

Database Schema Definition

- [Column Library](#)
- [Indexes](#)
- [ChangelogBehavior](#)
- [Multi-Valued Entity Attributes](#)
- [Organizational Identity Sources](#)

Prior to Registry v5.0.0, the database schema was defined using the [ADOdb xmlschema](#) format.

Starting with Registry v5.0.0, schema management is handled via Doctrine's [Database Abstraction Layer \(DBAL\)](#). DBAL does not have a schema file format, so Registry defines one (that is also used by Match) that closely aligns with DBAL's internal format. Processing is handled by `DatabaseCommand`.

The schema file is found in `config/schema/schema.json`. The format is not exhaustively defined here, since it's pretty straightforward to look at existing tables to see how things work. However, there are several shortcuts available to reduce copy/paste and redundancy, and those are highlighted here:

Column Library

The column library pre-defines common columns (such as `co_id`). The library values become defaults that are merged into the actual column definition, so portions (or the whole definition) can be overridden. For example, the column library defines `co_id` as not null. If a table needs to be defined where `co_id` is optional, the following could be used in the table column definition:

```
"co_id": { "notnull": false }
```

Indexes

By default, DBAL will create indexes for all foreign keys, and will create random identifiers for them. To facilitate index reconstruction (which can be very slow for large datasets), define indexes using names of the form `table_i1`. Define indexes for foreign keys even if they aren't necessary, to avoid DBAL random identifier names.

`DatabaseCommand` will automatically generate certain indexes, as described below (for `ChangelogBehavior`, `Multi-Valued Entity Attributes`, and `Organizational Identity Sources`).

ChangelogBehavior

Columns and indexes for `ChangelogBehavior` are automatically inserted for all tables by default. This can be disabled by adding the following to the table definition:

```
"changelog": false
```

Multi-Valued Entity Attributes

`DatabaseCommand` will automatically generate columns and indexes for Multi-Valued Entity Attributes (MVEAs, previously known as Multi-Valued Person Attributes or MVPAs) if configured with the list of parent tables in the table definition:

```
"mvea": [ "co_person", "org_identity" ]
```

Organizational Identity Sources

Tables associated with Organizational Identity Source Records track their provenance via *source* foreign keys. `DatabaseCommand` will automatically generate columns and indexes for these foreign keys if configured with the following in the table definition:

```
"sourced": true
```