# MIRACUM MAPPER

# *Installation und Administration*

Stand: 2024-08-26, Autor: Sebastian Mate

# 1 Anforderungen

* Windows-Umgebung mit .NET-Framework 4.8
* PostgreSQL-Server zur Speicherung der Daten
* Der Programmcode kann mit Microsoft Visual Studio 2017 kompiliert werden.

# 2 Vorbereitungen

## 2.1 Installation des PostgreSQL-Datenbank-Servers unter Ubuntu-Linux

Der PostgreSQL-Server kann mit folgenden Befehlen installiert werden:

ufw allow 5432  
apt install postgresql  
sudo -u postgres psql postgres  
CREATE USER mapper WITH PASSWORD 'xxxxxxxxxxxx';  
CREATE DATABASE mapper;  
GRANT ALL PRIVILEGES ON DATABASE mapper to mapper;

Vorgehen gemäß: <https://bosnadev.com/2015/12/15/allow-remote-connections-postgresql-database-server/>

## 2.2 Zusätzlichen Benutzer nur mit Lesezugriff anlegen:

Möchte man zusätzlichen Wissenschaftlern, z.B. für Auswertungen (siehe unten), lediglich Lesezugriff auf die Datenbank geben, kann dies wie folgt erfolgen:

sudo -u postgres psql postgres  
\c mapper;  
CREATE ROLE readonly WITH LOGIN PASSWORD 'xxxxxxxx' NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE NOREPLICATION VALID UNTIL 'infinity';  
GRANT CONNECT ON DATABASE mapper TO readonly;  
GRANT USAGE ON SCHEMA public TO readonly;  
GRANT SELECT ON ALL TABLES IN SCHEMA public TO readonly;  
GRANT SELECT ON ALL SEQUENCES IN SCHEMA public TO readonly;

# 3 Konfiguration des Programms

Das Programm wird über die Datei Config.dat konfiguriert. Dabei handelt es sich um eine zweispaltige Text-Datei die in der ersten Zeile den Datenbank-Connctionstring und in der zweiten Zeile den LDAP-Server enthält:

Server=hostname;Port=5432;User Id=mapper;Password=password;Database=mapper;Pooling=false;preload reader=true  
my.ldap.server.de

Die Datei wird beim ersten Programmstart automatisch verschlüsselt.

# 4 Die Datenbank des MIRACUM-Mappers

## 4.1 Übersicht über alle Tabellen

Mit den folgenden Daten kann die MIRACUM-Datenbank initialisiert werden (die SQL-Skripte müssen ggf. noch angepasst werden):

[MIRACUM Mapper (ohne LOINC).zip](file:///C:\download\attachments\126583705\MIRACUM%20Mapper%20(ohne%20LOINC).zip%3fversion=2&modificationDate=1578488328960&api=v2)

[MIRACUM Mapper (incl. LOINC).zip](file:///C:\download\attachments\126583705\MIRACUM%20Mapper%20(incl.%20LOINC).zip%3fversion=1&modificationDate=1578488328867&api=v2)

## 4.2 Übersicht über alle Tabellen

Im Folgenden werden die Tabellen in der Datenbank aufgelistet:

|  |  |
| --- | --- |
| **Tabelle** | **Beschreibung** |
| userids | Verwaltung der Benutzer und deren Rechte. |
| expertlevel | Definiert die Bezeichner der Zustände (siehe “Das Zustandsmodell des MIRACUM Mapper” in der Bedienungsanleitung). |
| terminfo | Dient zum Nachschlagen von Ziel-Codes, die beim Mapping vom Benutzer ins Feld “Ziel-Code” (oben rechts) eingegeben werden. |
| sourceterms | Speichert die vollständige lokale Terminologie, die auf einen Standard gemappt werden soll. |
| mapping | Speichert die eigentlichen Mappings. |
| log | Speichert das Programm-Log. |
| requiresversion | Dient zum Invalidieren älterer Programmversionen, falls sich das Datenbankschema so geändert hat, dass die alten Programmversionen dazu nicht mehr kompatibel sind. |

In den folgenden Abschnitten werden einige Tabellen näher erläutert.

## 4.3 Tabelle userids - Verwaltung der Benutzer, deren Rechte und Zustandsübergänge

Die Tabelle “userids” dient der Verwaltung der Benutzer und deren Rechte.

Benutzer, die sich zum ersten Mal in das Programm einloggen, aber die noch nicht registriert sind, werden automatisch in diese Tabelle mit enabled=false eingetragen.

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| userid | varchar | Die ID des Users, die auch zum Login über LDAP verwendet wird. |
| filterfrom | int4 | Das Default-Filterlevel (aus Tabelle “expertlevel”), das unter “Von:” in der GUI beim Programmstart angezeigt wird. |
| filterto | int4 | Das Default-Filterlevel (aus Tabelle “expertlevel”), das unter “Bis:” in der GUI beim Programmstart angezeigt wird. |
| allowbrowsehistory | bool | Wenn “true”, darf die Versionhistorie eines Mappings durchgeblättert werden. |
| forcedfilter | varchar | Ein zusätzlicher Filter, der auf die Tabelle “sourceterms” beim Laden der Quellterminologie angewendet wird. Hierüber kann gesteuert werden, dass ein Benutzer nicht alle Quell-Codes zu sehen bekommt. Beispiel: source\_code like '6%' |
| enabled | bool | Wenn “true”, ist der Benutzer für die Verwendung freitgeschalten. |
| transitions | varchar | Definiert die Zustandsübergänge für den Benutzer. Siehe unten. |
| local\_password\_md5 | varchar | Mit MD5 verschlüsseltes lokales Passwort. Ist dieses gesetzt, wird nicht die LDAP-Authentifizierung verwendet. **Mit Version 1.8.0 eingeführt.** |

Definition der Zustandsübergänge in der Spalte “transitions”:

Über diesen Eintrag wird definiert, von welchem Start-Zustand ein Mapping in welchen Ziel-Zustand überführt werden darf. Diese Definition erfolgt über eine Liste mit Einträgen der Form **s-z**, getrennt durch Leerzeichen, wobei **s**=Start-Zustand und **z**=Ziel-Zustand. Die Werte für **s** und **z** beziehen sich dabei auf die Spalte “level” in Tabelle “expertlevel” (siehe Abschnitt 4.2 unten).

Die Standard-Werte sind:

* Default Experte: 0-2 0-3 1-2 1-3 2-3 2-2 3-2 3-3 5-5
* Default Labor: 3-1 3-4 3-5 5-4 4-1 4-2 4-5 5-5
* Alle Rechte: 0-0 0-1 0-2 0-3 0-4 0-5 1-0 1-1 1-2 1-3 1-4 1-5 2-0 2-1 2-2 2-3 2-4 2-5 3-0 3-1 3-2 3-3 3-4 3-5 4-0 4-1 4-2 4-3 4-4 4-5 5-0 5-1 5-2 5-3 5-4 5-5

Dabei handelt sich für Einträge mit **s** != **z**um echte Übergänge (z.B. 0-2); bei **s** = **z** wird nur ein zusätzliches Speichern erlaubt. Beispielweise kann der Experte auch validierte Mappings nochmals speichern (aber nicht in einen anderen Zustand überführen), da 5-5 definiert ist.

Beim Anlegen eines neuen Mappings werden in die Drop-Down-Liste “nächster Zustand:” alle Einträge **s** != **z** übernommen. Beim Editieren eines bereits vorhandenen Mappings werden, abhängig vom Ausgangszustand a, die Transitionen **s** = **z** mit **s** = **a** geladen. Die Befüllung der Drop-Down-Liste erfolgt entsprechend des “transitions”-Eintrags in Tabelle “userids” von links nach rechts.

## 4.4 Tabelle expertlevel - Definition der Bezeichner der Zustände

Die Tabelle “expertlevel” definiert die Bezeichner der Zustände (siehe “Das Zustandsmodell des MIRACUM Mapper” in der Bedienungsanleitung).

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| level | int4 | Das Level. |
| description | varchar | Die Beschreibung des Levels. |

Die Default-Einträge sind:

* 0: Ungemappt
* 1: Zum Überarbeiten (Exp.)
* 3: Gemappt (Experte)
* 4: In Bearbeitung (Labor)
* 5: Validiert (Labor)
* 2: In Bearbeitung (Experte)

## 4.5 Tabelle terminfo - Zielcode-Lookup

Die Tabelle “terminfo” dient dem Nachschlagen von Ziel-Codes, die beim Mapping vom Benutzer ins Feld “Ziel-Code” (oben rechts) eingegeben werden.

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| code | varchar(50) | Der Code der Zielterminologie. |
| description | varchar(500) | Die Beschreibung des Codes. |

## 4.6 Tabelle sourceterms - Die zu mappenden lokalen Codes

Die Tabelle “sourceterms” speichert die vollständige lokale Terminologie, die auf einen Standard gemappt werden soll.

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| source\_code | varchar(50) | Der Code der Quellterminologie. |
| source\_desc | varchar(500) | Die Beschreibung des Codes. |

## 4.7 Tabelle mapping - Die Mappings

Die Tabelle “mapping” speichert die eigentlichen Mappings.

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| source\_code | varchar | Der Code der Quellterminologie. |
| target\_code | varchar | Der Code der Zielterminologie. |
| sec\_source\_code | varchar | Der Sekundärcode aus der Quellterminologie.Noch nicht festgelegt oder implementiert. |
| sec\_source\_code\_cond | varchar | Die mit dem Sekundärcode verknüpfte Bedingung. Noch nicht festgelegt oder implementiert. |
| documentation | varchar | Der Dokumentationstext / Verlauf. |
| mapping\_level | int4 | Der aktuelle Mapping-Zustand. |
| version | int4 | Die Version des Mappings. |
| deleted | int4 | Gibt an, ob das Mapping gelöscht (=1) oder aktiv ist (=0). |
| timestamp | timestamp | Der Zeitstempel der angibt, wann das Mapping gespeichert wurde. |
| saved\_by | varchar | Gibt an, welcher User das Mapping gespeichert hat. |
| sw\_version | varchar | Gibt an, welche Software-Version des Tools das Mapping gespeichert hat. |

## 4.8 Tabelle log - Das Log

Die Tabelle “log” speichert das Programm-Log.

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| timestamp | timestamp | Der Zeitstempel der angibt, wann der Eintrag gespeichert wurde. |
| text | varchar(4000) | Der eigentliche Logging-Text. |
| ip | varchar | Die IP-Adresse des Anwenderrechners. |

## 4.9 Tabelle requiresversion - Invalidieren älterer Programmversionen

Die Tabelle “requiresversion” dient dem Invalidieren älterer Programmversionen, falls sich das Datenbankschema so geändert hat, dass die alten Programmversionen dazu nicht mehr kompatibel sind.

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| requiresversion | int4 | Die Schema-Version, die das Client-Programm unterstützen muss. |

## 4.10 Tabelle schemaversion

Der Eintrag “schemaversion” dient dazu, Programmversionen ab 1.8.2 mitzuteilen, welche Features (z.B. lokale Passwörter) im DB-Schema unterstützt werden.

|  |  |  |
| --- | --- | --- |
| **Spalte** | **Datentyp** | **Beschreibung** |
| schemaversion | int4 | Die aktuelle Version des DB-Schemas. |

# 5 Abfragen der Datenbank des MIRACUM-Mappers

Das Programm verwendet eine recht komplizierte Tabelle mit Versionierung. Um eine einfache Mapping-Tabelle zu bekommen, kann man folgendes Statement verwenden:

## 5.1 Export der jetzt aktuellen Mappings (letzte Version) abhängig vom Zustand

**select**

**distinct** source\_code SWISSLAB,

swisslab\_description,

loinc\_description,

target\_code loinc\_code,

**version**,

expertlevel.**level** mapping\_level,

expertlevel.description mapping\_level\_description

**from**

(

**select**

sourceterms.source\_code, sourceterms.source\_desc swisslab\_description,

terminfo.description loinc\_description,

**coalesce**(**mapping**.target\_code, '---') **as** target\_code,

sourceterms.source\_desc, **coalesce**(**mapping**.mapping\_level, 0) **as** mapping\_level,

**coalesce**(**mapping**.**version**, 0) **as** **version**, **coalesce**(**mapping**.deleted, 0) **as** deleted

**from**

sourceterms

**left** **join** (

**select**

m.source\_code, m.target\_code, m.**version**, m.deleted, m.mapping\_level

**from**

**mapping** m, (

**select**

source\_code, target\_code, **max**(**version**) **as** **version**

**from**

**mapping**

**group** **by**

source\_code, target\_code) **as** l

**where**

m.source\_code = l.source\_code

**and** m.target\_code = l.target\_code

**and** m.**version** = l.**version**) **as** **mapping** **on**

sourceterms.source\_code = **mapping**.source\_code

**left** **join** terminfo **on**

**mapping**.target\_code = terminfo.code ) **as** a,

expertlevel

**where**

mapping\_level **in** ('1', '3', '5')

**and** deleted = 0

**and** target\_code != 'Kein Mapping'

**and** target\_code != '---'

**and** **level** = mapping\_level

**order** **by**

mapping\_level **desc**,

source\_code

## 5.2 Export der ehemals gültigen Mappings (evtl. alte Version), bearbeitet damals durch eine gewisse Person

**Mapping-Tabelle nur auf Bearbeitungen von Katharina (Mapping-Expertin) beschränken (ignoriert alle späteren Änderungen durch Labore, etc.):**

**create** **or** **replace** **view** NurKatharina **as** (

**select** **distinct** source\_code, target\_code, sec\_source\_code, sec\_source\_code\_cond, documentation, mapping\_level, **version**, deleted, **timestamp**, saved\_by, sw\_version

**from** **mapping**

**except**

-- Entferne alle Mappings, die nicht von Katharina bearbeitet wurden oder von Katharina nach dieser Bearbeitung

**select** **distinct** **mapping**.source\_code, **mapping**.target\_code, **mapping**.sec\_source\_code, **mapping**.sec\_source\_code\_cond, **mapping**.documentation, **mapping**.mapping\_level,

**mapping**.**version**, **mapping**.deleted, **mapping**.**timestamp**, **mapping**.saved\_by, **mapping**.sw\_version

--, ErsteExterneBearbeitung.source\_code, ErsteExterneBearbeitung.ts

**from** **mapping**

**join**

(**select** **distinct** source\_code, **min**(**timestamp**) ts **from** (

**select** source\_code, target\_code, sec\_source\_code, sec\_source\_code\_cond, documentation, mapping\_level, **version**, deleted, **timestamp**, saved\_by, sw\_version **from** **mapping** m1

**join** (**select** **replace**(userid, 'MEDADS\', '') userid **from** userids **where** userid **not** **like** '%ehrenbka%' **and** userid **not** **like** '%matesn%') KommentiertDurchLabor

**on** m1.documentation **like** '%' || KommentiertDurchLabor.userid || '%'

**union**

**select** source\_code, target\_code, sec\_source\_code, sec\_source\_code\_cond, documentation, mapping\_level, **version**, deleted, **timestamp**, saved\_by, sw\_version **from** **mapping**

**where** saved\_by **in** (**select** userid **from** userids **where** userid **not** **like** '%ehrenbka%' **and** userid **not** **like** '%matesn%')

) MappingsNichtVonKatharina

**group** **by** source\_code) ErsteExterneBearbeitung

**on** ErsteExterneBearbeitung.source\_code = **mapping**.source\_code

**where** ErsteExterneBearbeitung.ts <= **mapping**.**timestamp**

**order** **by** **timestamp**

);

**Exportiert das letzte Mapping von Katharina (angepasst auf obigen View):**

**select**

**distinct** source\_code SWISSLAB,

swisslab\_description,

target\_code loinc\_code

**from** (

**select** sourceterms.source\_code,

sourceterms.source\_desc swisslab\_description,

**coalesce**(NurKatharina.target\_code, '---') **as** target\_code,

sourceterms.source\_desc,

**coalesce**(NurKatharina.mapping\_level, 0) **as** mapping\_level,

**coalesce**(NurKatharina.**version**, 0) **as** **version**,

**coalesce**(NurKatharina.deleted, 0) **as** deleted

**from**

sourceterms

**left** **join** (

**select** m.source\_code, m.target\_code, m.**version**, m.deleted, m.mapping\_level

**from** NurKatharina m,

( **select** **distinct** source\_code, target\_code, **max**(**version**) **as** **version**

**from** NurKatharina

**where** mapping\_level **in** ('3')

**group** **by**

source\_code, target\_code) **as** l

**where** m.source\_code = l.source\_code **and** m.target\_code = l.target\_code **and** m.**version** = l.**version**) **as** NurKatharina

**on** sourceterms.source\_code = NurKatharina.source\_code) **as** a,

expertlevel

**where** mapping\_level **in** ('3') **and** deleted = 0 **and** **level** = mapping\_level

**order** **by** loinc\_code;