


SQL Source

 This plugin is experimental, and interfaces may change across minor releases.

- [Modes](#)
- [Installation](#)
- [Configuration](#)
 - [Group Mappings](#)
- [Preparing Source Data](#)
 - [Modified Timestamps](#)
 - [Flat Mode](#)
 - [Relational Mode](#)
 - [Archive Tables](#)
 - [Bootstrapping Archive Tables](#)
- [Debugging](#)
 - [Table Not Found In Datasource](#)
 - [Archive Tables Appear To Be Ignored](#)
- [See Also](#)

The SQL Organizational Identity Source Plugin is designed to integrate inbound data via a SQL database. SqlSource is available as of Registry v4.1.0.

Modes



Org Identity Source Mode	Support
Manual Search and Linking	Supported
Enrollment, Authenticated	Not supported
Enrollment, Claim	Not supported
Enrollment, Search	Supported
Enrollment, Select	Supported

Org Identity Sync Mode	Support
Full	Supported
Query	Supported
Update	Supported
Manual	Supported

Installation

This is a non-core plugin, see [Installing and Enabling Registry Plugins](#) for more information.

Configuration

1. Add a new Server (CO > Servers >  *Add a New Server*).
 - a. Set the Server Type to **SQL**.
 - b. Set the appropriate connection information. Note that SqlSource only supports Postgres and MySQL/MariaDB databases.
2. Add a new Organizational Identity Source (CO > Configuration > Organizational Identity Sources >  *Add Organizational Identity Source*).
 - a. Set the Plugin to *SqlSource*, and other [configurations](#) as appropriate.
3. On the plugin configuration page
 - a. Server: Select the Server created above
 - b. Table Mode: Select the appropriate table mode, see *Preparing Source Data* (below) for more information
 - c. Source Table: Enter the exact name of the source table (or primary source table for Relational Mode)
 - d. For Flat Mode, set the desired types for the various related models such as *Name* and *Email Address*.

Group Mappings

[Ad Hoc Attribute](#) tags may be used to configure [Group Mappings](#).



In *Relational Tables* mode, the available set of Ad Hoc Attributes is calculated based on the actual data in the inbound table (described below). As such, no attributes will be available for Group Mapping until data is populated into the tables.

Preparing Source Data

Inbound data may be represented using any supported database technique, including physical tables, foreign tables, views, and materialized views, so long as the representation meets the requirements described in this documentation.

SqlSource supports two modes for importing data.

- In *Flat Mode*, there is a single inbound table with one row per inbound Organizational Identity, with at most one value supported for related models such as *Name* and *Email Address*.
- In *Relational Mode*, there is a primary table with one row per inbound Organizational Identity, and a set of related tables allowing for multiple values for associated models.

Flat Mode is simpler to set up, but Relational Mode allows for more complex records. There is a small performance penalty for Relational Mode (since multiple tables or views need to be queried), but this should be negligible under most circumstances.

Modified Timestamps

When processing a [full sync](#), SqlSource will attempt to pull all records from the Source Table. For larger data sets, it is recommended instead to add a `modified` column to the Source Table indicating when the record was last changed. If this column is present, SqlSource will use it instead to obtain a list of updated records. In Relational Mode, only the primary table supports the `modified` column. If present, values should be provided for all rows.

As of Registry v4.3.0, an additional option using archive tables is available, see below for more information.

Flat Mode

Flat Mode expects a single Source Table whose definition is consistent with the following (the name may be any valid SQL name):

```
CREATE TABLE my_source_table (  
    id integer PRIMARY KEY,  
    sorid character varying(1024) UNIQUE,  
    honorific character varying(32),  
    given character varying(128) NOT NULL,  
    middle character varying(128),  
    family character varying(128),  
    suffix character varying(32),  
    affiliation character varying(32),  
    date_of_birth date,  
    valid_from timestamp without time zone,  
    valid_through timestamp without time zone,  
    title character varying(128),  
    o character varying(128),  
    ou character varying(128),  
    manager_identifier character varying(512),  
    sponsor_identifier character varying(512),  
    mail character varying(256),  
    identifier character varying(512),  
    telephone_number character varying(64),  
    address character varying(512),  
    url character varying(256),  
    -- if the modified column is present, it will be used to detect records changed since the last full sync  
    modified timestamp without time zone  
);
```

where

- `id`: A unique integer that serves as a Primary Key. This column is not currently used by SqlSource but may be used in a future release.
- `sorid`: The System of Record Identifier, and must be unique within the table.
- `honorific`, `given`, `middle`, `family`, and `suffix`: Used to construct the (Primary) [Name](#), which will be assigned the type as configured above. Which elements are actually required is governed by CO Settings *Name Required Fields* (and *Name Permitted Fields*).
- `affiliation`, `date_of_birth`, `valid_from`, `valid_through`, `title`, `o`, `ou`, `manager_identifier`, `sponsor_identifier`: Used to construct the [Org Identity](#).
- `mail`: Used to construct the [Email Address](#), which will be assigned the type as configured above.

- **identifier**: Used to construct the [Identifier](#), which will be assigned the type as configured above.
- **telephone_number**: Used to construct the [Telephone Number](#), which will be assigned the type as configured above. Only the `number` field will be populated.
- **address**: A comma separated set of fields used to construct the [Address](#), which will be assigned the type as configured above. The value consists of the concatenated fields `street,locality,state,postal_code`.
- **url**: Used to construct the set of fields used to construct the [Url](#), which will be assigned the type as configured above.
- **modified**: If this column is defined, the time this record was last updated (GMT).

Any additional columns will be treated as AdHoc Attributes.

For search and retrieval of larger data sets, indexes should be maintained on the following fields:

- `sorid`
- `given` (lowercase index)
- `family` (lowercase index)
- `mail`
- `identifier`
- `modified` (if the column is defined)

Relational Mode

Relational Mode expects a primary Source Table (whose name is exactly as configured in the plugin configuration) and a series of one or more related Source Tables (whose name is the Source Table name, an underscore (`_`), and the fixed string describing the related table. Only the `_names` table is required, all others will be silently skipped if not present. The tables must be consistent with these SQL definitions:

```
CREATE TABLE my_source_table (
    id integer PRIMARY KEY,
    sorid character varying(1024) UNIQUE,
    affiliation character varying(32),
    date_of_birth date,
    valid_from timestamp without time zone,
    valid_through timestamp without time zone,
    title character varying(128),
    o character varying(128),
    ou character varying(128),
    manager_identifier character varying(512),
    sponsor_identifier character varying(512),
    -- if the modified column is present, it will be used to detect records changed since the last full
sync    modified timestamp without time zone
);

CREATE TABLE my_source_table_names (
    id integer PRIMARY KEY,
    sorid character varying(1024) REFERENCES my_source_table (sorid),
    honorific character varying(32),
    given character varying(128) NOT NULL,
    middle character varying(128),
    family character varying(128),
    suffix character varying(32),
    type character varying(32) NOT NULL,
    language character varying(16),
    primary_name boolean
);

CREATE TABLE my_source_table_addresses (
    id integer PRIMARY KEY,
    sorid character varying(1024) REFERENCES my_source_table (sorid),
    street text NOT NULL,
    room character varying(64),
    locality character varying(128),
    state character varying(128),
    postal_code character varying(16),
    country character varying(128),
    type character varying(32) NOT NULL,
    language character varying(16)
);

CREATE TABLE my_source_table_email_addresses (
    id integer PRIMARY KEY,
    sorid character varying(1024) REFERENCES my_source_table (sorid),
    mail character varying(256) NOT NULL,
    type character varying(32) NOT NULL,
```

```

        verified                boolean
    );

CREATE TABLE my_source_table_identifiers (
    id                          integer PRIMARY KEY,
    sorid                       character varying(1024) REFERENCES my_source_table (sorid),
    identifier                   character varying(512) NOT NULL,
    type                        character varying(32) NOT NULL,
    login                       boolean
);

CREATE TABLE my_source_table_telephone_numbers (
    id                          integer PRIMARY KEY,
    sorid                       character varying(1024) REFERENCES my_source_table (sorid),
    country_code                character varying(3),
    area_code                   character varying(8),
    number                      character varying(64) NOT NULL,
    extension                   character varying(16),
    type                        character varying(32) NOT NULL
);

CREATE TABLE my_source_table_urls (
    id                          integer PRIMARY KEY,
    sorid                       character varying(1024) REFERENCES my_source_table (sorid),
    url                         character varying(256) NOT NULL,
    type                        character varying(32) NOT NULL
);

CREATE TABLE my_source_table_ad_hoc_attributes (
    id                          integer PRIMARY KEY,
    sorid                       character varying(1024) REFERENCES my_source_table (sorid),
    tag                         character varying(128) NOT NULL,
    value                       character varying(256)
);

```

where

- **id**: A unique integer that serves as a Primary Key. This column is not currently used by SqlSource but may be used in a future release.
- **sorid**: The System of Record Identifier, and must be unique within the primary table only. For related tables the sorid is effectively a foreign key to the primary table.
- **affiliation, date_of_birth, valid_from, valid_through, title, o, ou, manager_identifier, sponsor_identifier**: Used to construct the [Org Identity](#).
- **modified**: If this column is defined, the time this record was last updated (GMT). Only applies to the primary table.
- **honorific, given, middle, family, suffix, type, language, and primary_name**: Used to construct a [Name](#), at least one of which must be provided and exactly one of which must be flagged `primary_name`. Which elements are actually required is governed by CO Settings *Name Required Fields* (and *Name Permitted Fields*).
- **street, room, locality, state, postal_code, country, type, and language**: Used to construct an [Address](#).
- **mail, type, and verified**: Used to construct an [Email Address](#). If verified is not provided, the Email Address will be considered unverified.
- **identifier, type, and login**: Used to construct an [Identifier](#). If login is not provided, the Identifier will not be permitted for login.
- **country_code, area_code, number, extension, and type**: Used to construct a [Telephone Number](#). Only the number field will be populated.
- **url and type**: Used to construct the set of fields used to construct a [Url](#).
- **tag and value**: Used to construct the set of fields used to construct an [Ad Hoc Attribute](#).

For search and retrieval of larger data sets, indexes should be maintained on the following fields:

- sorid (for all tables)
- given (lowercase index)
- family (lowercase index)
- mail
- identifier
- tag
- modified (if the column is defined)

Archive Tables

Registry v4.3.0 adds an additional option for more efficiently determining changed records, using database level diff calculations. It is possible to create *Archive Tables*, which are used by SQL Source to track which records have already been processed. Archive Tables are only supported when the corresponding Org Identity Source is operated in [Full or Update Mode](#).

The specific requirements of Archive Tables may be governed by the database server in use, but in general the Archive Tables must have an identical (or at least "union compatible") definition as the main tables. SQL Source expects the Archive Tables to be named with an `_archive` suffix. So, using the Flat Mode example above, the table name would be `my_source_table_archive`, and it would have the same construction as `my_source_table`.

Archive Tables are also supported in Relational Mode, each table must have a corresponding archive version (eg: `my_source_table_names_archive`, etc).

Archive Tables will automatically be used if detected, *unless* the Source Table (or primary Source Table for Relational Mode) has a `modified` column, in which case the modified timestamp approach will be used instead.

Because of the structure of the current implementation of the Org Identity Source sync infrastructure, the Archive Tables are only updated after a successful Sync run. If a Sync run fails to complete for some reason, there may be some records that were actually processed but are not reflected in the Archive Table. These records will be reprocessed at the next run, but should not result in any changes, and the Archive Table will eventually update. Similarly, manually (re)processing a single record will not update the Archive Table, but again at the next Sync run they will be reprocessed (resulting in no changes) and the Archive Table eventually updated.



Archive Table functionality is implemented using `EXCEPT` statements, which require full table scans in order to perform comparisons. As such, use of Modified Timestamps as described above are likely to be faster in many circumstances, and should generally be considered preferred when available.

For more information on `EXCEPT`, see the database specific documentation:

- [Postgres Set Operations](#)
- [MySQL EXCEPT](#) (supported since v8.0.31)
- [MariaDB EXCEPT](#) (supported since v10.3)

Bootstrapping Archive Tables

When Archive Tables are added to an already operational instance of `SqlSource`, the tables must be bootstrapped with the current records from the Source Tables. The easiest way to do this is to ensure that no changes are being made to the Source Tables (and that the current values in the Source Tables have been processed) and then creating the Archive Tables from the Source Tables, eg:

```
SQL> CREATE TABLE my_source_table_archive as SELECT * FROM my_source_table;
```

Make sure the ownership of the newly created tables is correct.

For a new `SqlSource` instance that has not yet had any records processed and that will be run in Full mode, bootstrapping is not required.

Debugging

Table Not Found In Datasource

If the specified table is not in fact in the database, the error message "Table `my_source_table` for model `SourceRecord` was not found in datasource `sourcedb`." probably means the ownership or permissions on the Source Tables are incorrect, and Registry is unable to query them. (The actual database issued permission error is masked by Cake, but may be visible in the database error logs.)

Archive Tables Appear To Be Ignored

Check that

1. The Source Table has a `modified` column (in which case that is used instead)
2. The ownership or permissions on the Archive Tables is correct (otherwise Registry is unable to query or update them)
3. If there is already data in the Source Tables that has been processed, that the Archive Tables have been bootstrapped.

Note that manually running individual records (eg: as a test) will *not* populate the Archive Tables.

See Also

- [cm_sql_sources](#)
- [SQL Provisioning Plugin](#)