

# SQL Provisioning Plugin

The SQL Provisioning Plugin provisions CO Person and CO Group records to a SQL database.

- [Operations](#)
  - [Reference Data](#)
- [Installation](#)
- [Configuration](#)
- [Target Database Schema](#)
  - [Table Name Prefix](#)
  - [Updating the Target Database Schema and Reference Data](#)
  - [Building Views for Application Integration](#)
- [Database Permissions](#)
- [See Also](#)

## Operations

Registry CO Person Transaction	SQL Action
Add	Synchronize the CO Person and their CO Group Memberships
Edit	Synchronize the CO Person and their CO Group Memberships
Enter Grace Period	No changes (unless attributes change as part of grace period)
Expiration / Becomes Inactive	Synchronize the CO Person and their CO Group Memberships
Unexpire / Becomes Active	Synchronize the CO Person and their CO Group Memberships
Delete	Remove the CO Person
Manual Provision	Synchronize the CO Person and their CO Group Memberships

Registry CO Group Transaction	Changelog Action
Add	Synchronize the CO Group and its Memberships
Edit	Synchronize the CO Group and its Memberships
Delete	Remove the CO Group
Manual Provision	Synchronize the CO Group and its Memberships

## Reference Data

In addition to the [usual](#) CO Person and CO Group data, the SQL Provisioning Plugin will write *Reference Data* to the target database. Reference Data can be used to understand foreign keys in the provisioned operational data. Reference Data is fully provisioned when the plugin is initially configured, and then updated whenever there are changes to the Reference Data.

The following models are provisioned as Reference Data:

1. COU
2. CoTermsAndConditions
3. OrgIdentitySource (as of Registry v4.0.0)

## Installation

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

## Configuration

1. Create a new, empty database where the Provisioner will write. This database is the *Target Database*.
2. Define a new Server in Registry.
  - a. *Servers > Add a New Server*
  - b. ServerType: SQL
  - c. On the next page, enter the configuration information for the Target Database.
3. Configure a new Provisioning Target in Registry.
  - a. *Configuration > Provisioning Targets > Add Provisioning Target*
    - i. Plugin: SqlProvisioner

- ii. On the next page, select the Server created in the previous step.
- iii. As of Registry v4.2.0, the table prefix may also be specified. See *Target Database Schema*, below, for more information.



The Target Database must be available when the configuration is saved. The Provisioner will instantiate the target database schema when the configuration is saved, and also populate the initial reference data.



The Provisioner will write one COs worth of records to the Target Database. Multiple COs should not share the same Target Database.

## Target Database Schema

The target database schema is a subset of the [regular database schema](#), and only reflects records that are [ordinarily eligible for provisioning](#). Table names are prefixed with `sp_` to distinguish them from operational tables, and also to avoid conflicting with existing code (technical: to avoid auto-binding to existing models).

Table	Type	Derived From	Available Since
<a href="#">sp_ad_hoc_attributes</a>	Operational	<a href="#">cm_ad_hoc_attributes</a>	v3.3.0
<a href="#">sp_addresses</a>	Operational	<a href="#">cm_addresses</a>	v3.3.0
<a href="#">sp_co_groups</a>	Operational	<a href="#">cm_co_groups</a>	v3.3.0
<a href="#">sp_co_group_members</a>	Operational	<a href="#">cm_co_group_members</a>	v3.3.0
<a href="#">sp_co_people</a>	Operational	<a href="#">cm_co_people</a>	v3.3.0
<a href="#">sp_co_person_roles</a>	Operational	<a href="#">cm_co_person_roles</a>	v3.3.0
<a href="#">sp_co_t_and_c_agreements</a>	Operational	<a href="#">cm_co_t_and_c_agreements</a>	v3.3.0
<a href="#">sp_email_addresses</a>	Operational	<a href="#">cm_email_addresses</a>	v3.3.0
<a href="#">sp_identifiers</a>	Operational	<a href="#">cm_identifiers</a>	v3.3.0
<a href="#">sp_names</a>	Operational	<a href="#">cm_names</a>	v3.3.0
<a href="#">sp_telephone_numbers</a>	Operational	<a href="#">cm_telephone_numbers</a>	v3.3.0
<a href="#">sp_urls</a>	Operational	<a href="#">cm_urls</a>	v3.3.0
<a href="#">sp_cous</a>	Reference	<a href="#">cm_cous</a>	v3.3.0
<a href="#">sp_co_terms_and_conditions</a>	Reference	<a href="#">cm_co_terms_and_conditions</a>	v3.3.0
<a href="#">sp_org_identity_sources</a>	Reference	<a href="#">cm_org_identity_sources</a>	v4.0.0



Currently, all records associated with a CO Person are provisioned, including those that are created via [Organizational Identity Sources](#) and [Registry Pipelines](#). This may result in what appear to be multiple copies of the same record (eg: multiple Names for the same person) in the Target Database. Viewing the CO Person record will also show the multiple copies, albeit with additional metadata to explain their origins.

As of Registry v4.0.0, relevant models have a foreign key `org_identity_source_id` that points to the [sp\\_org\\_identity\\_sources](#) Reference Table, which allows for determining the provenance of a given attribute. (If the value is NULL, the attribute is a direct Registry value.) The Operational and Reference Tables can be joined together, with SQL such as

```
SELECT sp_email_addresses.*, sp_org_identity_sources.description
FROM sp_email_addresses
INNER JOIN sp_org_identity_sources ON sp_email_addresses.org_identity_source_id=sp_org_identity_sources.id
WHERE sp_email_addresses.co_person_id=2650;
```



The Target Database Schema will not automatically be removed if the plugin configuration is deleted. Drop the schema manually if it is no longer required.

## Table Name Prefix

As of Registry v4.2.0, the table name prefix (`sp_` in the above documentation) may be configured. The prefix must use valid SQL characters (ie: alphanumerics and underscores), the use of a trailing underscore (`_`) is not required but is recommended.



The prefix should not be omitted entirely, as this may cause conflicts in the underlying framework code.

## Updating the Target Database Schema and Reference Data

The Target Database Schema and Reference Data are automatically updated whenever the SqlProvisioner configuration is saved. As of Registry v4.0.0, it is also possible to manually perform either operation by viewing the plugin configuration and clicking the appropriate button.

Resyncing Reference Data will also resync all CO Groups, since these are effectively Reference Data as well.

## Building Views for Application Integration

Some view definitions that may be helpful for application integration are available [here](#).

## Database Permissions

SqlProvisioner requires permissions to create and alter tables, and manage all rows within those tables. It is recommended that each SqlProvisioner instance operate within its target schema.

### Sample Postgres GRANT

```
SQL> GRANT CREATE, SELECT, INSERT, UPDATE, DELETE, TRUNCATE ON ALL TABLES IN SCHEMA reporting_database TO  
user_role;
```

## See Also

- [cm\\_co\\_sql\\_provisioner\\_targets](#)