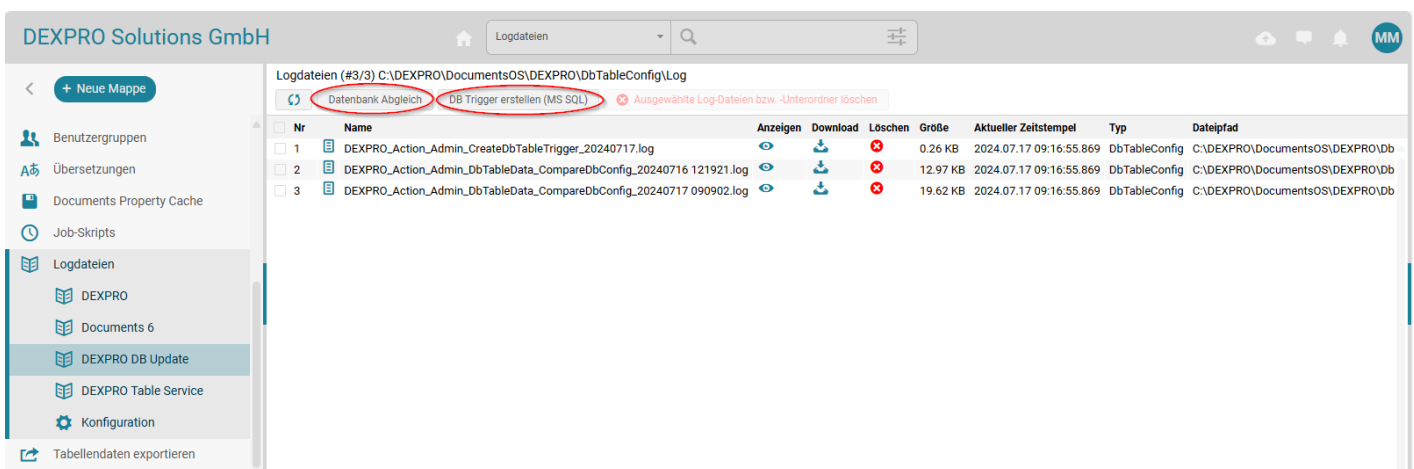


# Update Datenbank-Tabellen

## Abgleich-Skript

Auf der "**Administrations**"-Outbar befindet sich auf der Sub-Outbar "**Allgemein**" der Ordner "**Logdateien**" mit dem Unterordner "**DEXPRO DB Update**". Hierüber wird das Skript "**DEXPRO\_Action\_Admin\_DbTableData\_CompareDbConfig**" ausgeführt. Alternativ kann das Skript auch über den Manager ausgeführt werden. Die zugehörige Log-Datei wird nach einer Aktualisierung der Ansicht direkt im Ordner angezeigt. Das Skript benötigt etwas Zeit und ggf. muss die Ansicht erneut aktualisiert werden. Über das Augen-Symbol kann die Logdatei direkt angesehen werden. Abweichungen werden als Fehler **rot** markiert.



Das Portal-Skript liest die aktuell beim Kunden vorhandene Datenbank aus und sucht nach fehlenden Tabellen und Tabellen-Spalten und legt diese an. Zudem werden alle Tabellen-Spalten auf Datentyp, Default-Werten und der Einstellung ob NULL-Werte zulässig sind überprüft. Zusätzlich werden Trigger und Primary- sowie Unique-Keys überprüft. Fehlende Datenbank-Trigger unter MS SQL können über das Skript "**DEXPRO\_Action\_Admin\_CreateDbTableTrigger\_All**" hinzugefügt werden.

Das Skript kann viele Abweichungen wie fehlende Spalten oder Tabellen direkt beheben. Einige Abweichungen wie Veränderungen bei einem Unique-Key müssen manuell behoben werden. Hinweise hierzu folgen weiter unten auf dieser Seite.

Anpassungen an einer Tabellen-Spalte schlagen jedoch fehl, wenn die Spalte in einem Unique-Key verwendet werden. Unique-Keys werden aktuell nicht automatisch erstellt oder angepasst. Somit kann es sein, dass manuell nachgearbeitet werden muss.

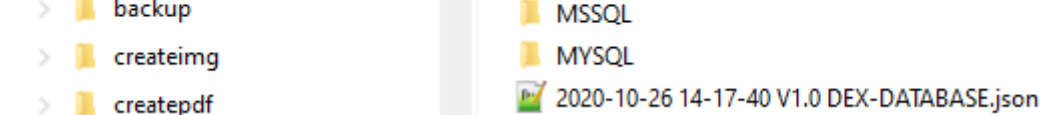
Wenn das Skript auch nach mehrfacher Ausführung Abweichungen nicht automatisch auflösen kann, müssen die Anpassungen manuell hergestellt werden. Das Skript kann nach jeder Anpassung erneut ausgeführt werden. Der Vorgang kann wiederholt werden, bis die Log-Datei

keine weiteren Abweichungen entdeckt. Hilfestellungen zu den Fehlern werden weiter unten gegeben.

## Technischer Hintergrund

Für das Update der Datenbanken enthält der "**DEXPRO**"-Ordner den Unterordner "**DbTableConfig**". Dieser enthält wiederum einen "Log-Ordner" und die Ordner "**MSSQL**" und "**MYSQL**" sowie eine JSON-Datei "**DEX-DATABASE**" mit der Versionsnummer und einem Zeitstempel im Namen. Mit jedem Update wird eine neue JSON-Datei ausgeliefert. Die alten Dateien sollten aus dem Ordner entfernt werden, damit garantiert die neueste Version verwendet wird! Die Datei enthält die komplette Datenbank-Struktur im Soll-Zustand.

Die Ordner "**MSSQL**" und "**MYSQL**" enthalten wiederum Unterordner für die beiden Datenbanken "**DEX\_MasterData**" und "**DEX\_Workflow**", in welchen pro Tabelle einzelne SQL-Dateien mit den CREATE-Befehlen der Tabellen enthalten sind.



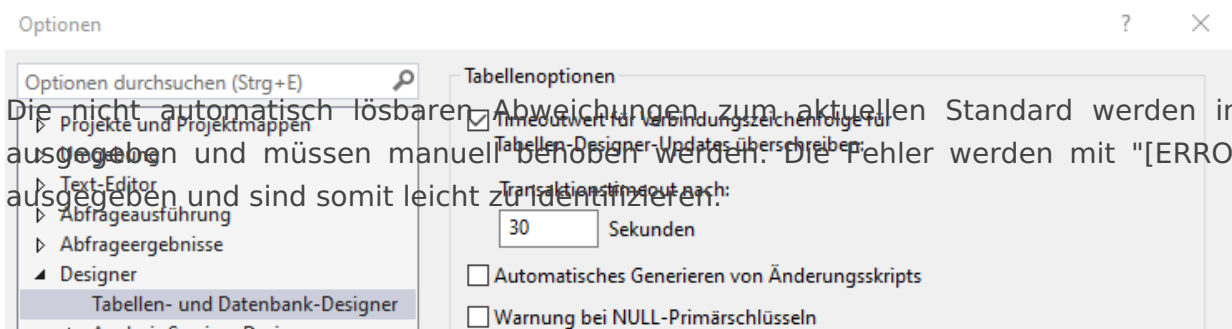
## Datenbank-Berechtigungen

Der ausführende Benutzer muss Datenbank-Tabellen hinzufügen können und er muss bestehende Tabellen ändern können!

In jedem Fall muss ein Zugriff mit entsprechenden Rechten auf die beiden Datenbanken gegeben sein! Bei MS SQL zum Beispiel über ein Management Studio.

## Manuelle Anpassungen an den Datenbanken

Auch wenn viele Anpassungen automatisch über das folgende Skript ausgeführt werden kann es sein, dass manuelle Anpassungen an bestehenden Tabellen vorgenommen werden müssen. Bei MS SQL ist dies in der Voreinstellung häufig deaktiviert. Die Einstellung "Speichern von Änderungen verhindern, die die Neuerstellung der Tabelle erfordern" muss hierfür aktiviert werden. Die Einstellung befindet sich unter "Extras" -> "Optionen" im Abschnitt "Designer" -> "Tabellen- und Datenbank-Designer". Die Checkbox muss wie im Screenshot deaktiviert sein.



Die nicht automatisch lösbaren Abweichungen zum aktuellen Standard werden im Skript-Log ausgegeben und müssen manuell behoben werden. Die Fehler werden mit "[ERROR]" markiert ausgegeben und sind somit leicht zu identifizieren.

## Datentyp-Abweichung korrigieren

Ein abweichender Datentyp wird im Log als Fehler angezeigt - muss aber nicht zwingend ein Fehler sein. Besonders in den Tabellen "Invoice\_Posting\_Head" und "Invoice\_Posting\_Pos" kann es sein, dass projektspezifisch bewusst Anpassungen auf sehr spezifische Typen vorgenommen wurden.

Im folgenden Beispiel wurde aus einer "int"-Spalte eine "bigint"-Spalte. Ohne die oben beschriebene Designer-Konfiguration würde beim Speichern ein Fehler erscheinen, da die Tabelle beim Speichern neu erstellt werden muss.

**[ERROR] [fields pos definition][ID] divergent data type .. standard:bigint != custom:int**

Microsoft SQL Server Management Studio

### Key-Abweichung korrigieren

Wie bereits erwähnt werden Abweichungen bei Unique-Schlüsseln nicht automatisch korrigiert. Bei einer MySQL Installation über eine MariaDB können die Unique-Keys genau so wie die Primärschlüssel direkt über die Tabellen-Konfiguration konfiguriert werden.

Spaltenname	Datentyp	NULL-Werte...
ID	int	
Licence	varchar(50)	
TemplateFile	varchar(50)	

Im Microsoft SQL Server Management Studio muss die Tabelle zunächst im Entwurfs-Modus geöffnet werden. In der Tabellen-Ansicht kann aber nur der Primärschlüssel für die Tabelle festgelegt werden. Durch das Öffnen der Tabelle erscheinend in der oberen Leiste der Reiter "Tabellen-Designer". Hier befindet sich der Eintrag "Indizes/Schlüssel...".

In dem Pop-Up werden die Primärschlüssel und die eindeutigen Schlüssel definiert.

Tabellen-Designer Extras Fenster Hilfe

Primärschlüssel entfernen

Indizes/Schlüssel

Ein bestehender Unique-Schlüssel um die Spalte "TemplateFile" erweitert wurde. Der technische Name von Unique-Keys ist nicht relevant, denn das Skript sucht nach einem Schlüssel mit denselben Spalten unabhängig vom Namen.

PK Fields\_Pos\_Definition 1  
Unique\_Fields\_Pos\_Definition

Eigenschaften für Primärschlüssel/eindeutiger Schlüssel oder Index.  
(Für Details klicken Sie auf die Schaltfläche "Hilfe".)

NULL-Werte...

```
26 [2020.11.04 14:46:53][ERROR] [fields_pos_definition] function compareTableKeys( standardKeysObj, customKeysObj, fields_pos_definition ) .. missing specific key 'licence_techfieldname'
27 .. standardKeysObj[uniqueKeys]: [
28 {
29   "desc": "Licence_TechFieldName",
30   "columns": [
31     "Licence",
32     "TechFieldName",
33     "TemplateFile"
34   ]
35 }
36 ]
37 .. customKeysObj[uniqueKeys]: [
38 {
39   "desc": "Unique_Fields_Pos_Definition",
40   "columns": [
41     "Licence",
42     "TechFieldName"
43   ]
44 }
45 ]
```

Wenn der Unique-Key noch nicht existieren sollte, dann muss zunächst ein neuer Schlüssel über den Button "Hinzufügen" erzeugt werden. Der Schlüssel muss vom Typen "Eindeutiger Schlüssel" sein. Wenn ein Schlüssel existiert aber nicht gefunden werden konnte, dann wurde evtl. ein falscher Schlüssel zugeordnet.

Unter "Identität" sollte ein sprechender Name für den Schlüssel angegeben werden. Bei Neuanlage beginnt der vordefinierte Name mit "IX", da als Typ "Index" voreingestellt ist. Aber auch wenn man

den Typen auf "Eindeutiger Schlüssel" ändert, wird das Präfix nicht angepasst. In unserer Auslieferung beginnen die Unique-Keys bei MS SQL mit "Unique", damit man den Typen des Schlüssels direkt über den Namen ableiten kann.

Unter "Spalten" können die Tabellen-Spalten zugeordnet werden. In diesem Beispiel muss die Spalte "TemplateFile" hinzugefügt werden. Das Pop-Up muss geschlossen werden. Das Speichern erfolgt im Anschluss durch das Speichern der Tabelle über das Disketten-Symbol.

DESKTOP-10PUD4K\S...ds\_Pos\_Definition

Spaltenname	Datentyp	NULL-Werte...
ID	bigint	<input type="checkbox"/>
TemplateFile	nvarchar(50)	<input type="checkbox"/>

**Null-Werte (nicht) zulassen**

Indizes/Schlüssel

Primärschlüssel/eindeutiger Schlüssel oder Index (ausgewählt):

PK\_Fields\_Pos\_Definition\_1

Eigenschaften für Primärschlüssel/eindeutiger Schlüssel oder Index

ist eindeutig: ☒ Ja

Spalten: Licence (ASC), TechFieldName (ASC)

Index: Spalten

Geben Sie die Spalten und die Sortierreihenfolge für diesen Index an:

Spaltenname	Sortierreihenfolge
TechFieldName	Aufsteigend
TemplateFile	Aufsteigend

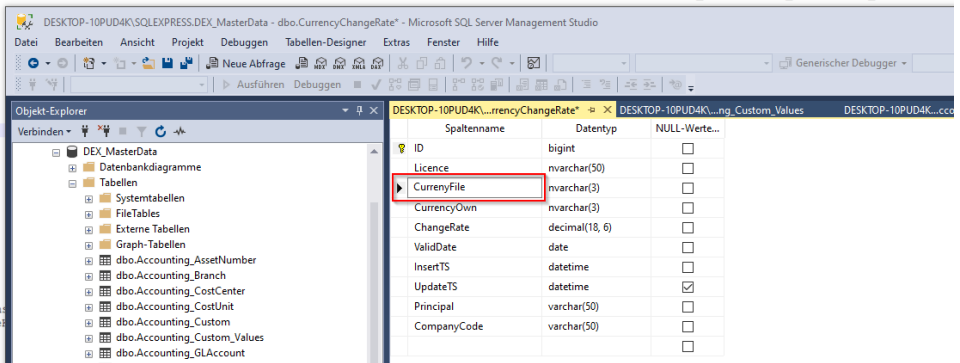
```
77 [2020.11.04 15:02:55][INFO] [lang_activelanguages][DateFormat] divergent 'IS NULLABLE' .. standard:NO != custom:YES
78 [2020.11.04 15:02:55][ERROR] [lang_activelanguages][DateFormat] could not alter nullable: [INFO] db_TableObj(DEX_Workflow, lang_activelanguages)
79 [INFO][MSSQL] USE Dex_Workflow SELECT TABLE_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE (TABLE_CATALOG='Dex_Workflow') AND (TABLE_NAME='lang_activelanguages')
80 [INFO] function alterNullable({
81   "COLUMN_NAME": "DateFormat",
82   "ORDINAL_POSITION": 7,
83   "DATA_TYPE": "varchar",
84   "TYPE_INTERNAL": "varchar",
85   "COLUMN_DEFAULT": "'dd.mm.yyyy'",
86   "IS_NULLABLE": "NO",
87   "CHARACTER_MAXIMUM_LENGTH": 50,
88   "NUMERIC_PRECISION": null,
89   "NUMERIC_SCALE": null
90 }) .. this.UseSQLType(MSSQL)
91 [ERROR] Fehler bei der Ausführung eines SQL-Statements
92 SQL error in 'ALTER TABLE Dex_Workflow.dbo.lang_activelanguages ALTER COLUMN DateFormat nvarchar(50) NOT NULL':
93 ReturnCode: -1
94 SqlState: 37000
95 NativeError: 5074
96 ErrorMsg: [Microsoft][SQL Server Native Client 11.0][SQL Server]Objekt 'DF_Lang_ActiveLanguages_DateFormat' ist von Spalte 'DateFormat' abhängig.
```

Die Spalte muss manuell über die Tabellen-Konfiguration geändert werden. Wenn NULL-Werte nicht mehr zugelassen werden, dann darf es in der Tabelle keinen Eintrag mit NULL geben. Das sollte in solchen Fällen immer gegeben sein, denn sonst würde die Spalte nicht entsprechend umgestellt werden. Sollte es dennoch Einträge mit NULL geben, muss ggf. manuell angepasst werden.

## Spalten hinzufügen

In wenigen Fällen können Spalten nicht automatisch hinzugefügt werden. Im folgenden Beispiel war ein Schreibfehler bei einem Spaltennamen enthalten, wodurch gleichzeitig eine Abweichung beim Unique-Key entdeckt wird. Durch die Korrektur des Spaltennamens werden automatisch 2 Fehler behoben.

```
153 [2020.11.04 15:02:55][ERROR] [currencychangerate] could not add standard column 'CurrencyFile': [INFO] db_TableObj(DEX_MasterData, currencychangerate)
154 [INFO][MSSQL] USE Dex_MasterData SELECT TABLE_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE (TABLE_CATALOG='Dex_MasterData') AND (TABLE_NAME='currencychangerate')
155 [INFO] function addColumn(
156   "COLUMN_NAME": "CurrencyFile",
157   "ORDINAL_POSITION": 5,
158   "DATA_TYPE": "nvarchar",
159   "TYPE_INTERNAL": "nvarchar",
160   "COLUMN_DEFAULT": null,
161   "IS_NULLABLE": "NO",
162   "CHARACTER_MAXIMUM_LENGTH": 3,
163   "NUMERIC_PRECISION": null,
164   "NUMERIC_SCALE": null
165 ) .. this.UseSQLType(MSSQL)
166 [ERROR] Fehler bei der Ausführung eines SQL-Statements
167 SQL error in 'ALTER TABLE Dex_MasterData.dbo.currencychangerate ADD CurrencyFile nvarchar(3) NOT NULL':
168 Error while retrieving Error Data
169 [INFO] function addColumn() .. return false
170
171 [2020.11.04 15:02:55][ERROR] [currencychangerate] function compareTableKeys( standardKeysObj, customKeysObj, currencychangerate ) .. missing specific key 'licence_currencyfile_currencyown_validate'
172 .. standardKeysObj[uniqueKeys]: {
173   {
174     "desc": "Licence_CurrencyFile_CurrencyOwn_ValidDate",
175     "columns": [
176       "CurrencyFile",
177       "CurrencyOwn",
178       "Licence",
179       "ValidDate"
180     ]
181   }
182 }
183 .. customKeysObj[uniqueKeys]: {
184   {
185     "desc": "Unique_CurrencyChangeRate",
186     "columns": [
187       "CurrencyOwn",
188       "CurrencyFile",
189       "Licence",
190       "ValidDate"
191     ]
192   }
193 }
194 [2020.11.04 15:02:55][INFO] [customer] function compareColumns
195 [2020.11.04 15:02:55][ERROR] [customer] function compareTableKeys
196 .. standardKeysObj[uniqueKeys]: {
197   {
198     "desc": "Licence_Principal_CompanyCode_CustomerID"
```



Nachdem alle Anpassungen vorgenommen wurden sollte das Skript erneut ausgeführt werden und das Log sollte erneut kontrolliert werden. Erst wenn keine relevanten Abweichungen mehr entdeckt werden, sind die Anpassungen abgeschlossen.

Revision #9

Created 4 November 2020 10:00:44 by Markus Meisner

Updated 17 July 2024 07:41:16 by Markus Meisner