

documentation

Elexis Datenbank Wartungs Tool

Bezeichnung: Datenbank Wartungs Tool

Author: Marco Descher <descher@medevit.at>

Version: 1.0.13, September 2011

Beschreibung: Dieses Plug-In prüft die von Elexis 2.1 verwendete Datenbank auf syntaktische und semantische Fehler sowie Problemen bei der referenziellen Integrität.

Anforderungen

Diese Plug-In benötigt Elexis in der 2.1 Version. Als Datenbank-Backend werden derzeit MySQL und PostgreSQL unterstützt. Derzeit wird auf Datenbank-Backend V1.8.6 geprüft.

Installation

Die Installation des Plug-Ins selbst erfolgt mittels des Konfigurators, es müssen keinerlei spezielle Massnahmen ergriffen werden.

Konfiguration

Das Plug-In benötigt keine spezielle Konfiguration.

Unterstützte Datenbanken und Versionen

Datenbank	Version
Postgres	1.8.6, 1.8.7, 1.8.8, 1.8.9, 1.8.10, 1.8.11
MySQL	1.8.6, 1.8.7, 1.8.8, 1.8.9, 1.8.10, 1.8.11

Begriffsdefinition

Fehler in der Datenbank-Struktur können sich auf drei verschiedenen Ebenen darstellen:

- Syntaktik
- Semantik
- Referenzielle Integrität

Diese Sektion wird die Bedeutung dieser Begriffe definieren sowie Beispiele zu Fehlzuständen liefern.

Syntaktische Korrektheit

Die *syntaktische Korrektheit* einer Datenbank stellt den gültigen Zustand der durch das Modell definierten Tabellen und deren Felder dar. Das bedeutet dass sämtliche, innerhalb der Datenbank benötigten, Tabellen erstellt sind, diese über die per Definition notwendigen Felder verfügen und diese Felder den korrekten Datentyp aufweisen.

Ist dies nicht gegeben kann es zu Fehlermeldungen wie zB `Unknown column 'synonyms' in 'field list'` beim Fehlen eines Feldes kommen. Ist ein falscher Datentyp für das Feld spezifiziert können Fehler wie `ch.elexis.data.PersistenceException: Fehler bei: UPDATE TABELLE SET ERROR:wert=?, lastupdate=? WHERE ID='VERSION' (wert=0.1.0) entstehen.`

Dieser Fehl-Zustand kann erreicht werden, wenn ein Upgrade der Datenbank aus irgendwelchen Gründen fehlschlägt. Um solche Fehler zu identifizieren überprüft das Tool die syntaktische Korrektheit sowie Vollständigkeit der Einträge.

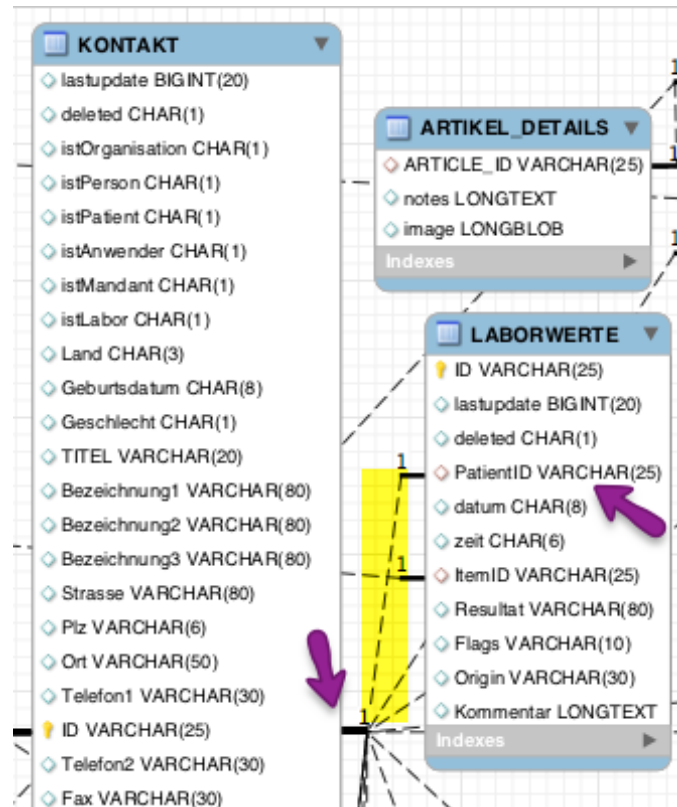
Semantische Korrektheit

Die *semantische Korrektheit* definiert den bedeutungs-korrekten Zustand. Die Semantik einer Aussage definiert die Bedeutung der Aussage. Ein Beispiel, die Berechnung $1+2=3$ weißt eine andere Syntaktik als die Berechnung $1-0+(2-0)=3$ auf, die Semantik der Operation (in diesem Fall die Addition von 1 zu 2) ist jedoch ident.

Semantische Fehl-Zustände können sich bilden wenn Einträge in der Datenbank existieren die zwar syntaktisch korrekt sind, deren Bedeutung aber keinen Sinn ergibt. So ist zum Beispiel ein Eintrag in der Tabelle **KONTAKT** der per Definition ein Patient aber keine Person ist semantisch inkorrekt. (Es sei denn es können auch Organisationen auf eine Krankheit behandelt werden; diese evtl. im Consulting-Bereich gültige Tatsache wird aussen vorgelassen).

Referenzielle Integrität

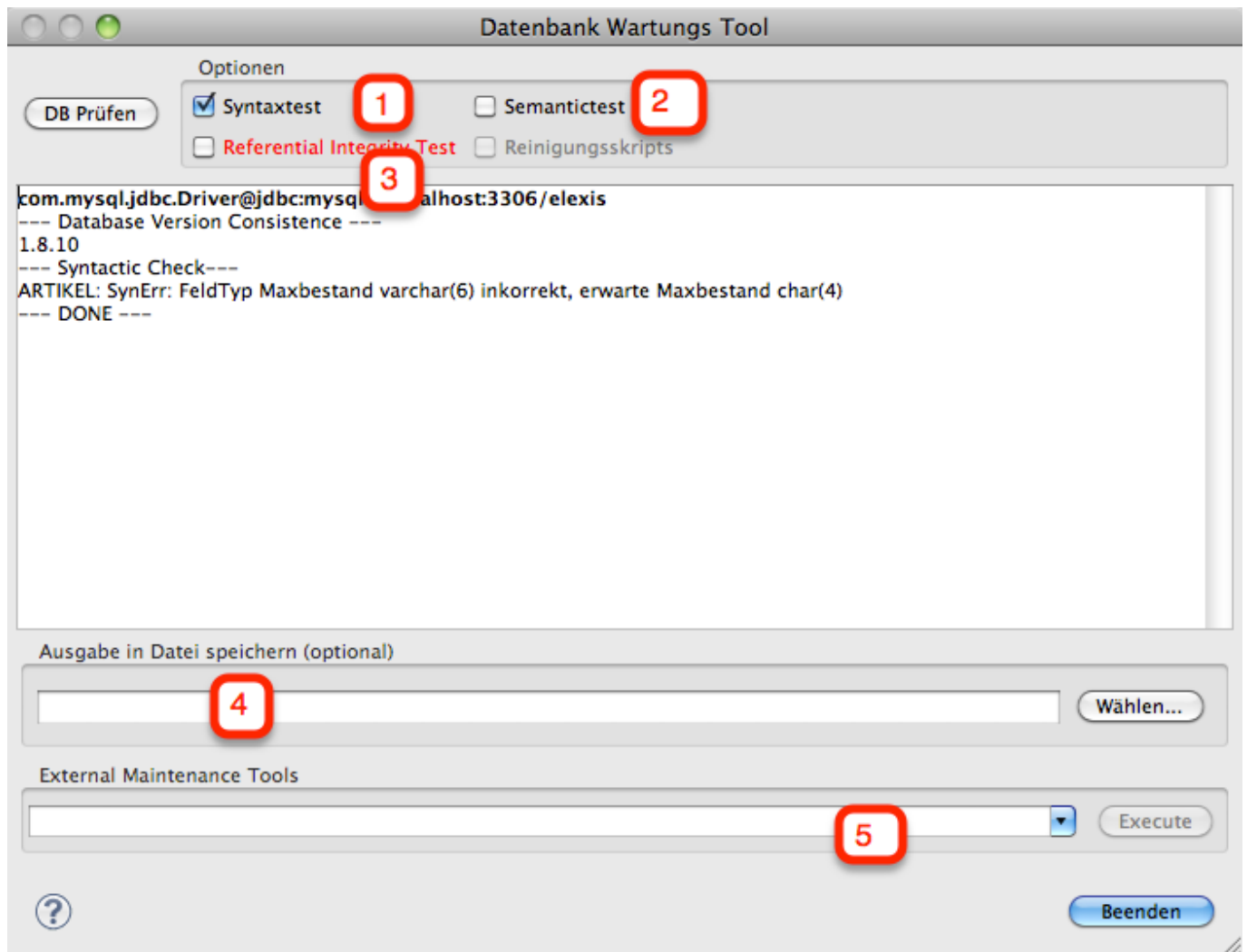
Die *referenzielle Integrität* definiert die Korrektheit von Einträgen in Beziehung aufeinander. Zum Beispiel soll sich ein Eintrag a nicht auf einen Eintrag b beziehen, der nicht in der Datenbank vorhanden ist.



Beispiel: Die dargestellte Tabelle **LABORWERTE** beinhaltet die Einträge welche im LaborView angezeigt werden. Die Verbindung zwischen einem Laborwert und einem Kontakt bildet hier das Feld **PatientID**.
Existiert nun ein Eintrag in der Tabelle **LABORWERTE** welche einen Eintrag **PatientID** besitzt, der in der Tabelle **KONTAKT** nicht existiert, handelt es sich um einen Fehler in der referenziellen Integrität; ein solcher Datensatz ist "verloren" bzw. steht in keiner Verbindung und wird daher nicht angezeigt.
Fehler in der referenziellen Integrität können zu inkonsistentem Verhalten der Applikation führen.

Verwendung des Tools

Das Tool kann über den Menüeintrag **Hilfe / DB Überprüfung** aufgerufen werden.



Unter **Optionen** definieren Sie welche Tests sie durchführen lassen wollen:

1. entspricht der einfachen syntaktischen Überprüfung
2. dem semantischen Korrektheitstest
3. überprüft die referenzielle Integrität (ACHTUNG: Dieser Test kann je nach Datenbank-Größe sehr lange dauern)
4. bieten Ihnen optional die Möglichkeit die Ausgabe in einem Logfile zu speichern. Sie können diese Datei ihrem Supporter zukommen lassen um eventuelle Fehler in der Datenstruktur leichter zu identifizieren. **Es werden hierbei keinerlei sensitive Daten gespeichert!**
5. Ist ein sogenannter *Hook* für externe Tools die Datenbankbereinigungs-Skripts integrieren wollen. Je nachdem welche Plug-Ins sich sonst auf Ihrem System befinden sehen Sie hier Aufgaben die gestartet werden können, oder nicht.

Mittels **DB Prüfen** führen Sie die respektiven gewählten Tests durch.

Mögliche Fehler Syntaktik-Test

Folgende mögliche Fehler können beim Syntaktik-Test auftreten, wobei die Ausgabe folgende ist:

TABELLE: SynErr: Beschreibung

Fehler	Beschreibung	Lösung	Fix-Beispiel
--------	--------------	--------	--------------

TABELLE: SynErr: Feld FeldName FeldDatenTyp nicht gefunden!	Ein Feld das in der Tabelle TABELLE existieren sollte wurde nicht gefunden.	Das entsprechende Feld muss in Tabelle TABELLE mit dem korrespondierenden Datentyp erstellt werden.	MySQL: ALTER TABLE Artikel ADD LastImport char(8);
TABELLE: SynErr: FeldTyp FeldName FeldDatenTyp inkorrekt, erwarte FeldName FeldDatenTyp!	Ein Feld in der Tabelle TABELLE entspricht nicht dem erwarteten Datentyp.	Der Datentyp muss mittels SQL ALTER Befehl korrekt gesetzt werden.	MySQL: ALTER TABLE kontakt MODIFY Website varchar(80);

Mögliche Fehler Semantik-Test

Eine Tabelle kann gezielt nach semantisch inkorrekten Einträgen befragt werden, im Fall der Identifikation eines solchen Wertes erhalten Sie eine Ausgabe wie zB:

```
KONTAKT: Semantischer Fehler bei Query <<Bezeichnung1 LIKE ''>> auf ID
5a0be2a6f053f43d1a2263
```

wobei innerhalb << und >> die Abfrage angegeben ist, welche den Fehlzustand identifiziert hat.

Diese Ausgabe zB. zeigt einen semantischen Fehler in der Tabelle KONTAKT an. Der Fehler entsteht aus der Tatsache dass der Eintrag mit der ID 5a0be2a6f053f43d1a2263 in Feld Bezeichnung1 keinen Eintrag enthält. Dies ist kein gültiger Zustand, da ein Kontakt ohne Bezeichnung nicht korrekt angezeigt wird. Der entsprechende Eintrag kann mit hoher Wahrscheinlichkeit gelöscht werden.

Mögliche Fehler Referenzelle Integrität

Fehler in diesem Bereich stellen sich immer mittels TABELLE-A: ID Eintrag-ID - FeldName does not have an associated entry in TABELLE-B or is NULL dar. Dies bedeutet das ein Gegeneintrag (Fremdschlüssel, siehe Datenbank-Model) der für einen Eintrag FeldName in TABELLE-A existieren soll kein entsprechender Gegeneintrag in TABELLE-B existiert oder der Eintrag FeldName selbst nicht gesetzt ist.

Es muss von Fall zu Fall identifiziert werden ob es sich hier um einen kritischen Eintrag handelt, dies ist zum Beispiel möglich wenn in einer Datenbank-Konsole folgendes eingegeben wird:

```
SELECT * FROM TABELLE WHERE ID = 'ID';
```

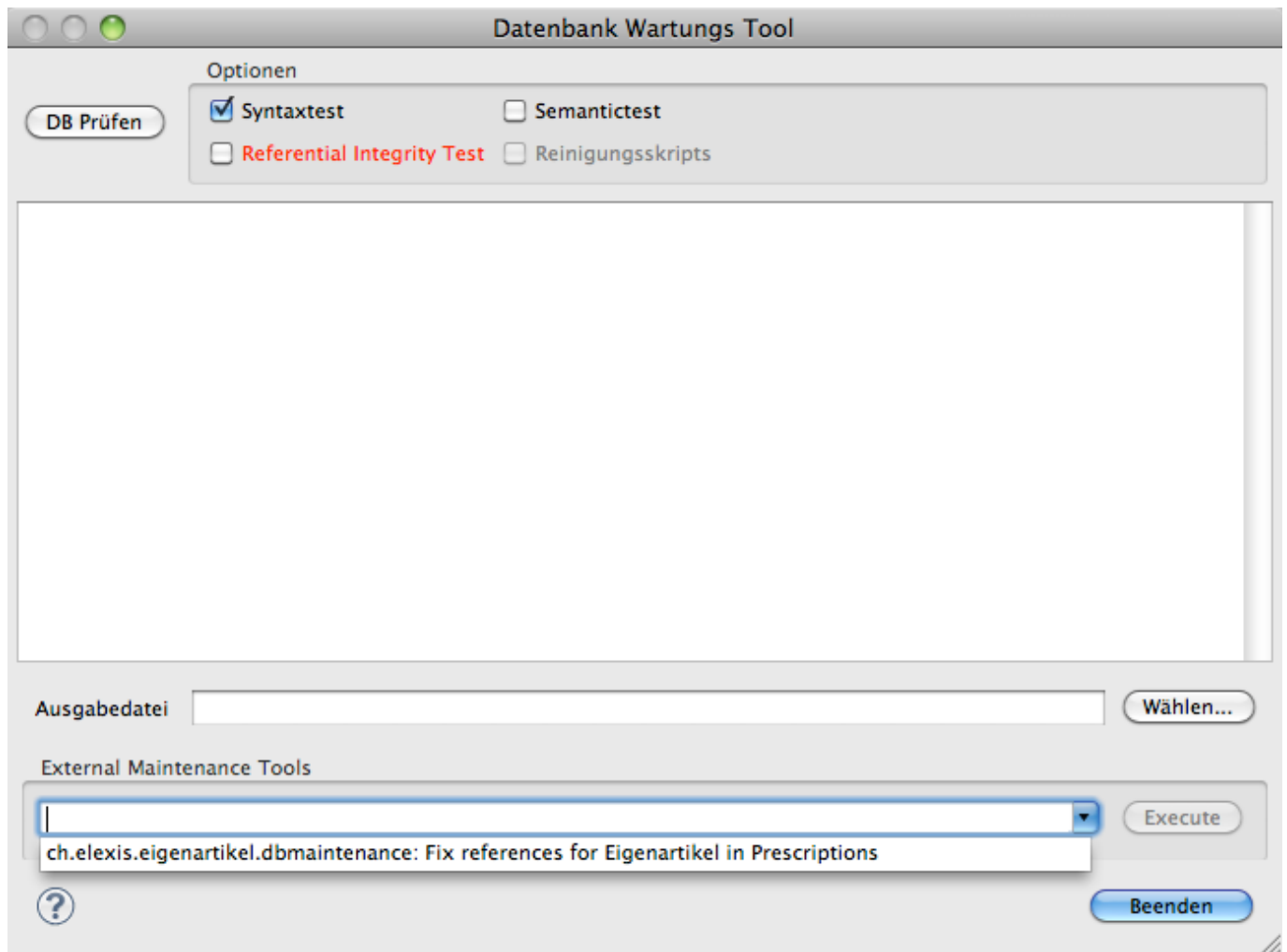
Es wird dann der komplette Eintrag für ID angezeigt, und die Zuordnung kann identifiziert werden.

Datenbank-Model

Im Rahmen der Erstellung dieses Wartungs-Tools wurde ein Entity-Relationship-Diagramm mittels MySQL-Workbench erstellt, welche die Beziehungen zwischen den Tabellen darstellt. Eine PDF Version dieser Datei finden Sie beiliegend unter Flexis-DB-Model-1.8.6-optimized.pdf.

Externe Verwedung

Seit Version 1.0.4 verfügt das Plug-In über den Erweiterungspunkt ExternalMaintenance. Hier können sich dritte Plug-Ins ankoppeln um Datenbank-Wartungsskripts durchzuführen. Die beim System registrierten Third-Party Skripts werden nun neu im Datenbank-Wartungs-Tool angezeigt.



Weiter Informationen zur programmatischen Verwendung des Extension Points finden Sie in der Extension Point Dokumentation.

Anmerkungen

Dieses Tool erhebt keinen Anspruch darauf sämtliche möglichen Fehler in eine Datenbank identifizieren zu können. Es ist in diesem Kontext nicht möglich sämtliche Fehlerzustände logisch darzustellen und überprüfbar zu machen. (Dies ist ein Logik-Problem, dass sich aus der relativen Unschärfe der Datenbank-Struktur ergibt, d.h. gegeben den Zustand der Datenbank ist es praktisch nicht möglich jeglichen Komplementär-Zustand – also Fehlerzustand – zu beschreiben).

Werden jedoch im Rahmen identifizierter Probleme bzw. Bugfixes solche Zustände identifiziert, wird eine Kontrolle auf einen solchen Zustand in das Wartungs-Tool aufgenommen.

Offene Punkte

- Derzeit werden Fehler nicht automatisch behoben, dies ist in einer späteren Version geplant.

Changelog

1.0.1 – 18.04.2011

- Modified SyntacticCheckMySQL to care for upper case written Tables too

- Integrated Context-Help into TrayDialog
- 1.0.2 - 09.05.2011
 - Support for 1.8.7 and 1.8.8
 - Datenbank Test wird nun aufgrund des Eintrages in der DB und nicht Hub.DBVersion gewählt
- 1.0.3 - 09.05.2011
 - Support for 1.8.9
 - Fix after merge with 2.1.5.x
- 1.0.4 - 10.05.2011
 - Support for external Maintenance Codes (Extension Point and Superclass ExternalMaintenance.java)
- 1.0.5 - 6.6.2011
 - Fix for table name case sensitivity on MySQL
- 1.0.6 - 10.6.2011
 - Added contribution script to fix ticket #13
- 1.0.7 - 21.6.2011
 - Ticket #13 script now also checks deleted items, and sets articles with subid '' to '0000000'
- 1.0.8 - 30.6.2011 - N. Giger
 - Added support for DB Model 1.8.10
- 1.0.9 - 30.6.2011 - M. Descher
 - Fixed bug in DBTestDialog leading to InvocationTargetException when no file selected
- 1.0.10 - 7.7.2011 - M. Descher
 - Fixed bugs mentioned in Ticket #65, i.e.:
 - Usability problem with selection of output file and possible creation of new log file
 - The user now gets a more meaningful progress report
 - Fixes for erroneous report of datatype comparison with MySQL
 - Code cleaning
 - Fixed InvocationTargetException in Referential Integrity Test Execution
- 1.0.11 - 28.8.2011 - G. Weirich
 - adapt to ch.elexis.core
- 1.0.12 - 4.9.2011 - M. Descher
 - Experimental implementation of fix script (MySQL only) (Ticket #580)
 - Test-Integration table collation (Ticket #571)
 - Minor fixes
- 1.0.13 - 9.9.2011 - M. Descher
 - Added support for DB Model 1.8.11