

PATH documentation repository

This repository is maintained by Julian Bogdani

julian.bogdani@uniroma1.it

Welcome to the [PATHs](#) documentation repository, which is still at a **very early phase of drafting**.

All documentation is released under [MIT License](#)

As soon as new documentation is being made available by PATHs' team, it will immediately be added and published here (source code in [Markdown format](#)) and in our web site, paths.uniroma1.it (PDF format).

The PDF version is automatically exported from the Markdown one, available and maintained as a standalone repository on [PATHs' account on GitHub](#). Please refer always to the online version and consider the PDF version as a secondary source.

Please do not hesitate to report any inconsistency, error, or typographical misprint you may notice by opening a new [issue](#).

Available documentation:

- [Database schema](#)

Changelog

- v. **0.1** [2018-01-05] Initial commit

PAThs database schema documentation

This subfolder of PAThs' main documentation repository provides a complete documentation of the PAThs database, currently available at <http://db.bradypus.net>. The documentation is written in [Markdown](#) by the PAThs' team (Julian Bogdani) and is available in PDF format in [PAThs' website](#).

The current database engine used by PAThs is [SQLite](#), «a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. SQLite is the most used database engine in the world» (<https://www.sqlite.org>). The [DDL](#) describing each table is standard SQL and will work—with minor changes—with other database engines, like [MySQL](#), [PostgreSQL](#), etc.

Table of contents

- [The database managing system](#)
- Table list
 - i. [Manuscripts](#)
 - ii. [Collections](#)
 - iii. [Works](#)
 - iv. [Titles](#)
 - v. [Colophons](#)
 - vi. [Authors](#)
 - vii. [Places](#)
 - viii. [Bibliographic database](#)
- Auxiliary tables
 - i. [m_biblio](#)
 - ii. [m_msplaces](#)
 - iii. [m_placephase](#)
 - iv. [m_quires](#)

- v. [m_shelfmarks](#)
- vi. [m_toponyms](#)
- vii. [m_wkauthors](#)

- System tables

- i. [Files](#)
 - ii. [Charts](#)
 - iii. [Queries](#)
 - iv. [Rs](#)
 - v. [Userlinks](#)
 - vi. [Vocabularies](#)

PATHs' database managing system

This document lists a verbose description of the relationship of the different entities of the database, which are:

- Manuscripts
- Collections
- Works
- Titles
- Colophons
- Authors
- Places
- Bibliographic database
- Files

Manuscripts

- A `Manuscript` (ie. *Codicological Unit*) is composed by **one or many** fragments, identified by a `Collection` name, a `Shelfmark number` and eventually by page numbers
- A `Manuscript` is linked to **zero, one or many** `Places` ; for each connection a type (*production, storage or discovery*) must be specified
- A `Manuscript` is made up of **one or many** quires.
- A `Manuscript` is linked to **zero, one or many** `Bibliographical records`
- A `Manuscript` is linked to **zero, one or many** `Works`
- A `Manuscript` contains **zero, one or many** `Titles`
- A `Manuscript` contains **zero, one or many** `Colophons`
- A `Manuscript` can be linked to **zero, one or many** `Files`

Collections

- A `Collection` is linked to **zero, one or many** `Bibliographical records`

Manuscripts

DDL

```
CREATE TABLE paths__manuscripts (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user  
  cmclid      TEXT,  
  tm          TEXT,  
  ldab        TEXT,  
  lcbm        TEXT,  
  alias       TEXT,  
  issinglefrag TEXT, -- 1|0  
  stratigraphy TEXT,  
  modernhistory TEXT,  
  contents    TEXT,  
  dialect     TEXT, -- vocabularies.def = 'dialects'  
  dialectnotes TEXT,  
  chronofrom  INTEGER,  
  chronoto    INTEGER,  
  datingcriteria TEXT,  
  bookform    TEXT, -- vocabularies.def = 'book_form'  
  writingsupport TEXT, -- vocabularies.def = 'writing_support'  
  leaftotextant TEXT,  
  fragtot     INTEGER,  
  fragdim     TEXT,  
  leaftotestimated INTEGER,  
  quiresextant INTEGER,  
  gennotes    TEXT,  
  columns     TEXT, -- vocabularies.def = 'columns'  
  colnotes    TEXT,  
  linespercolumn INTEGER,  
  charsperline INTEGER,  
  alignment   TEXT, -- vocabularies.def = 'alignment'  
  alignmentnotes TEXT,  
  leafw       TEXT,  
  leafh       TEXT,  
  framew      TEXT,  
  frameh      TEXT,  
  intercolumnium TEXT,  
  marginup    TEXT,  
  marginlow   TEXT,  
  marginout   TEXT,  
  marginin    TEXT,  
  marginleft  TEXT,  
  marginright TEXT,  
  heighttenlines TEXT,  
  prophw      TEXT,  
  propwuw     TEXT,  
  dimnotes    TEXT,
```

```

quirenotes TEXT,
quiresig INTEGER, -- 1|0
quiretype TEXT,
quireposition TEXT, -- semi-colon-separated-values: vocabularies.def = 'positi
quirenrnotes TEXT,
pagination INTEGER, -- 1|0
pagtype TEXT, -- semi-colon-separated-values: vocabularies.def = 'regula
pagposition TEXT, -- semi-colon-separated-values: vocabularies.def = 'positi
pagnote TEXT,
foliation INTEGER, -- 1|0
foltype TEXT, -- vocabularies.def = 'regular_irregular'
folposition TEXT, -- semi-colon-separated-values: vocabularies.def = 'positi
folnotes TEXT,
modulartypology TEXT, -- vocabularies.def = 'modular_typology'
writingaxis TEXT, -- vocabularies.def = 'axis'
thickandthin BOOLEAN, -- 1|0
typeofeos TEXT, -- vocabularies.def = 'type_of_eos'
typeofm TEXT, -- vocabularies.def = 'type_of_m'
typeofr TEXT, -- vocabularies.def = 'type_of_r'
typeofy TEXT, -- vocabularies.def = 'type_of_r'
typeoff TEXT, -- vocabularies.def = 'type_of_r'
writingnotes TEXT,
lectionalsigns TEXT, -- semi-colon-separated-values: vocabularies.def = 'lectio
structuralsigns TEXT,
signsnotes TEXT,
abbreviations TEXT, -- semi-colon-separated-values: vocabularies.def = 'abbre
abbrnotes TEXT,
ruling BOOLEAN, -- 1|0
rulingtype TEXT, -- vocabularies.def = 'ruling_types'
rulingtool TEXT, -- vocabularies.def = 'ruling_tool'
pricking BOOLEAN, -- 1|0
rulingnotes TEXT,
leaftabs TEXT,
kolleseis INTEGER, -- 1|0
kolleseseisdescr TEXT,
tracesnotes TEXT,
kollemata INTEGER,
kollno INTEGER,
kolldim TEXT,
bindings INTEGER, -- 1|0
sewings INTEGER, -- 1|0
sewingtype TEXT,
covers TEXT, -- semi-colon-separated-values: vocabularies.def = 'cover
coversnotes TEXT,
holesno INTEGER,
holesarrangement TEXT, -- vocabularies.def = 'holes_arrangement'
holesnotes TEXT,
changeofhand TEXT,
marginalnotes TEXT,
palimpsest BOOLEAN, -- 1|0
palimpsestnotes TEXT,
restancient TEXT,
restmodern TEXT,
inks TEXT, -- semi-colon-separated-values: vocabularies.def = 'ink_t
redink BOOLEAN, -- 1|0
redinknotes TEXT,

```

```
decortype      TEXT,  -- semi-colon-separated-values: vocabularies.def = 'decor
decorsubj      TEXT,  -- semi-colon-separated-values: vocabularies.def = 'decor
decorationdescr TEXT,
enlargedinitials TEXT,
sourceinfo     TEXT,
editors        TEXT -- semi-colon-separated-values: vocabularies.def = 'persons
);

);
```

Links

- manuscripts >—< biblio (via m_biblio [+ details])
- manuscripts —< m_quires
- manuscripts >—< collections (via m_shelfmarks [+ shelfmark, pp])
- manuscripts >—< places (via paths__m_msplaces [+ type, source info, source note])
- manuscripts >—< works (via paths__userlinks)

- A **Collection** can be linked to **zero, one or many** **Files**
-

Works

- A **Work** is linked to **zero, one or many** **Bibliographical records**
 - A **Work** is written by **zero, one or many** **Authors** ; for each connection an authorship type (*stated*, *creator* or *master*) must be specified
 - A **Work** can be paired by **zero, one or many** **Titles**
 - A **Work** can be linked to **zero, one or many** **Files**
 - (A **Works** is contained in **one or many** **Manuscripts**)
-

Titles

- A **Title** is linked to **zero, one or many** **Bibliographical records**
 - A **Title** is contained in **one or many** fragments, identified by a **Collection** name, a **Shelfmark number** and eventually by page numbers.
 - A **Title** can be linked to **zero, one or many** **Files**
 - (A **Title** is contained in **one** **Manuscript**)
-

Colophons

- A **Colophon** is linked to **zero, one or many** **Bibliographical records**
 - A **Colophon** is contained in **one or many** fragments, identified by a **Collection** name, a **Shelfmark number** and eventually by page numbers
 - A **Colophon** can be linked to **zero, one or many** **Files**
 - (A **Colophon** is contained in **one** **Manuscript**)
-

Places

- A **Place** is linked to **zero, one or many** **Bibliographical records**
 - A **Place** can have **one or many** **Chronological phases**
 - A **Place** can be linked to **zero, one or many** **Files**
 - A **Place** have **one or many** **Geodata**
 - (A **Place** can be linked to **zero, one or many** **Manuscripts**)
 - [A **Place** can have **one or many** **Toponyms**]
-

Bibliographic record

- (A **Bibliographic record** can be linked to **zero, one or many** **Manuscripts**)
 - (A **Bibliographic record** can be linked to **zero, one or many** **Collections**)
 - (A **Bibliographic record** can be linked to **zero, one or many** **Works**)
 - (A **Bibliographic record** can be linked to **zero, one or many** **Titles**)
 - (A **Bibliographic record** can be linked to **zero, one or many** **Colophons**)
 - (A **Bibliographic record** can be linked to **zero, one or many** **Places**)
 - (A **Bibliographic record** can be linked to **zero, one or many** **Files**)
-

Files

- (A **File** can be linked to **zero, one or many** **Manuscripts**)
- (A **File** can be linked to **zero, one or many** **Collections**)
- (A **File** can be linked to **zero, one or many** **Works**)
- (A **File** can be linked to **zero, one or many** **Titles**)

- (A File can be linked to **zero, one or many** Colophons)
- (A File can be linked to **zero, one or many** Places)
- (A File can be linked to **zero, one or many** Bibliographic record)

Collections

DDL

```
CREATE TABLE paths__collections (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user id  
  fullname    TEXT,  
  cmclname    TEXT,  
  country     TEXT,  
  town        TEXT,  
  institution TEXT,  
  name        TEXT,  
  notes       TEXT,  
  editors     TEXT -- semi-colon-separated-values: vocabularies.def = 'persons'  
);
```

Links

- `collections` >—< `biblio` (via `m_biblio` [+ details])
- `collections` >—< `manuscripts` (via `m_shelfmarks` [+ shelfmark, pp])

Works

DDL

```
CREATE TABLE paths__works (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user  
  cmcl        TEXT,  
  title       TEXT,  
  cpq         TEXT,  
  cant        TEXT,  
  cavt        TEXT,  
  bho         TEXT,  
  bhg         TEXT,  
  bhl         TEXT,  
  cae         TEXT,  
  period      TEXT, -- CMCL legacy semi-colon-separated-values vocabularies.de  
  problems    TEXT, -- CMCL legacy  
  shelfmarks  TEXT, -- CMCL legacy  
  notes       TEXT,  
  litperiod   TEXT,  
  editors     TEXT -- semi-colon-separated-values: vocabularies.def = 'persons'  
);
```

Links

- `works >—< biblio` (via `m_biblio` [+ details])
- `works >—< author` (via `m_wkauthor` [+ type])
- `works >—< manuscripts` (via `paths__userlinks`)
- `works —< titles`

Titles

DDL

```
CREATE TABLE paths__titles (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user  
  cc          TEXT,  
  workcc      TEXT REFERENCES paths__works (cmcl), -- foreign key: works.cmcl,  
  msid        TEXT REFERENCES paths__manuscripts (id), -- foreign key: works.cm  
  type        TEXT, -- vocabularies.def = 'title_type'  
  description TEXT,  
  columns     TEXT,  
  modulartypology TEXT, -- vocabularies.def = 'modular_typology'  
  writingaxis  TEXT, -- vocabularies.def = 'axis'  
  thickandthin TEXT, -- 1|0  
  text        TEXT,  
  translation TEXT,  
  taggedtext  TEXT,  
  editors     TEXT -- semi-colon-separated-values: vocabularies.def = 'persons'  
);
```

Links

- titles >—< biblio (via m_biblio [+ details])
- titles >—< collections (via m_shelfmarks [+ shelfmark, pp])
- titles >— manuscripts
- titles >— works

Colophons

DDL

```
CREATE TABLE paths__colophons (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user  
  msid        TEXT REFERENCES paths__manuscripts (id), -- foreign key: manuscri  
  chronofrom  TEXT,  
  chronoto    TEXT,  
  datingcriteria TEXT,  
  dialect     TEXT, -- semi-colon-separated-values: vocabularies.def = 'dialect  
  dialectnotes TEXT,  
  greektext   TEXT,  
  writingaxis  TEXT, -- vocabularies.def = 'dialects'  
  thickandthin TEXT, -- 1|0  
  cryptography TEXT, -- 1|0  
  minusculecopitc TEXT, -- 1|0  
  minusculegreek TEXT, -- 1|0  
  description TEXT,  
  text        TEXT,  
  translation TEXT,  
  taggedtext  TEXT,  
  editors     TEXT -- semi-colon-separated-values: vocabularies.def = 'persons'  
);
```

Links

- [colophons](#) >—< [biblio](#) (via [m_biblio](#) [+ details])
- [colophons](#) >—< [collections](#) (via [m_shelfmarks](#) [+ shelfmark, pp])
- [colophons](#) >— [manuscripts](#)

Authors

DDL

```
CREATE TABLE paths__authors (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user id  
  name        TEXT, --  
  copticname  TEXT, --  
  greekname   TEXT, --  
  viaf        TEXT, --  
  title       TEXT, -- vocabularies.def = 'titles'  
  bio         TEXT, --  
  cmclname    TEXT, --  
  birth       INTEGER, --  
  death       INTEGER, --  
  editors     TEXT -- semi-colon-separated-values: vocabularies.def = 'persons'  
);
```

Links

- `author` >—< `biblio` (via `m_biblio` [+ details])
- `author` >—< `works` (via `m_wkauthor` [+ type])

Places

DDL

```
CREATE TABLE paths__places (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current u  
  name        TEXT,  
  copticname  TEXT,  
  greekname   TEXT,  
  arabicname  TEXT,  
  arabictranslname TEXT,  
  egyptianname TEXT,  
  tm          TEXT,  
  pleiades    TEXT,  
  timm        TEXT,  
  region      TEXT, -- vocabularies.def = 'dialects'  
  area        TEXT, -- semi-colon-separated-values: vocabularies.def = 'are  
  nomos       TEXT, -- vocabularies.def = 'nomos'  
  province    TEXT, -- vocabularies.def = 'provinces'  
  episcopalsee TEXT, -- 1|0  
  episcopalseefrom TEXT,  
  noteseπισκοπalsee TEXT,  
  typology    TEXT, -- vocabularies.def = 'site_typology'  
  toporeferredto TEXT, -- places.id  
  notestoporeferredto TEXT,  
  historicalreference TEXT,  
  datefrom    INTEGER,  
  dateto      INTEGER,  
  datingcriteria TEXT,  
  description TEXT,  
  hasplans    BOOLEAN, -- 1|0  
  hasphotos   BOOLEAN, -- 1|0  
  ismonastery BOOLEAN, -- 1|0  
  monasterynotes TEXT,  
  isnomoscapital BOOLEAN, -- 1|0  
  accuracy    TEXT,  
  editors     TEXT -- semi-colon-separated-values: vocabularies.def = 'pers  
);
```

Links

- `places` >—< `biblio` (via `m_biblio` [+ details])
- `places` —< `m_placefase`

- `places` —< `manuscripts` (via `paths__m_msplaces` [+ type, source info, source note])
- `places` —< `geodata` (via `paths__geodata`)
- ...

Bibliography

DDL

```
CREATE TABLE paths__biblio (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user id  
  short       TEXT,  
  authors     TEXT,  
  title       TEXT,  
  journal     TEXT,  
  volume      TEXT,  
  issue       TEXT,  
  voleditors  TEXT,  
  series      TEXT,  
  year        TEXT,  
  place       TEXT,  
  pages       TEXT,  
  isbn        TEXT,  
  issn        TEXT,  
  doi         TEXT,  
  url         TEXT,  
  notes       TEXT,  
  worldcat    TEXT,  
  editors     TEXT -- semi-colon-separated-values: vocabularies.def = 'persons'  
);
```

Links

- `biblio` >—< `authors` (via `m_biblio` [+ details])
- `biblio` >—< `collections` (via `m_biblio` [+ details])
- `biblio` >—< `colophons` (via `m_biblio` [+ details])
- `biblio` >—< `files` (via `m_biblio` [+ details])
- `biblio` >—< `manuscripts` (via `m_biblio` [+ details])
- `biblio` >—< `places` (via `m_biblio` [+ details])
- `biblio` >—< `titles` (via `m_biblio` [+ details])
- `biblio` >—< `works` (via `m_biblio` [+ details])

m_biblio

`m_biblio` is a junction table that servers the purpose of linking one or more records of any other table to one or more records of table `Bibliography`

DDL

```
CREATE TABLE paths__m_biblio (  
  id          INTEGER PRIMARY KEY, -- primary key  
  table_link  TEXT, -- dynamic foreign key, combined with id_link  
  id_link     INTEGER, -- dynamic foreign key, combined with table_link  
  short       TEXT,  
  details     TEXT,  
  images      TEXT  
);
```

Links

- Bibliographic database >— `m_biblio` -< Authors
- Bibliographic database >— `m_biblio` -< Collections
- Bibliographic database >— `m_biblio` -< Colophons
- Bibliographic database >— `m_biblio` -< Files
- Bibliographic database >— `m_biblio` -< Manuscripts
- Bibliographic database >— `m_biblio` -< Places
- Bibliographic database >— `m_biblio` -< Titles
- Bibliographic database >— `m_biblio` -< Works

m_msplaces

m_msplaces is a junction table that servers the purpose of linking one or more records of any table Manuscripts to one or more records of table Places

DDL

```
CREATE TABLE paths__m_msplaces (  
  id          INTEGER PRIMARY KEY, -- primary key  
  table_link  TEXT, -- dynamic foreign key, combined with id_link  
  id_link     INTEGER, -- dynamic foreign key, combined with table_link  
  type        TEXT,  
  place       INTEGER,  
  sourceinfo  TEXT,  
  sourcenotes TEXT,  
  sort        INTEGER  
);
```

Links

- Manuscripts >— m_msplaces —< Places

m_placephase

m_placephase servers the purpose of adding to each record of table Places one or more chronological phase description

DDL

```
CREATE TABLE paths__m_placephase (  
  id          INTEGER PRIMARY KEY, -- primary key  
  table_link  TEXT, -- dynamic foreign key, combined with id_link  
  id_link     INTEGER, -- dynamic foreign key, combined with table_link  
  phperiod    TEXT,  
  phperiodnote TEXT,  
  phtypology  TEXT,  
  phdescription TEXT  
);
```

Links

- Places —< m_placephase

m_quires

m_quires servers the purpose of adding to each record of table Manuscripts one or more quire description

DDL

```
CREATE TABLE paths__m_quires (  
  id          INTEGER PRIMARY KEY, -- primary key  
  table_link  TEXT, -- dynamic foreign key, combined with id_link  
  id_link     INTEGER, -- dynamic foreign key, combined with table_link  
  quireno     TEXT,  
  quirelayout TEXT,  
  quirelayoutnotes TEXT,  
  quirepatype TEXT,  
  quirepartype TEXT,  
  quirenotes  TEXT  
);
```

Links

- Manuscripts —< m_quires

m_shelfmarks

m_shelfmarks servers the purpose of adding to each record of table Manuscripts , Titles and Colophons one or more reference to a shelfmark ID

DDL

```
CREATE TABLE paths__m_shelfmarks (  
  id          INTEGER PRIMARY KEY, -- primary key  
  table_link  TEXT, -- dynamic foreign key, combined with id_link  
  id_link     INTEGER, -- dynamic foreign key, combined with table_link  
  fullsegnat  TEXT,  
  segnatid    TEXT,  
  collection  TEXT,  
  shelfmark   TEXT,  
  pp          TEXT  
);
```

Links

- Manuscripts —< m_shelfmarks
- Titles —< m_shelfmarks
- Colophons —< m_shelfmarks

m_toponyms

m_toponyms servers the purpose of adding to each record of table Places , one or more toponyms

DDL

```
CREATE TABLE paths__m_toponyms (  
  id          INTEGER PRIMARY KEY, -- primary key  
  table_link  TEXT, -- dynamic foreign key, combined with id_link  
  id_link     INTEGER, -- dynamic foreign key, combined with table_link  
  toponym     TEXT,  
  language    TEXT,  
  transcription TEXT,  
  reference    TEXT,  
  referencedetails TEXT,  
  datefrom    TEXT,  
  dateto      TEXT  
);
```

Links

- Manuscripts —< m_toponyms

m_wkauthors

m_wkauthors is a junction table that servers the purpose of linking one or more records of any table Works to one or more records of table Authors

DDL

```
CREATE TABLE paths__m_wkauthors (  
  id          INTEGER PRIMARY KEY, -- primary key  
  table_link  TEXT, -- legacy field  
  id_link     INTEGER REFERENCES paths__works (id), -- foreign key referring works.i  
  id          INTEGER PRIMARY KEY,  
  table_link  TEXT,  
  id_link     INTEGER REFERENCES paths__works (id), -- foreign key referring works.i  
  type        TEXT,  
  author      INTEGER REFERENCES paths__authors (id), -- foreign key referring autho  
  source      TEXT,  
  notes       TEXT,  
  sort        INTEGER  
);
```

Links

- Works —< m_wkauthors >— Authors

Files

DDL

```
CREATE TABLE paths__files (  
  id          INTEGER PRIMARY KEY, -- primary key  
  creator     TEXT, -- automatically filled by the RDBMS with the current user id  
  ext         TEXT,  
  keywords    TEXT,  
  description TEXT,  
  printable   INTEGER,  
  filename    TEXT  
);
```

Links

- files >—< authors (via userlinks)
- files >—< biblio (via userlinks)
- files >—< collections (via userlinks)
- files >—< colophons (via userlinks)
- files >—< manuscripts (via userlinks)
- files >—< places (via userlinks)
- files >—< titles (via userlinks)
- files >—< works (via userlinks)