

Documentation for the Mapping Medieval Conflict (MEDCON) database

Author: Alexander Watzinger (alexander.watzinger@oeaw.ac.at), Seta Štuhec

License: CC-BY 4.0

Last update: 2020-03-22

Project:

MEDCON <https://oeaw.academia.edu/MappingMedievalConflict>

Software used for data entry:

OpenAtlas - <https://openatlas.eu>, Dokumentation - <https://redmine.openatlas.eu/projects/uni/wiki>

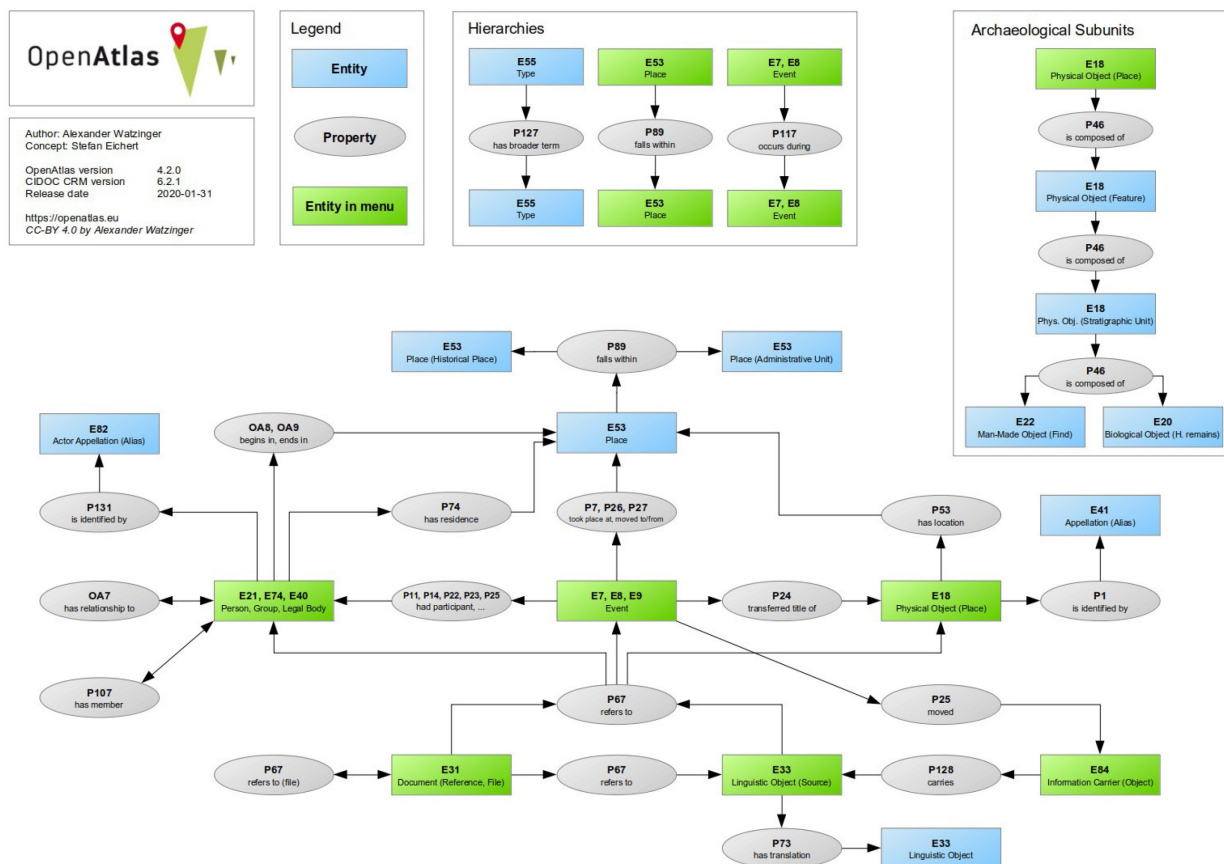
This file contains a dump of the PostgreSQL database containing two schemas. It should be possible to restore the database with this file.

1. Model schema

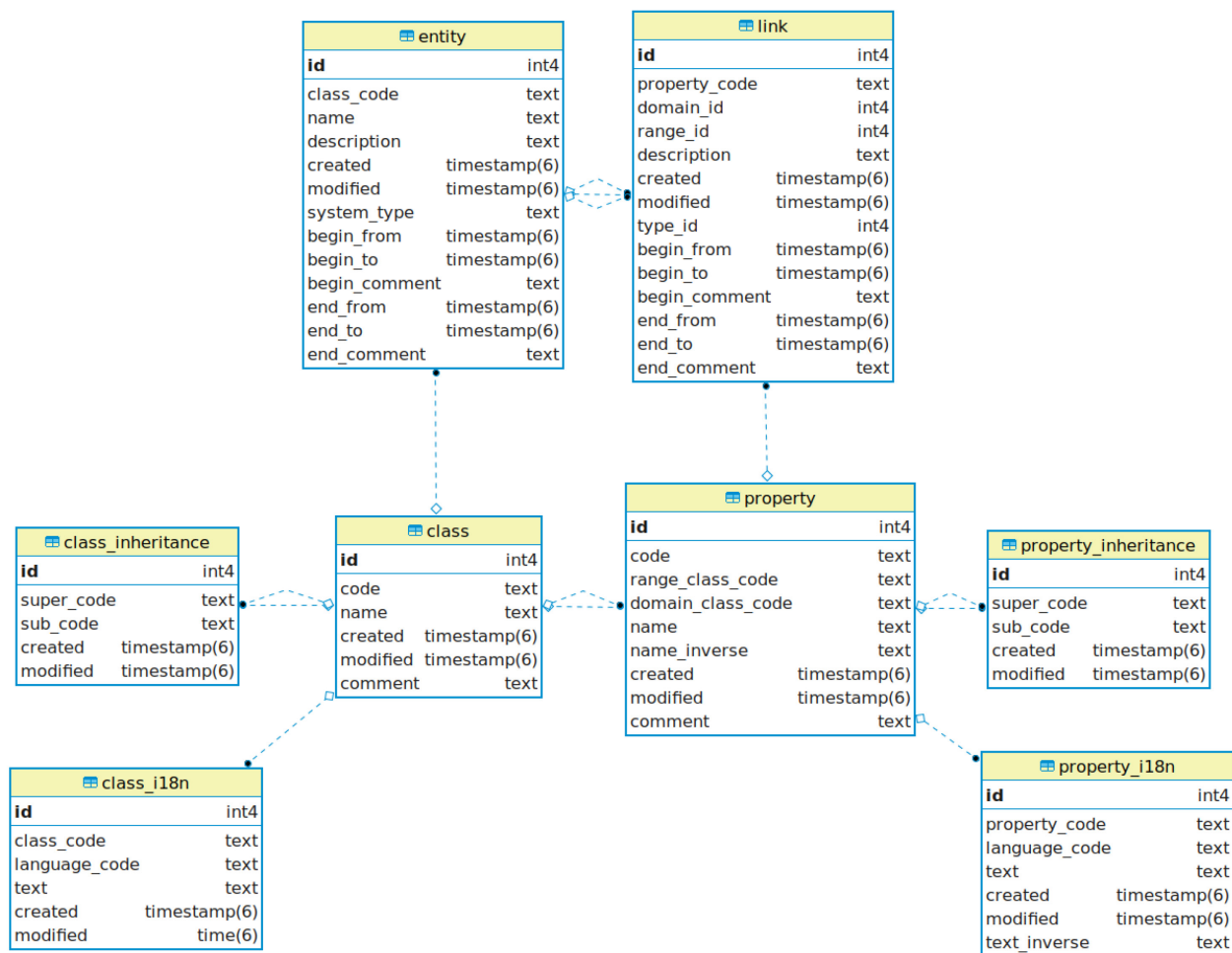
The OpenAtlas model for MEDCON is based on classes and properties of the CIDOC CRM: <http://www.cidoc-crm.org>, version 6.2.1.

Entities are saved in the database with a CIDOC CRM class and can have links with a CIDOC CRM property to other entities. E.g. an entity with the class **Person** (E21) is connected via a link with the property **has current or former residence** (P74) to another entity with the class **Place** (E53).

Below is a simplified OpenAtlas model visualisation based on CIDOC CRM.



Database tables in schema model:



Following tables were imported from the CIDOC CRM:

class - all classes e.g. person, place, ...

class i18n - translations for class labels in multiple languages

class inheritance - hierarchical relations between classes

property - all properties e.g. has current or former residence

property_i18n - translations for property labels in multiple languages

property inheritance - hierarchical relations between properties

Entered project data resides in these two tables:

entity - most important fields are:

name

class code - the CIDOC CRM class code e.g E21

date fields: (begin from, begin to, begin comment, end from, end to, end comment)

link - used to link entities, most important fields are:

property code - the CIDOC CRM property code e.g P74

domain id - corresponding entity id

range_id - corresponding entity id

type id - corresponding type entity id

date fields: (begin from, begin to, begin comment, end from, end to, end comment)

2. GIS schema

linestring	
id	int4
entity_id	int4
name	text
description	text
type	text
created	timestamp(6)
modified	timestamp(6)
geom	geometry

point	
id	int4
entity_id	int4
name	text
description	text
type	text
created	timestamp(6)
modified	timestamp(6)
geom	geometry

polygon	
id	int4
entity_id	int4
name	text
description	text
type	text
created	timestamp(6)
modified	timestamp(6)
geom	geometry

This schema was used for storing geospatial data containing the tables linestring, point and polygon. An entity can have none, one or multiple entries in these tables. Most important fields are:

id - unique identifier

name - optional, e.g. second possible location

description

entity_id - corresponding entity id

geom - GIS information about a line, point or polygon

3. Recreate the database

The SQL file can be used to recreate the PostgreSQL database. E.g. in the terminal on a Linux system as postgres user:

Create an openatlas user for PostgreSQL

```
$ createuser openatlas -P
```

Create an empty PostgreSQL database owned by the openatlas user:

```
$ createdb openatlas -O openatlas
```

Add the postgis extension to the database

```
$ psql openatlas -c "CREATE EXTENSION postgis;"
```

Import the SQL

```
$ psql openatlas < path/to/sql_file.sql
```