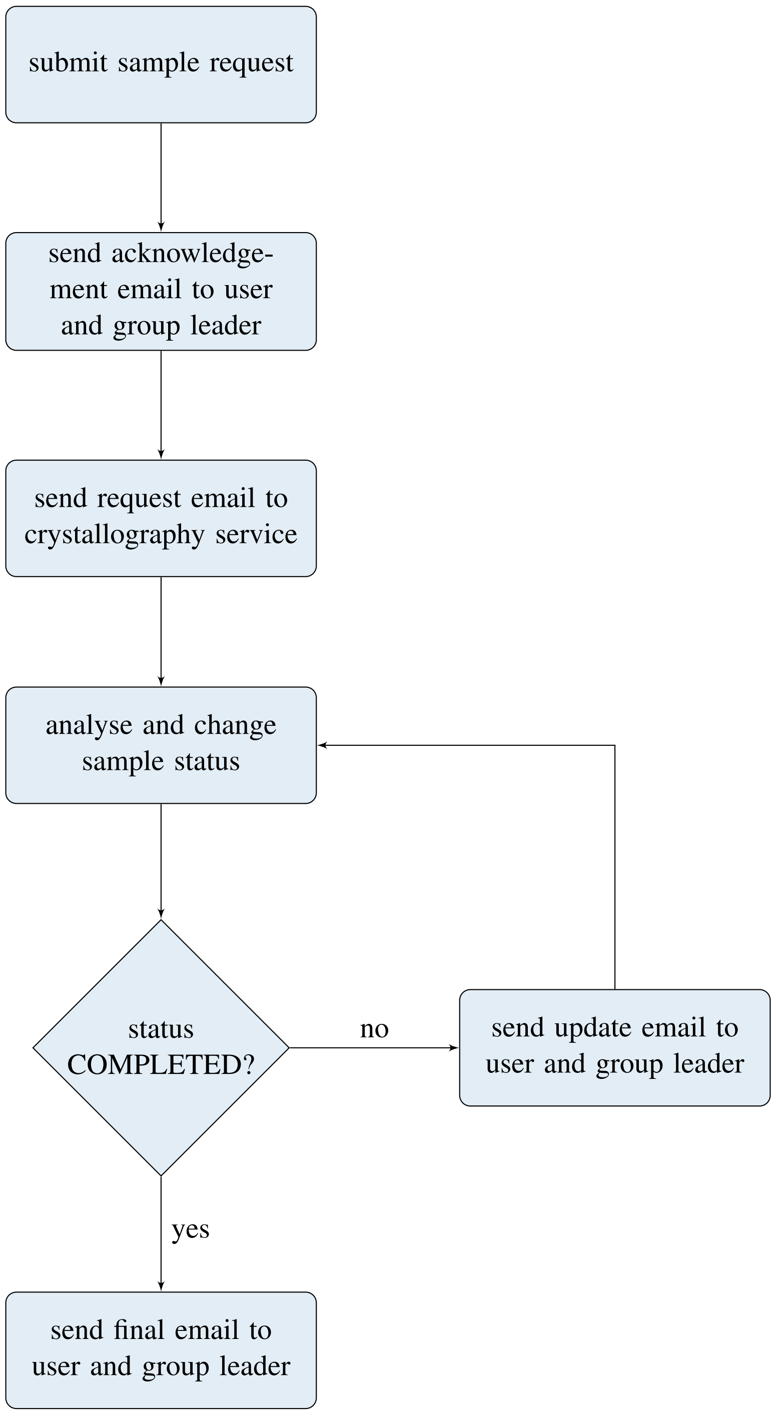
submit sample request
send acknowledge-ment email to user and group leader
send request email to crystallography service
analyse and change sample status
status COMPLETED?
no
send update email to user and group leader
yes
send final email to user and group leader

Figure 1: Typical workflow for sample processing cycle. Emails are sent automatically by the system whenever the status of a sample is updated.

code a string, automatically generated by the system having the general form `AAA-AA-YY-1111` where the `AAA` and `AA` represent 3-letter codes for group and submitter respectively; the `YY` represents the year and the `1111` represents a number which is incremented for that group but reset to zero at the start of each calendar year.

cif a string representing the chemical formula of the sample in cif format.

synth a string representing the file name of an image file specifying the details of the synthesis.

coshh_name a string representing the name of the solvent (if any).

coshh_info another string describing any procedures in case of contact with the sample.

coshh_desc a text field providing a brief description of the sample (e.g. organic amide).

params a string representing unit cell parameters or CSD/Newcastle code for possible by-products or previously obtained, unpublished results.

priority an integer between 1 and 9 to give an indication of priority.

powd a boolean parameter indicating if the sample requires powder diffraction (y/n).

chiral another boolean indicating whether the molecule is chiral (y/n).

costcode a string providing a cost centre code for charging if relevant.

barcode a string field for an automatically generated Code39 standard barcode.

user_id this integer holds the `id` field of the user who requested the sample analysis.

flag_id an integer holding the `id` field of the status flag of the sample.

userref a string for a user-defined reference. This is required to be an alphanumeric sequence of characters *without spaces*.

zipdata a string holding the name of a zipfile containing the results of the analysis.

sampleimage a string holding the name of an image file of the sample molecule after it has been identified by the analysis.

reference a text field for a published reference (typically in the form of a DOI).

comments a text field for any general comments the user wishes to make about the sample.

colour a string holding information about the colour of a sample after analysis.

size a string holding information about the size of a sample after analysis.

shape a string holding information about the shape of a sample after analysis.

2.2 The Users Table

The users table, in addition to maintaining a record of users and their samples, also serves as a key part of the authentication and authorization system which will be described later. the users table is related to the samples table via a *one-to-many* relationship, i.e. *one* user has *many* samples.

email a string holding the email address of the user. This serves also as the user login id.

encrypted_password a string holding the user's password in an encrypted form.

reset_password_token a string containing a special token used if the user has forgotten his password and needs to reset it.

reset_password_sent_at a datetime field recording the time a token enabling a user to reset his password was sent.

remember_created_at a datetime field specifying the time at which a user requested that his login id be remembered by the browser so he need not type in his credentials.

sign_in_count an integer holding the number of times a user has logged-in.

current_sign_in_at a datetime field holding the sign-in time for the current session.

last_sign_in_at a datetime field holding the last sign-in time for the user.

current_sign_in_ip a datetime field holding the user's ip address for the current session.

last_sign_in_ip a datetime field holding the previous login ip address for the user.

group_id an integer representing the `id` field of the group to which the user belongs.

admin a boolean field indicating whether the user is an administrator (y/n).

firstname a string holding the user's first name.

lastname a string holding the user's last name.

leader a boolean field indicating whether the user is a group leader (y/n).

enabled a boolean field indicating if the account is enabled (y/n).

2.3 The Stores, Hazards and Sensitivities Tables

These tables are each very similar and have the same basic structure. They are used to specify storage, hazard and sensitivity properties for a sample. They all have a *many-to-many* relationship with the samples table. This is because a sample can have, for example, *many* storage requirements, but also a single storage requirement can be associated with *many* samples. All these tables have essentially the same fields:

name a string defining a short name for the property.

description a text field describing the property at greater length.

For historical reasons, the hazards table uses the names `hazard_abbr` and `hazard_desc` for the `name` and `description` fields. Also the `hazard\_desc` field is a text field rather than a string.

Associated with these tables are three further *join tables* which facilitate the many-to-many relationship between a sample and its properties. These join tables are called `samples_stores`, `samples_hazards` and `samples_sensitivities`. They all contain two fields corresponding to the sample `id` field and the associated property `id` field. For example, `samples_stores` contains the fields `sample_id` and `store_id`. Both these fields are integers of course.

2.4 The Groups Table

This table represents groups of users, normally research groups but also perhaps external companies etc. It is a simple table, but important in the way the whole system works. It contains the following fields:

group_abbr a 3-letter string as an abbreviation for the group. Amongst other things this is used to form part of the sample code string mentioned earlier.

group_desc a string giving a more complete description of the group.

2.5 Other Tables

There are several other tables which are less important than the ones discussed so far in the sense that they are strictly not necessary for a working sample tracking system. However, they do assist in making the system much easier to manage and also help making the system much friendlier for users. These tables are the `assets`, `pages` and `popups` tables.

2.5.1 The Pages Table

The purpose of this table is to provide a means by which administrators can add 'static' content to the sample tracking web site. Each static page has its content stored in this table. The fields are:

name a string storing a name for the page. This is typically used to provide a title for the page in a web browser window.

permalink another string used to provide a short, quick URL for the page.

content a text field which contains the page content. This is expected to be written in Textile markup language (although a mixture of pure HTML and Textile can be used.

2.5.2 The Assets Table

The assets table keeps a record of general files which have been uploaded to the server. These files are typically graphical images, pdf documents etc. and will usually be referenced in one of the static pages created by administrators which are stored in the `pages` table. An 'asset' is simply one of these uploaded documents and the `assets` table keeps a record of it. The fields are:

document the full path name of an uploaded document. This path name is ultimately assigned using the carrierwave file uploading plugin to ruby on rails.

description a text field giving a brief description of the document.

2.5.3 The Popups Table

This table stores descriptive information about the primary fields in the samples table. It has two fields:

name this string should have the same name as one of the sample fields for which a detailed description is required.

description a text field giving a detailed description of the associated sample field in the corresponding `name` field.

The `popups` table, as its name implies, provides descriptive text in popup boxes whenever a user hovers the mouse over the appropriate field in the sample submission form.

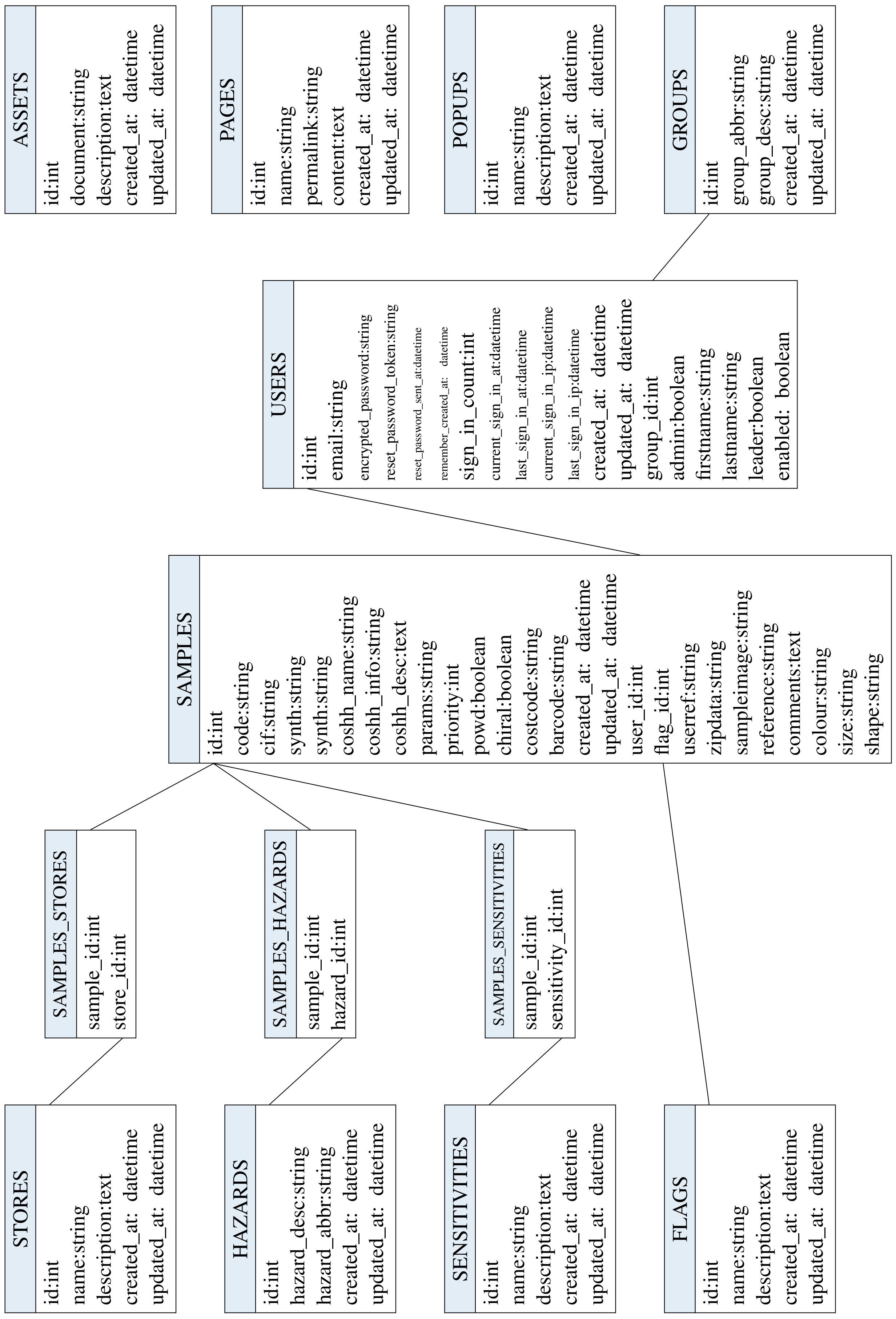

Figure 2: The overall database schema. Relationships between tables are indicated with lines joining the relevant fields. Note that the tables `SAMPLES_STORES`, `SAMPLES_HAZARDS` and `SAMPLES_SENSITIVITIES` are *join tables* which serve only to facilitate a many-to-many relationship between the tables they link.

Contents