

Introduction

The Audit module is not a module in the same sense as the other Chado modules. It is a set of recommendations on how one might implement auditing in Chado. This documentation describes an example of such an implementation which is in use at [FlyBase](#). The SQL used at Flybase is `chado/modules/audit/audit_tables.sql`.

Audit Policy

We consider the implementation of auditing policy as a database administration issue and subject to the local requirements of specific database groups. We therefore have not developed a detailed audit module, providing instead this policy statement.

Database project groups who would use chado as their primary DBMS would be well advised to implement as robust and detailed an audit trail as possible, while groups who would use chado more informally, for instance primarily as a read-only resource, may have no need for auditing.

We here describe how auditing will be implemented at FlyBase, where chado will serve as the primary DBMS.

Auditing will be implemented in a separate, orthogonal database to the chado, with each chado table having a corresponding shadow audit table which will include all of the columns of the chado table, plus these additional columns:

```
transaction_timestamp NOT NULL
transaction_char NOT NULL
```

Where *transaction_type* value is one of:

- i (insertion)
- u (update)
- d (deletion)

Thus an chado db instance `chado.1` containing a table *gene*, for example:

```
CREATE TABLE gene (
    gene_id INT NOT NULL,
    PRIMARY KEY (gene_id),
    VARCHAR(255) NOT NULL,
    dbxref_id
FOREIGN KEY (dbxref_id) REFERENCES dbxref(dbxref_id),
UNIQUE (name),
UNIQUE (dbxref_id)
);
```

Would have a corresponding audit database, `chado.1_audit`, containing a shadow audit table, *gene_audit*:

Chado_Audit_Module

```
CREATE TABLE gene_audit (
  gene_id INTEGER NOT NULL,
  PRIMARY KEY (gene_id),
  name CHAR(255) NOT NULL,
  dbxref_id
  FOREIGN KEY (dbxref_id) REFERENCES dbxref(dbxref_id),
  UNIQUE (name),
  UNIQUE (dbxref_id),
  transaction_date TIMESTAMP NOT NULL,
  transaction_type CHAR NOT NULL
);
```

Transaction Use Cases

Insertion of a New Record

When a record is initially inserted into an chado table, the same record is inserted into the corresponding shadow audit table, with the current timestamp and *transaction_type* = 'i'.

Update of an Existing Record

When a record is to be updated, before the update transaction is executed in chado, the existing chado record is copied to the corresponding shadow audit table, with the current timestamp and *transaction_type* = 'u'. The update transaction is then executed in chado.

Deletion of an Existing Record

When a record is to be deleted, before the delete transaction is executed in chado, the existing chado record is copied to the corresponding shadow audit table, with the current timestamp and *transaction_type* = 'd'. The delete transaction is then executed in chado.

Implementation Issues

The benefit of this system is that it allows a certain amount of "time travel", or the ability to reconstruct the database to its state at a particular time in the past. We don't anticipate making this functionality available to the general user population, but recognize that from a database programming point of view, it is often quite useful, and sometimes vital, for a programmer to be able to track changes in the database in detail.

While the auditing policy described here fulfills all of our auditing requirements in principle, it remains to be seen how it will work in practice. The size of the shadow audit database may become an issue over time. How the handling of audit transactions will affect overall database transaction performance is also unknown. FlyBase makes a practice of reporting certain audit details to the public (e.g. date entered and date last updated of gene records) - it is not known whether implementing this will be too expensive using this shadow audit db method

Listing of MODs' Implemented Audit Tracking Systems

A listing of instances of audit tracking systems implemented by different MODs is provided [here](#).