

byron

Byron Informatik AG
Lohweg 6
CH-4054 Basel
Tel. +41 (0)61 690 96 00
Fax +41 (0)61 690 96 09

byron@byron.ch
www.byron.ch

Byron/BIS mit SQL-Datenbanken

Inhaltsverzeichnis

1	Einleitung	6
2	Anmerkungen zu den verschiedenen Datenbanksystemen	6
2.1	Microsoft SQL Server	6
2.1.1	Datenbank	6
2.1.2	Datenbankbenutzer	6
2.1.3	Vorbereitungen auf dem Byron/BIS-Client.....	6
2.1.4	Connection String.....	7
2.1.5	Verschlüsselung der Verbindung zum Server.....	7
2.1.6	Wichtig: Transaktionen.....	7
2.1.7	Pflege der Datenbank.....	8
2.2	Oracle	11
2.3	MySQL.....	11
3	Konfiguration der Datenbankschnittstelle	12
4	Änderung der SQL-Datenbankstruktur durch Byron/BIS	16
5	Erzeugen einer Byron/BIS Datenbank mit DbSetup	16
6	Erzeugen einer Byron/BIS-Datenbank mit BisRestore	17
7	Migration Byron/BIS ObjectStore => Byron/BIS SQL	19
7.1	Backup der Demodatenbank	19
7.2	Anlegen einer Anmeldung in MS SQL Server	20
7.3	Anlegen und Konfigurieren einer Datenbank in MS SQL Server	21
7.4	Anpassung des Modells	24
7.5	Einlesen des Backups in die MS SQL Server Datenbank.....	24
7.6	Verwenden der MS SQL Server Datenbank in Byron/BIS	24
8	FAQ.....	26
8.1	Laufzeitfehler: «Der dbo.IdTable-Datentyp ... wurde nicht gefunden»	26
8.2	SQL-Fehler beim Ändern einer Indexierung eines Attributs.	27
8.3	Wird <i>Microsoft SQL AlwaysOn Cluster</i> unterstützt?	27
8.4	Fehler bei BisRestore	28
8.5	Welche Änderungen in Byron/BIS lösen einen «Modelländerungsfehler» aus?	28
8.6	Laufzeitfehler: «Object 'TransactionBegin' is not a sequence object».....	28
8.7	Fehler BisRestore – Zugriffsverletzung.....	30
8.8	Abweichendes Datenbank-Schema (dbo)	30
8.9	Fehler «Eine vorhandene Verbindung wurde vom Remotehost geschlossen»	30

8.10	Welche Datenbank-Indices sollen erzeugt werden?	30
------	---	----

Version History

Version	Datum	Autor	Beschreibung
2.0	27.02.2018	R. Helbing	Dokument mit Berichtsvorlage neu aufgesetzt
2.1	27.02.2018	R. Helbing	Erste Schritte mit MS SQL Server ergänzt
2.2	29.03.2018	R. Helbing	Kapitel 7.1 (FAQ) ergänzt.
2.3	03.07.2018	R. Helbing	Benötigte Zugriffsrechte im Kapitel Datenbankbenutzer ergänzt (dbo.IdTable).
2.4	08.05.2019	R. Helbing	Parameter SQLQueryIterations in Kapitel Konfiguration der Datenbankschnittstelle ergänzt.
3.0	23.02.2020	R. Helbing	Oracle und MySQL nicht mehr unterstützt. Transaction Isolation Level konfigurierbar für Schreib- und Lese-Transaktion. Wartungsfenster konfigurierbar. Datenbankfehler nach Modelländerungen verhindert. Vorgaben für <i>MaxChildrenCached</i> und <i>MaxChildrenStored</i> geändert (nur \geq v5.4.7)
3.1	05.06.2020	R. Helbing	Konfiguration von verschlüsselten Verbindungen zum SQL Server ergänzt
3.2	15.06.2020	R. Helbing	Verschlüsselung der Verbindung zum Server überarbeitet.
3.3	01.12.2020	R. Helbing	Kapitel 7 umbenannt. Kapitel 7.4 Anpassung des Modells ergänzt
3.4	12.04.2021	R. Helbing	FAQ zu Microsoft SQL AlwaysOn Cluster ergänzt
3.5	22.04.2021	R. Helbing	FAQ zu einem Fehler mit BIS-Restore hinzugefügt.
3.6	03.05.2021	R. Helbing	FAQ Welche Änderungen in Byron/BIS lösen einen «Modelländerungsfehler» aus? hinzugefügt
3.7	03.06.2022	R. Helbing	FAQ Laufzeitfehler: «Object 'TransactionBegin' is not a sequence object» hinzugefügt
3.8	11.04.2023	R. Helbing	Konfiguration der Datenbankschnittstelle um <i>Statistics</i> ergänzt
3.9	04.05.2023	R. Helbing	Kapitel Connection String erweitert
3.10	05.09.2023	R. Helbing	Kapitel Indices zum Optimieren der Objekthistorie ergänzt
3.11	02.01.2024	R. Helbing	Kapitel Fehler «Eine vorhandene Verbindung wurde vom Remotehost geschlossen» ergänzt
3.12	09.01.2024	R. Queloz	Hinweise auf SQL Server Native Client ergänzt.
3.13	22.01.2024	R. Helbing	Konfiguration der Datenbankschnittstelle um <i>ForceSeek</i> , <i>LogStatement</i> und <i>OptimizeUnknown</i> ergänzt. Kapitel Welche Datenbank-Indices sollen erzeugt werden? ergänzt.
3.14	31.01.2024	R. Helbing	Konfiguration der Datenbankschnittstelle um <i>NavigationCache</i> , <i>NavigationHint</i> und <i>FixesAssoc</i> ergänzt.

3.15	06.03.2024	R. Helbing	Korrektur eines Skripts in Bereinigen der Tabellen, welche die Synchronisation der Clients unterstützen
3.16	30.04.2025	R. Helbing	Ergänzungen in Welche Datenbank-Indices sollen erzeugt werden?
3.17	15.05.2025	R. Helbing	Konfiguration der Datenbankschnittstelle um <i>AssociatedLimit</i> und <i>FindIndexedLimit</i> ergänzt.

1 Einleitung

Im Folgenden werden die Schritte beschrieben, welche nötig sind um den Betrieb einer Byron/BIS-Datenbank auf einem relationalen Datenbankserver vorzubereiten.

Nach Abschluss der Vorbereitungen kann die Datenbank entweder von Grund auf über [DbSetup](#) aufgebaut (Einlesen von Modulen) oder die Datenbank mit Hilfe von [BisRestore](#) aus einer Sicherungskopie eingelesen werden.

2 Anmerkungen zu den verschiedenen Datenbanksystemen

2.1 Microsoft SQL Server

2.1.1 Datenbank

Es muss eine neue SQL Server Datenbank angelegt werden. Vgl. auch [Transaktionen](#).

Es wird empfohlen, ein einfaches Wiederherstellungsmodell (simple recovery model) zu wählen, da sonst die Grösse des Logfiles zu stark anwächst. Falls die IT-Abteilung sich im Rahmen des Backups um die Reduzierung der Logfilegrösse kümmert, haben die Vorgaben der IT natürlich Vorrang.

2.1.2 Datenbankbenutzer

Für den Zugriff auf die Datenbank benötigt Byron/BIS einen Datenbankbenutzer. Dieser kann sich über Windows anmelden, wenn der SQL Server entsprechend konfiguriert wurde (Windows-Authentifizierung erlaubt) oder ein SQL-Server Benutzer, der sich über Name, Kennwort anmeldet. Hierzu ist der zuständige Datenbank-Administrator zu befragen.

Der Datenbankbenutzer eines *normalen* Byron/BIS Benutzers muss die Berechtigung besitzen, alle Tabellen der Datenbank zu verändern. **Zusätzlich** benötigt er das Recht den von Byron/BIS-SQL erzeugten und verwendeten **Tabellentyp `dbo.IdTable`** zu verwenden (Execute).

Der Datenbankbenutzer, welcher für das Aufbauen der Datenbank beziehungsweise für Byron/BIS - Modelländerungen verwendet wird, muss zusätzlich die Berechtigung besitzen, Tabellen der Datenbank zu erzeugen und zu löschen.

Der Datenbankbenutzer und dessen Kennwort werden in DbSetup zusammen mit dem Connection String eingetragen um einen Byron/BIS-Datenbankalias zu bilden.

2.1.3 Vorbereitungen auf dem Byron/BIS-Client

Auf dem Byron/BIS-Client (d.h. auch der Server auf welchem der BisService und die XDS-Skripte laufen) muss einen zum SQL-Server passende Client-Software sein. Diese Software (Provider, resp. Driver) wird dann im ConnectionString für die Verwendung spezifiziert.

Siehe auch: <https://learn.microsoft.com/en-us/sql/connect/odbc/download-odbc-driver-for-sql-server>

Es ist darauf zu achten, ob ein 32bit- oder 64bit-Paket benötigt wird (abhängig vom Client).

In allen Byron/BIS-Versionen ab dem 10.1.2024 ist der *ODBC Driver 17 for SQL Server* Bestandteil der Installation.

Die Verwendung des *SQL Server 2012 Native Client* (SNAC) wird nicht empfohlen, da bereits EOL.

2.1.4 Connection String

Wird Windows-Authentifizierung verwendet, dann muss der Connection String um `Trusted_Connection=true;` oder `Integrated Security=true;` ergänzt werden (vgl Datenbankbenutzer). Benutzername und Kennwort müssen in diesem Fall in `DbSetup` nicht angegeben werden.

Beispiele:

```
SQL Server mit ODBC (Byron/BIS >= v5.7.2) – empfohlen
Driver={ODBC Driver nn for SQL Server};
Server=DbServerTest;Database=ByronBISTest;
N.B. nn wird unterstützt für die Werte 13, 14, 15, 16, 17, 18
Siehe auch hier

SQL Server 2012 - veraltet
Provider=SQLNCLI11;Server=DbServer1;Database=ByronDatabase;

SQL Server 2008 - veraltet
Provider=SQLNCLI10;Server=DbServer1;Database=ByronDatabase;

SQL Server allgemein - nicht empfohlen
Provider=SQLOLEDB;Server=DbServerTest;Database=ByronBISTest;

SQL Server 2016 mit ODBC (Byron/BIS >= v5.2.0.3101)
Driver={ODBC Driver 13 for SQL Server};
Server=DbServerTest;Database=ByronBISTest;
```

Siehe auch [Connection String Reference](#). Eine Liste [aller Connection String-Schlüsselwörter](#).

Achtung: die dokumentierte Einstellung `«Connection Timeout=60;»` ist **falsch**. Richtig ist `«Connect Timeout=60;»`. Weiter hat die `Connect Timeout` Einstellung als [Randbedingung](#) die TCP/IP-Timeouts und kann deswegen nicht beliebig klein gewählt werden.

Die Konfiguration von Byron-BIS Einstellungen über den Connection String ist [unten](#) beschrieben.

Achtung: Wird ein Connection String mit `«Driver={SQL Server Native Client 11.0};»` ([SQL Server Native Client 11.0 ODBC Driver](#)) statt `«Provider=SQLNCLI11»` verwendet, dann kann `«Integrated Security=SSPI;»` nicht verwendet werden. Stattdessen muss die Windows Authentifizierung mit `«Trusted_Connection=yes»` konfiguriert werden.

2.1.5 Verschlüsselung der Verbindung zum Server

Um Verbindungen zum SQL-Server verschlüsseln zu können, muss dieser entsprechend konfiguriert sein. Diese [Microsoft-Dokumentation](#) beschreibt das dazu nötige Vorgehen. und dieser [Blog](#) bietet eine verständliche Übersicht zu diesem Thema.

2.1.6 Wichtig: Transaktionen

Transaktionen auf dem SQL Server setzen voraus, dass für die Datenbank **Snapshot Isolation aktiviert** ist:

```
ALTER DATABASE ByronIntern SET ALLOW_SNAPSHOT_ISOLATION ON;
```

Die einzelnen Transaktionen werden mit dem Isolation Level Snapshot ausgeführt, sofern nicht `Serializable` konfiguriert ist (vgl. unten).

Lesetransaktionen erhalten eine höhere Priorität als Schreibtransaktionen:

```
SET DEADLOCK_PRIORITY HIGH    für Lesetransaktionen
SET DEADLOCK_PRIORITY LOW    für Schreibtransaktionen
```

2.1.7 Pflege der Datenbank

Backup

Ein regelmässiges Backup (ein- bis zweimal täglich) wird empfohlen. Backups sollten nach den Vorgaben der IT erstellt werden. Wenn solche nicht vorhanden sind: 1x täglich (nachts) ein vollständiges Backup und 1x mal täglich (ca. 13 Uhr) ein differenzielles Backup.

```
BACKUP DATABASE [ByronBIS] TO DISK = N'C:\...\MSSQL_Sicherung_ByronBIS.bak'
WITH NOFORMAT, INIT, NAME = N'ByronBIS-Vollständig Datenbank Sichern', SKIP, NOREWIND, NOUNLOAD,
STATS = 10;
```

Bzw.

```
BACKUP DATABASE [ByronBIS] TO DISK = N'C:\...\MSSQL_Sicherung_ByronBIS.bak'
WITH DIFFERENTIAL, NAME = N'ByronBIS mittags differential'
```

Achtung: Backup beinhaltet **nicht alle** Informationen, welche sich in der Datenbank befinden. Stored Procedures und [Tabellentypen](#) werden z.B. nicht gespeichert. Siehe auch [hier](#).

Optimieren aller Indices

Die Indices der Datenbank sollten regelmässig (1 x täglich bis 1 x wöchentlich) optimiert – d.h. neu erstellt – werden. Folgendes (aus dem Internet bezogene) Skript erfüllt diese Aufgabe für alle Tabellen der Datenbank:

```
DECLARE @Table VARCHAR(255)
DECLARE @cmd VARCHAR(500)

SET QUOTED_IDENTIFIER ON

SET @cmd = 'DECLARE TableCursor CURSOR FOR
SELECT CONCAT(''[', TABLE_SCHEMA, '].[', table_name, ']'') as tableName
FROM INFORMATION_SCHEMA.TABLES
WHERE table_type = ''BASE TABLE''

-- create table cursor
EXEC (@cmd)
OPEN TableCursor

FETCH NEXT FROM TableCursor INTO @Table
WHILE @@FETCH_STATUS = 0
BEGIN

    SET @cmd = 'ALTER INDEX ALL ON ' + @Table + ' REBUILD WITH (FILLFACTOR = 95)'
    EXEC (@cmd)

    FETCH NEXT FROM TableCursor INTO @Table
END

CLOSE TableCursor
DEALLOCATE TableCursor
```

Bereinigen der Tabellen, welche die Synchronisation der Clients unterstützen

Die Tabellen *Modifications*, *TransactionBegin* und *TransactionEnd* wachsen mit jeder Update-Transaktion. Sie müssen regelmässig (1x täglich bis 1x wöchentlich) bereinigt werden.

Achtung: die Tabellen **dürfen nicht komplett geleert werden** (z.B. mit TRUNCATE TABLE), da sonst die Aktualisierung der Clientcaches nicht gewährleistet ist.

Folgendes Skript erfüllt diese Aufgabe:

```
use [Database]
go
delete from [dbo].[Modifications]
where [tid] < (select MAX([tid]) - 1000 from [dbo].[Modifications]);
go
delete from [dbo].[TransactionBegin]
where [tid] < (select MAX([tid]) - 100 from [dbo].[TransactionBegin]);
go
delete from [dbo].[TransactionEnd]
where [nr] < (select MAX([nr]) - 100 from [dbo].[TransactionEnd]);
go
```

Alternatives Skript, welches zu weniger Db-Blockaden führt:

```
use [Database]
go
WHILE (select count(DISTINCT [tid]) from [dbo].[Modifications]) > 1500
BEGIN
    delete [dbo].[Modifications]
    where [tid] < (select min([tid]) from [dbo].[Modifications])+20000
        and [tid] < (select max([tid]) from [dbo].[Modifications])-1000
    waitfor delay '00:00:02'
END
Go

WHILE (select count(DISTINCT [nr]) from [dbo].[TransactionEnd]) > 1500
BEGIN
    delete [dbo].[TransactionEnd]
    where [nr] < (select min([nr]) from [dbo].[TransactionEnd])+20000
        and [nr] < (select max([nr]) from [dbo].[TransactionEnd])-1000
    waitfor delay '00:00:02'
END
go
```

N.B.: Das Löschen von Einträgen aus [TransactionBegin] ist bei SQL Server >= 2012 und bei Datenbanken, die mit einer Version >= 5.2.0.3001 erzeugt wurden, nicht mehr notwendig, da die Tabelle durch eine SEQUENCE ersetzt wurde.

Wechsel von der Tabelle TransactionBegin zu SEQUENCE TransactionBegin:

1. System herunterfahren (alle Prozesse anhalten / Datenbank in SingleUser-Betrieb nehmen)
2. Datenbank anpassen

```
DROP TABLE TransactionBegin;
CREATE SEQUENCE TransactionBegin AS bigint START WITH 1;
```

```
DELETE FROM Modifications;  
DELETE FROM TransactionEnd
```

3. System hochfahren

2.2 Oracle

Oracle wird nicht unterstützt.

2.3 MySQL

MySQL wird nicht unterstützt.

3 Konfiguration der Datenbankschnittstelle

Die Datenbankschnittstelle von Byron/BIS kann mit Hilfe des Connection Strings konfiguriert werden. Dazu werden die möglichen Parameter im Connection String aufgeführt. Beispiel:

Provider=SQLNCLI11;Server=byrondb;Database=ByronIntern;SQLQueryTimeout=30;

Parameter	Vorgabe	Beschreibung
AssociatedLimit (ab v5.9.2)	100	Für Performance-Analysen. Legt fest, ab welcher Anzahl Objekte das Navigieren über eine Assoziation eine Logmeldung (Information) erzeugt.
CacheReduceInterval	40'000 (20'000 vor v5.2.2)	Der Objektcache der Datenbankschnittstelle wird regelmässig überprüft (welche Objekte können entfernt werden etc.). Dieser Parameter gibt in Anzahl Objektzuweisungen an, wie oft die Überprüfung stattfinden soll.
CacheSizeTarget	100'000	Zielgrösse für den Objektcache der Datenbankschnittstelle.
DirtyModelAllowed	0	Wenn nicht gesetzt (=0), dann wird nach Modelländerungen durch einen anderen Prozess verhindert, dass irgendwelche Änderungen auf der Datenbank vorgenommen werden. Wenn gesetzt (=1), dann werden nur Modelländerungen verhindert.
FindIndexedLimit (ab v5.9.2)	1000	Für Performance-Analysen. Legt fest, ab welcher Grösse das Ergebnis einer Indexsuche eine Logmeldung (Information) erzeugt.
FixesAssoc (ab v5.8.0)	1000, 250	Legt fest, wie in der Funktion «Fehler suchen» Assoziationen behandelt werden. Der erste Wert legt fest, bis zu welcher Anzahl Zielobjekte die Assoziation geprüft wird. Der zweite Wert legt fest, ab welcher Anzahl eine Warnung ausgegeben wird, welche darauf hinweist, dass die Assoziation indexiert werden soll. Soll nur geprüft werden, welche Assoziationen indexiert werden können, dann sollte die Einstellung 0, 100 sein.
ForceSeek (ab v5.8.0)	1	Legt fest, für welche Tabellen der Table-Hint FORCESEEK verwendet wird. Die einzelnen Werte sind Flags und können mit Addition kombiniert werden. Einzelne Werte: 0 keine Tabelle 1 Section_0 2 Index_S0_Assoc 4 Index_S0_I* bei der Suche nach Guid (Value) 8 Index_S0_I* bei der Suche nach dem Wert (Key)

		16 FORCESSEEK wird beim Iterieren über die Datenbank verwendet Die Vorgabe umfasst die Tabelle Section_0.
IgnoreBISTransactions	0	Ignoriert die durch den Byron/BIS-Kern gesteuerten Transaktionen, um den Datenbankserver zu entlasten. Nur für BISRestore (oder «Fehler suchen») verwenden!
IndexCache (ab v5.6.1)	100, 25 (bis v5.7.6) 250, 1000 (ab v5.8.0)	Legt das Verhalten des Cache für Index- und Instanzabfragen («START INSTANCES») fest. Der Cache optimiert identische Abfragen innerhalb einer Transaktion. Die erste Zahl legt die Anzahl der gespeicherten Abfragen fest (0 = Cache ausgeschaltet). Die zweite Zahl legt die maximale Anzahl von Ergebnissen fest, die eine Abfrage haben darf, wenn sie im Cache gespeichert werden soll.
IsolationLevel	0,1	Transaction Isolation Level für Schreib- und Lesetransaktionen. Die Konfiguration für die Schreibtransaktion kommt zuerst, danach die der Lesetransaktion 0= <i>SERIALIZABLE</i> 1= <i>SNAPSHOT</i> 2= <i>REPEATABLE READ</i> 3= <i>READ COMMITTED</i> <i>SERIALIZABLE</i> wird benötigt bei Modelländerungen, welche einen Index modifizieren. Innerhalb eines Wartungsfensters werden Transaktionen statt mit <i>SNAPSHOT</i> Isolation mit <i>SERIALIZABLE</i> durchgeführt.
LogStatements (ab v5.8.0)	0	Legt fest, ob alle SQL-Befehle ins Log geschrieben werden sollen. Die Ausgabe ins Log erfolgt mit Level «Information» bzw. 3. Wird mit dem Wert 1 aktiviert.
MaintenanceWindows		Angabe, an welchen Wochentagen und zu welchen Zeiten «Wartungsfenster» vorgesehen sind. Innerhalb eines Wartungsfensters wird der Transaction Isolation Level <i>SNAPSHOT</i> nicht verwendet, da dies zu Abbrüchen führen kann. Die Wartungsfenster werden konfiguriert mit dem Schema all = Tag {"", "Tag} " " Zeit "-" Zeit. Tag = 0 1 2 3 4 5 6. //Wochentag 0 = Montag Beispiel: 2, 6 01:30-02:15 Mittwoch und Sonntag zwischen 01:30 und 02:15
MaxChildrenCached (bis v5.7.6)	5000	Gibt an, wie viele Unterobjekte am Objekt zwischengespeichert (gecached) werden. Werden die Unterobjekte eines Objekts geladen, dann wird das Ergebnis am Objekt zwischengespeichert, sofern diese Anzahl nicht überschritten wird.

MaxChildrenStored (bis v5.7.6)	0	Gibt an, wie viele Unterobjekte zusätzlich am Objekt selbst gespeichert werden. Überschreitet die Anzahl der Unterobjekte diesen Wert, dann werden die Unterobjekte nicht mehr am Objekt gespeichert.
NavigationCache (ab v5.8.0)	10000, 5000	Legt das Verhalten des Cache für Navigationen («VIA») fest. Der Cache optimiert identische Abfragen innerhalb einer Transaktion. Die erste Zahl legt die Anzahl der gespeicherten Abfragen fest (0 = Cache ausgeschaltet). Die zweite Zahl legt die maximale Anzahl von Ergebnissen fest, die eine Abfrage haben darf, wenn sie im Cache gespeichert werden soll.
NavigationHint (ab v5.8.0)	7	Legt fest, welche Abfragen an die Datenbank dazu führen, dass die gesuchten Objekte direkt geladen werden, oder ob zuerst die Ids geladen werden. Die einzelnen Werte sind Flags und können mit Addition kombiniert werden. Einzelne Werte: 0 Keine Abfragen 1 Navigation zu den enthaltenen Objekten 2 Navigation über eine indizierte Assoziation 4 Indexierte Abfragen («START VALUE ...»)
NotificationWaitInterval	500	Wird in Byron/BIS auf Datenbank-Notifikationen gewartet, dann hat dies zu Folge, dass die Schnittstelle den Datenbankserver regelmässig abfragt. Dieser Parameter gibt in Millisekunden an, wie viel Zeit zwischen den einzelnen Abfragen gewartet wird.
OptimizeUnknown (ab v5.8.0)	0	Legt fest, wann an einen SELECT Befehl der Query-Hint OPTIMIZE FOR UNKNOWN angehängt werden soll. Die einzelnen Werte sind Flags und können mit Addition kombiniert werden. Einzelne Werte: 0 Query-Hint wird nie verwendet 1 Query-Hint wenn Section_0.Parent abgefragt wird. 2 Query-Hint wenn Index_S0_I* abgefragt wird. 4 Query-Hint wenn Index_S0_Assoc abgefragt wird. 8 Query-Hint wenn ITERATE verwendet wird. Diese Option sollte nur verwendet werden, wenn ForceSeek keinen Erfolg hat.
ReadOnlyConnectionCount	3	Anzahl Datenbankverbindungen, die für Lesetransaktionen im BisService verwendet werden.
SQLQueryIterations	2	Anzahl der Versuche eine SQL-Query auszuführen, wenn ein SQL-Command-Timeout auftritt. Vorgabe ist 2, d.h. es wird bei einem Timeout maximal ein (1) Mal wiederholt.

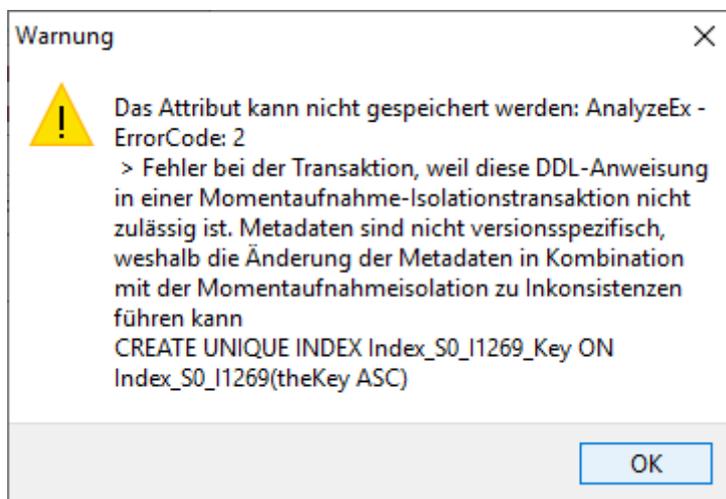
SQLQueryTimeout	30	Zeit in Sekunden, die ein SQL-Befehl dauern darf.
Statistics	0	Wenn gesetzt (= 1), dann werden im Log Informationen zur Verwendung der SQL-Datenbank (Anzahl übertragene Bytes bzw. Objekte etc.) angezeigt (mit Loglevel Information).
TryReconnect	1	Wenn gesetzt (= 1), dann versuchen die Byron/BIS-Programme bei SQL-Verbindungsunterbrüchen die Verbindung zum SQL-Server wiederherzustellen.

4 Änderung der SQL-Datenbankstruktur durch Byron/BIS

Verschiedene Byron/BIS-Prozesse ändern die Struktur der SQL-Datenbank. D.h. Tabellen und andere Datenbankobjekte werden ergänzt oder gelöscht. Dies geschieht im Wesentlichen beim Anlegen einer Datenbank (mit DbSetup oder BisRestore) oder bei Modelländerungen, bei denen ein Attribut-Index erzeugt oder gelöscht wird.

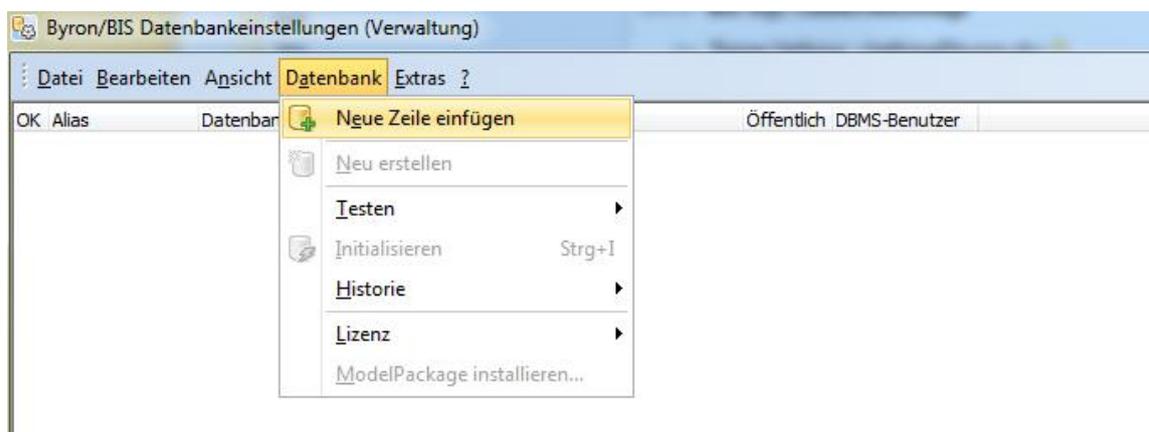
Da der SQL Server Änderungen der Datenbankstruktur innerhalb einer Transaktion mit dem Isolationslevel SNAPSHOT nicht unterstützt, muss für diese Aktionen [Serializable](#) konfiguriert werden, sofern das Byron/BIS-Programm (DbSetup, BISRestore) dies nicht bereits erledigt.

Wird die Konfiguration von Serializable vergessen, tritt folgender Fehler auf:

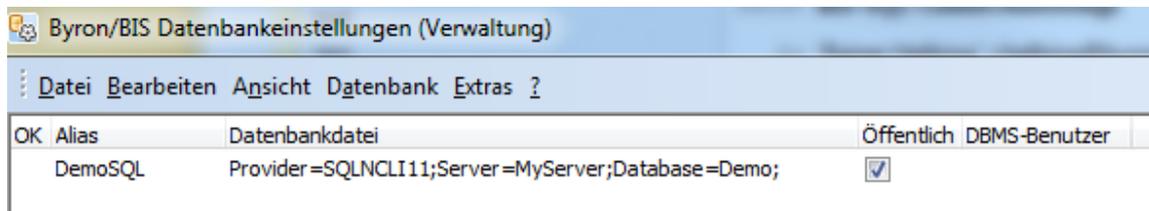


5 Erzeugen einer Byron/BIS Datenbank mit DbSetup

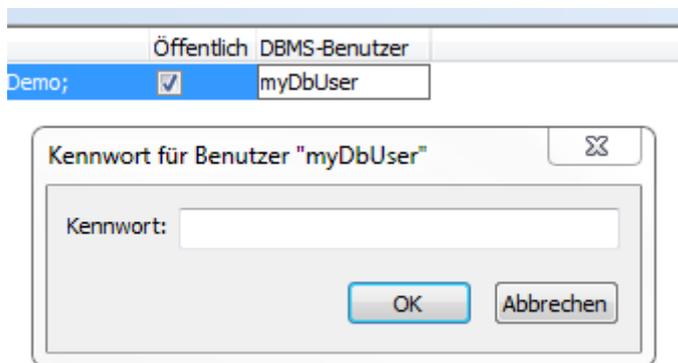
1. Starten Sie DbSetup und erzeugen Sie eine neue Zeile



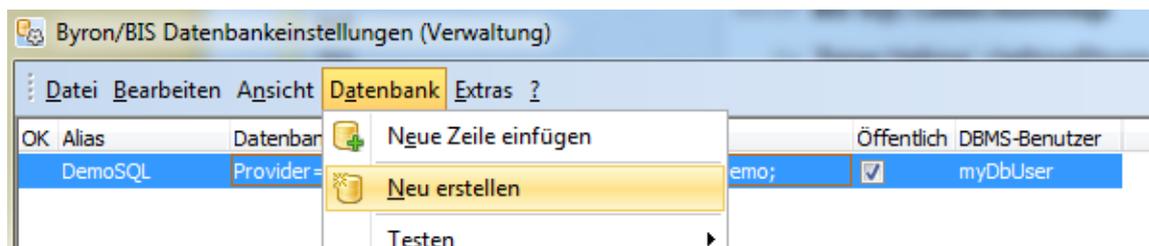
2. Ändern sie Alias und Connection String (Spalte Datenbankdatei)



3. Geben Sie den Datenbankbenutzer und dessen Kennwort ein



4. Selektieren Sie die neue Zeile und wählen Sie „Neu erstellen“ erstellen



Falls ein Fehler auftreten sollte, wiederholen Sie die Operation. Tritt der Fehler immer noch auf, dann wenden Sie sich an den Support.

5. Lesen Sie eine Lizenz in die Datenbank ein und initialisieren Sie die Datenbank.
6. Testen Sie den Zugriff auf die Datenbank mit *Strg-T* oder *Ctrl-T*.
7. Lesen Sie die gewünschten Byron/BIS-Module (BMO) in die Datenbank ein.
8. ...

DbSetup versucht, die Konfiguration von `Serializable` zu ergänzen, sofern [Änderungen des SQL-Datenbankstruktur](#) zu erwarten sind (Einlesen von BMO, Anlegen einer neuen Datenbank, ...).

6 Erzeugen einer Byron/BIS-Datenbank mit BisRestore

Mit BisRestore wird aus einer BisBackup erstellten Sicherungskopie eine lauffähige Byron/BIS Datenbank erstellt. Information zu BisBackup und BisRestore finden Sie hier: [ByronBIS_BackupRestore.pdf](#)

Statt eines Datenbankpfades muss BisRestore ein Connection String übergeben werden, welcher auch die Angaben zum Datenbankbenutzer und dessen Kennwort enthält.

Beispiel Connection String:

```
Provider=SQLNCLI11;Server=DbServer1;Database=ByronDatabase;  
User ID=myDbUser;Password=my-Db-Pwd33;IgnoreBISTransactions=1;
```

N.B. IgnoreBISTransactions=1; wird nur bei BISRestore benötigt, um die Grösse der SQL-Server-Transaktion zu begrenzen.

BISRestore wird beim Einlesen der Modellinformation (XSD-Datei) die Konfiguration von Serializable ergänzen, um [Fehler](#) zu vermeiden.

Ein Skript zum Erzeugen einer Datenbank könnte aussehen wie folgt:

```
REM Dateien der Sicherungskopie  
SET BACKUP_Files=D:\BackupDir\DemoBackup  
  
REM Connection String der MSSQL Datenbank  
SET TARGET_DB="Provider=SQLNCLI11;Server=DbServer1;Database=ByronDatabase;  
User ID=myDbUser;Password=my-Db-Pwd33;IgnoreBISTransactions=1;"  
  
SET EXEDIR=%BIS_ROOT%\br/>  
set BIS_DEBUG_LEVEL=T04  
  
REM Restore  
"%EXEDIR%BISRestore.exe" %TARGET_DB% "%BACKUP_Files%" /FastReferenceRestore /always  
  
PAUSE
```

7 Migration Byron/BIS ObjectStore => Byron/BIS SQL

In diesem Kapitel wird der Wechsel auf eine MS SQL Server Installation anhand der Demodatenbank gezeigt.

Wichtig: Das hier gezeigte Übernehmen des Modells und der Daten führt zu einem lauffähigen Byron/BIS-System, welches aber **nicht auf die besonderen Möglichkeiten des SQL Servers angepasst** ist. Zur Optimierung der Performance müssen Modell und Abfragen überarbeitet werden.

In den folgenden Unterkapiteln werden diese Schritte beschrieben:

1. Backup der Demo-Datenbank
2. Anlegen einer Anmeldung in MS SQL Server
3. Anlegen und Konfigurieren einer Datenbank in MS SQL Server
4. Einlesen des Backups in die MS SQL Server Datenbank
5. Verwenden der MS SQL Server Datenbank in Byron/BIS (DbSetup)

Es wird davon ausgegangen dass folgende Voraussetzungen erfüllt sind:

- MS SQL Server Version \geq 2012 ist installiert
- Ein passendes SQL Server Management Studio ist installiert
- Der [SQL Server Client](#) ist installiert (auf dem Computer mit Byron/BIS).

Bei der Verwendung der MS SQL Server Express Edition ist zu beachten, dass deren Verwendung [Beschränkungen](#) unterliegt (Version 2016):

- maximale Datenbankgrösse 10 GB
- maximal ein 1410 MB Arbeitsspeicher wird genutzt
- maximal ein 1 Prozessor wird genutzt

7.1 Backup der Demodatenbank

In der bestehenden Byron/BIS für ObjectStore Installation wird die Datenbank mit BisBackup in das Verzeichnis C:\temp gesichert. Das dazu verwendete CMD-Skript:

```
rem @Echo off
rem -----
rem Script for database backup with BisBackup
rem -----

rem Setting of the variables
rem Note the variable TEMP, BIS_ROOT and OS_ROOTDIR must already be set

rem Setting of database
set DATABASE="%ALLUSERSPROFILE%\ByronBIS\DemoDb\demo.bisdb"

rem Variables used for BISBackup
set TARGET=C:\temp\Demo

rem -----
rem Commands
set BISBACKUP="%BIS_ROOT%\BISBackup.exe"

rem set sizes large enough
set OS_AS_SIZE=0x60000000
set OS_CACHE_SIZE=0x08000000
```

```

rem -----
rem Database backup with BISBackup
echo.
echo. Backup Database with BISBackup...
echo.
"%BIS_ROOT%\BISLoader.exe" /Wait %BISBACKUP% %DATABASE% %TARGET%
/NoSharedAttributes /AddressSpaceControlLevel=1 /always

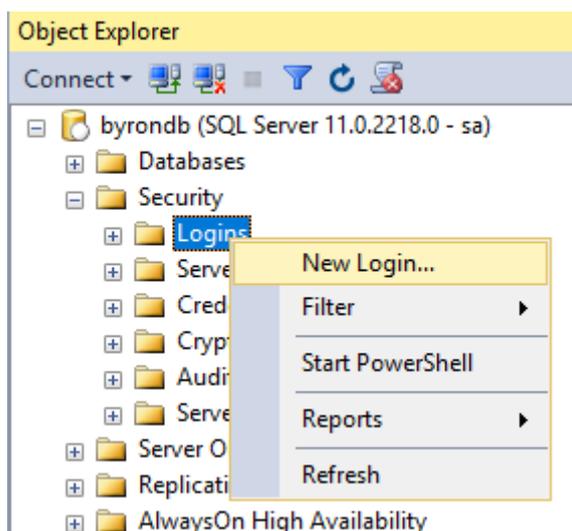
pause
rem end of script

```

7.2 Anlegen einer Anmeldung in MS SQL Server

Mit SQL Server Management Studio meldet man sich als Berechtigter (*Administrator* über Windows Authentifizierung oder *sa*) an der gewünschten MS SQL Server Instanz an.

Ein neuer Login / eine neue Anmeldung wird erzeugt



Folgende Einstellungen sind für diesen Fall ausreichend:

Login name: Search...

Windows authentication
 SQL Server authentication

Password:
Confirm password:

Specify old password
Old password:

Enforce password policy
 Enforce password expiration
 User must change password at next login

Mapped to certificate
 Mapped to asymmetric key
 Map to Credential Add

Mapped Credentials

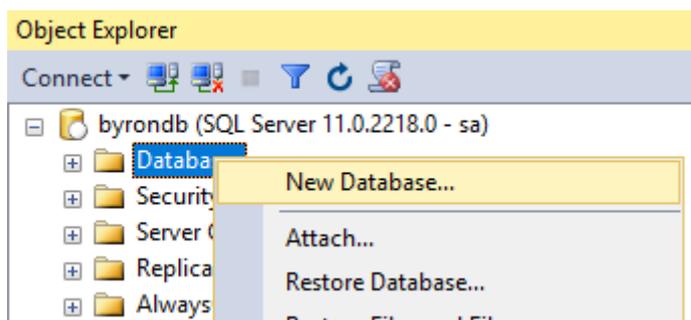
Credential	Provider
------------	----------

Remove

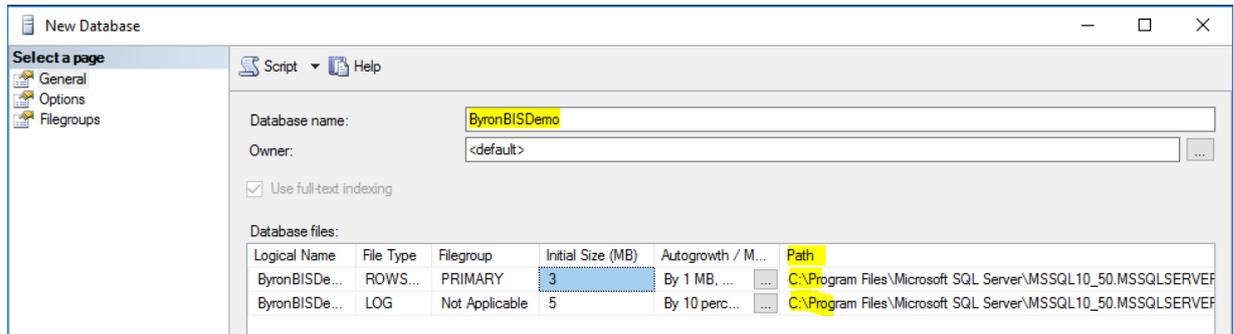
Default database:
Default language:

OK Cancel

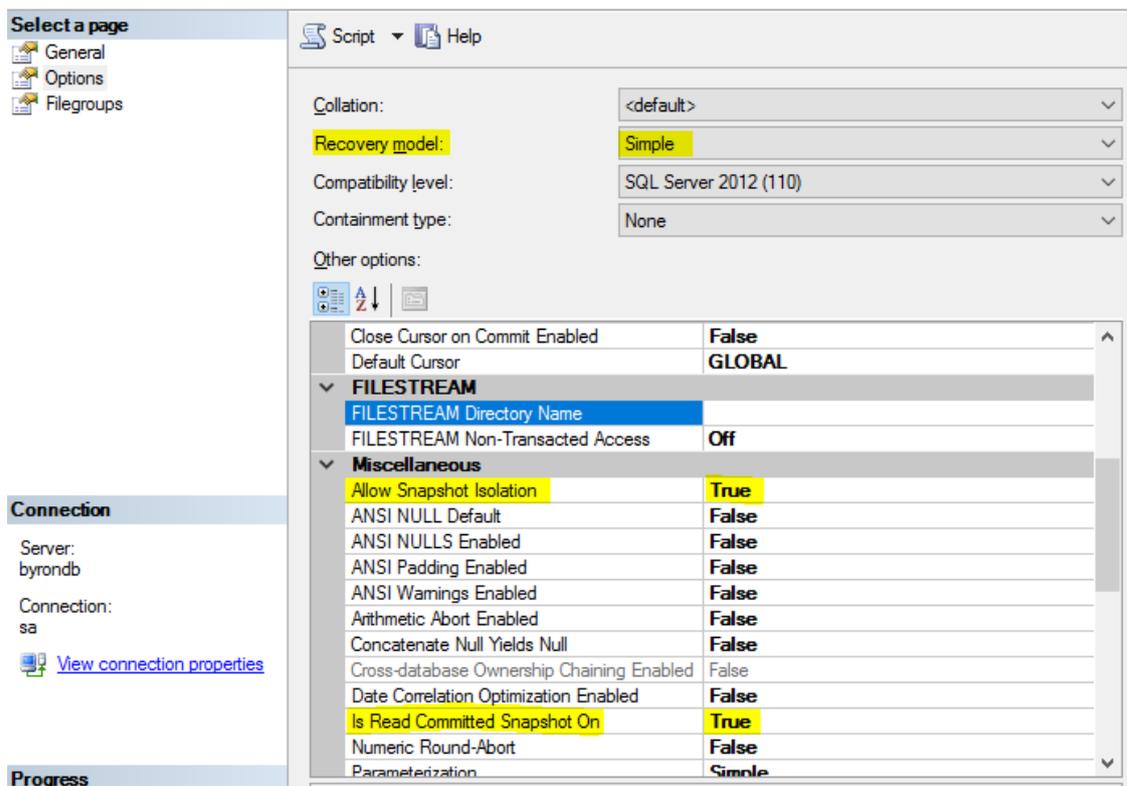
7.3 Anlegen und Konfigurieren einer Datenbank in MS SQL Server



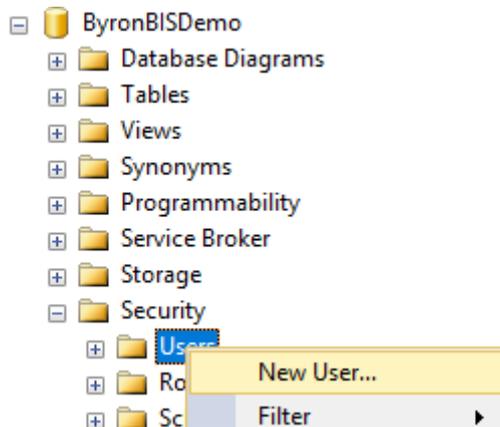
Auf der ersten Seite **muss** der Name der Datenbank angegeben werden. Der Pfad zu den Datenbankdateien kann bei Bedarf geändert werden.



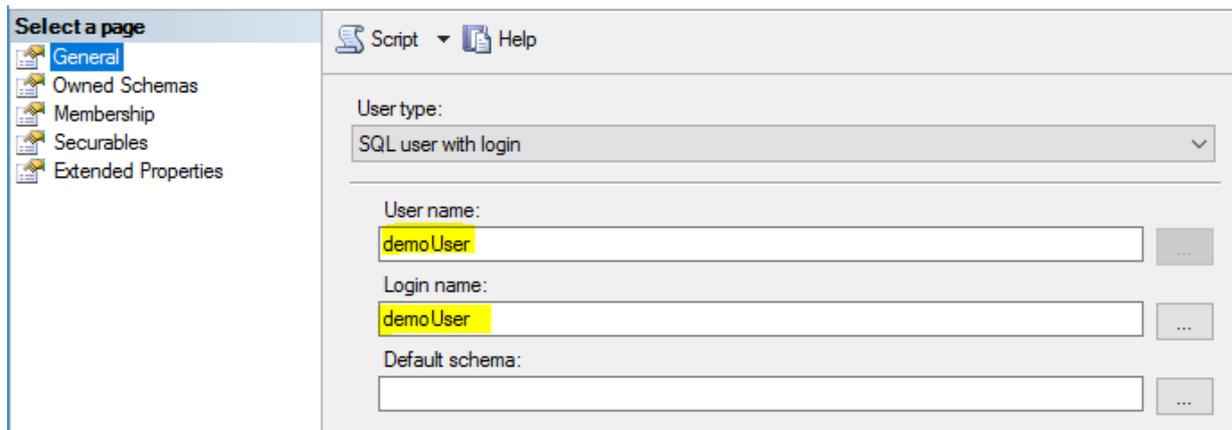
Auf der Seite Optionen **müssen** folgenden Einstellungen gemacht werden.



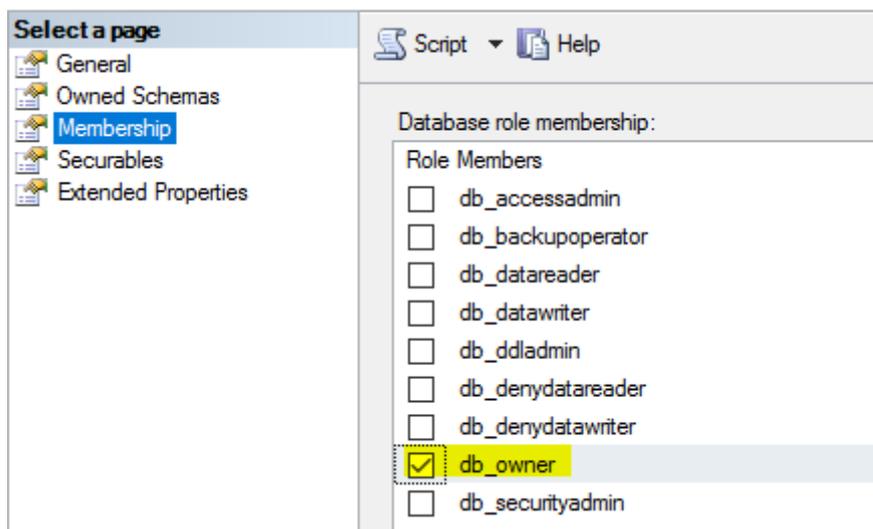
Nachdem die Datenbank angelegt ist, wird in der Datenbank ein Benutzer erzeugt, welcher zur bestehenden Anmeldung passt.



Folgende Eingaben müssen gemacht werden.



Der Einfachheit halber erhält der Benutzer «db_owner» Rechte



7.4 Anpassung des Modells

Ist beim Kunden *KeepAlive* in Betrieb, dann muss diese Funktionalität während der Migration entfernt werden:

- Das Attribut *bisB_KeepAliveTimeStamp* **löschen** (aus der Datenbank, nicht nur am User *entfernen*).

7.5 Einlesen des Backups in die MS SQL Server Datenbank

Auf einem Computer mit einer Byron/BIS für SQL Installation wird das zuvor erstellte Backup mit folgendem CMD-Skript eingelesen (vgl. [Erzeugen einer Byron/BIS-Datenbank mit BisRestore](#)):

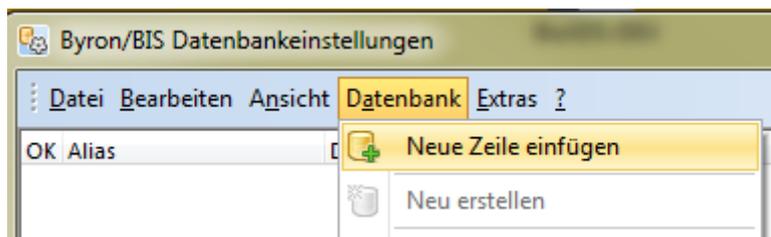
```
rem Speicherort des Backups
SET BACKUP_DB="C:\TEMP\Demo"

rem Connection string der SQL-Server Datenbank auf einer Zeile
SET TARGET_DB="Provider=SQLNCLI11;Server=byrondbdev\SQLEXPRESS;
                Database=ByronBISDemo;User ID=demoUser;Password=123456789;
                IgnoreBISTransactions=1;"

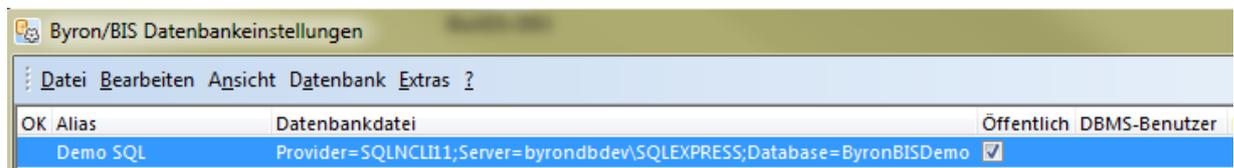
SET EXEDIR=%BIS_ROOT%\
"%EXEDIR%BISRestore.exe" %TARGET_DB% %BACKUP_DB% /always /FastReferenceRestore
```

7.6 Verwenden der MS SQL Server Datenbank in Byron/BIS

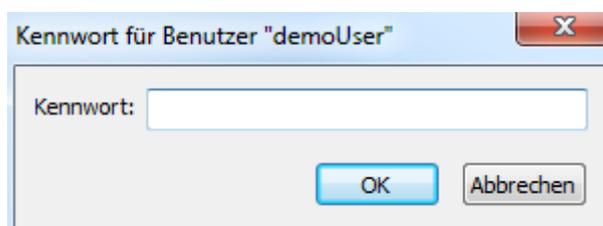
Auf einem Computer mit einer Byron/BIS für SQL Installation wird die Datenbank in DbSetup (Admin) eingetragen:



Die neue Zeile wird angepasst: «Alias», «Datenbankdatei» und «DBMS-Benutzer»

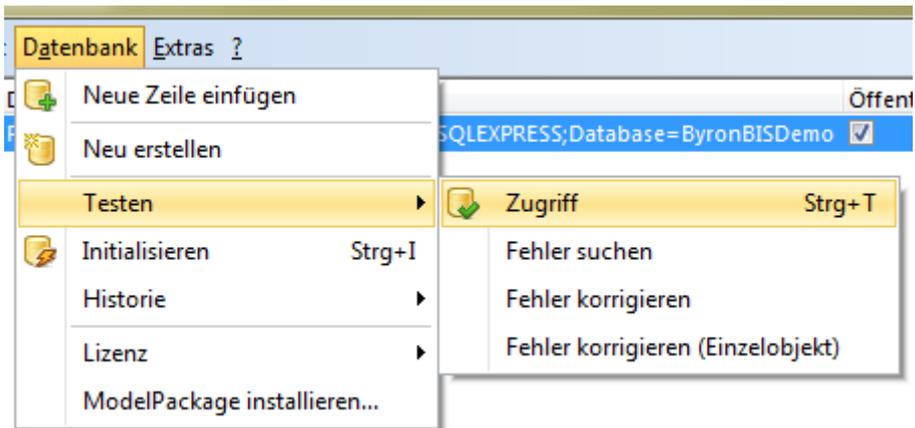


In die Spalte DBMS-Benutzer muss die **Anmeldung** / der Login am SQL Server eingetragen werden; in unserem Fall «demoUser». Danach wird das zugehörige Passwort abgefragt.



Wird für die Datenbank anmeldung [Windows-Authentication](#) verwendet, dann muss kein Benutzer / Passwort hinterlegt werden.

Anschliessend kann der Zugriff getestet werden.



Ist der Zugriff in Ordnung, kann Byron/BIS gestartet werden.

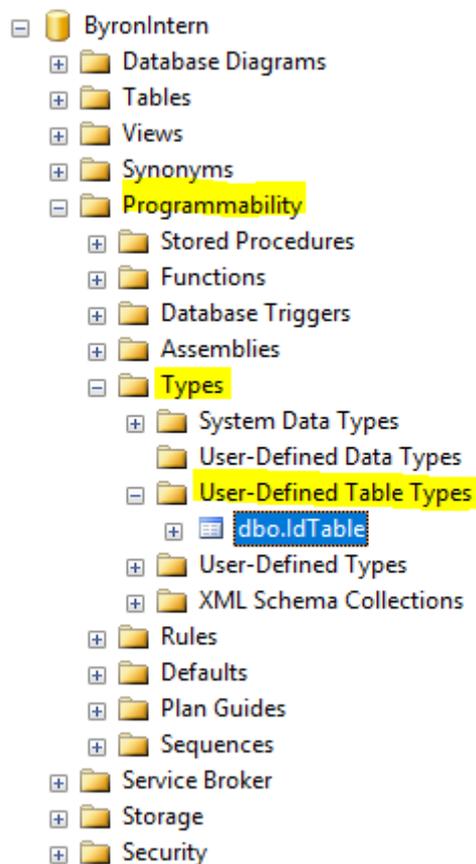
8 FAQ

8.1 Laufzeitfehler: «Der dbo.IdTable-Datentyp ... wurde nicht gefunden»

Bei der Ausführung von Byron/BIS auf MS SQL Server kommt es zu einem Fehler, welcher in Logfile in etwa so festgehalten wird:

```
AnalyzeEx - ErrorCode: 2
ObjectType_GetInstances > Der dbo.IdTable-Datentyp für die Spalte, den
Parameter oder die Variable Nr. '1' wurde nicht gefunden
SET NOCOUNT ON; DECLARE @t [dbo].[IdTable] ; INSERT INTO @t (Id) VALUES
('{1EA8CB12-E881-4FAF-95AD-0098BF538A3F}'), ('{F28A00DE-1962-4CEB-BFC0-00C087
DD98C4}'), ('{32ABA59E-7BDF-4304-A973-00F9B0A54576}'), ('{0ED078D2-1C1F-4D92-A
```

Der Fehler entsteht, weil der Tabellentyp **dbo.IdTable** in der Datenbank fehlt.



Dies kann z.B. nach einem Restore einer MS SQL Server Datenbank auftreten.

Lösung: der Tabellentyp wird von Hand erzeugt:

```
USE <Database-Name>;
CREATE TYPE [dbo].[IdTable] AS TABLE (
    [Id] [UNIQUEIDENTIFIER] NOT NULL PRIMARY KEY
);
```

Nicht vergessen – alle Benutzer benötigen Ausführungsberechtigungen auf diesem Tabellentyp (vgl. [Datenbankbenutzer](#)).

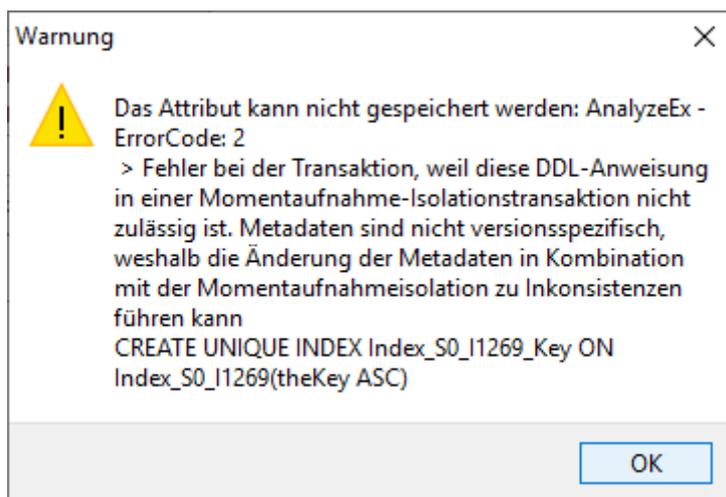
```
USE <Database-Name>;
GRANT EXECUTE ON TYPE :: [dbo].[IdTable] TO ByronBISUser
```

8.2 SQL-Fehler beim Ändern einer Indexierung eines Attributs.

Wird "indexiert" oder "eindeutig" eines Attributs geändert tritt folgender Fehler auf:

```
TSQLExecuter.AnalyzeError: 80040E14 - Fehler bei der Transaktion, weil diese
DDL-Anweisung in einer Momentaufnahme-Isolationstransaktion nicht zulässig
ist. Metadaten sind nicht versionsspezifisch, weshalb die Änderung der Meta-
daten in Kombination mit der Momentaufnahmeisolation zu Inkonsistenzen füh-
ren kann
```

Bzw. Folgende MessageBox:



Der Grund für diesen Fehler und die nötigen Massnahmen sind in [Änderung der SQL-Datenbankstruktur durch Byron/BIS](#) beschrieben.

8.3 Wird *Microsoft SQL AlwaysOn Cluster* unterstützt?

Ja, Byron/BIS mit SQL-Datenbanken unterstützt *Microsoft SQL AlwaysOn Cluster*.

Der ConnectionString muss dafür neben den Angaben des DB-Teams auch folgenden Wert enthalten:

```
MultiSubnetFailover=Yes;
```

Beispiel ConnectionString:

```
Driver={SQL Server Native Client 11.0};Persist Security Info=False;
Database=byronbis;Server=xxx;MultiSubnetFailover=Yes;
SQLQueryTimeout=120;NotificationWaitInterval=1000;IsolationLevel=0,3;
```

8.4 Fehler bei BISRestore

Situation:

- BISRestore funktioniert nicht, obwohl der verwendete Datenbankbenutzer dbowner-Rechte hat.
- Im Log von BISRestore hat es eine Zeile
`TSQLExecuter.AnalyzeError: 80040E14 - The default schema does not exist`

Massnahme: Dem verwendeten Benutzer muss ein Default-Schema zugeordnet werden. Im Normalfall heisst dieses «dbo».

8.5 Welche Änderungen in Byron/BIS lösen einen «Modelländerungsfehler» aus?

Siehe auch «DirtyModelAllowed» in [Konfiguration der Datenbankschnittstelle](#).

Folgende Änderungen in Byron/BIS lösen bei den anderen Clients einen «Modelländerungsfehler» aus:

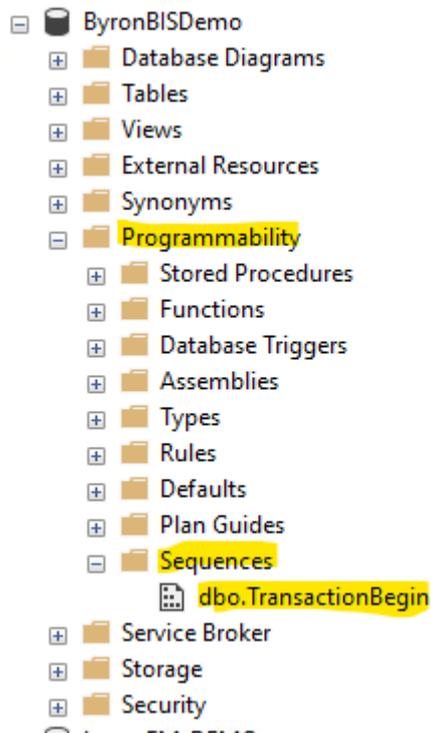
- Erzeugen von Klassen, Attributen oder Assoziationen
- Löschen von Klassen, Attributen oder Assoziationen
- Ändern des Namens einer Klasse, eines Attributs oder einer Assoziation
- Ändern der Aufzählung eines Aufzählungsattributs
- Ändern des Formats eines Attributs
- Ändern der Flags «Lexikalisch sortiert», «Klassenattribut» oder «eindeutig» eines Attributes
- Ändern der Kardinalität einer Assoziation
- Ändern des Flags «virtual» einer Assoziation
- Ändern der Zugriffsrechte eines Attributs oder einer Assoziation
- Ändern der Flags «Kopiert», «Indiziert», «Pseudo», «Traced» oder «im Datencluster gespeichert» eines Attributs oder einer Assoziation
- Erlauben oder Entfernen von Attributen und Assoziationen an einer Klasse
- Ändern des Default-Containers einer Klasse
- Ändern der Flags «Cluster erzeugen», «DatenCluster erzeugen», «Segment erzeugen», «Versteckt», «Erzeugen geschützt», «Löschen geschützt», «Traced» oder «Datencluster für Unterobjekte verwenden» einer Klasse
- Setzen einer «OnDestroy» - Methode an einer Klasse
- Ändern eines binären Attributs einer Klasse – da alle Methoden einer Klasse in einem binären Attribut gespeichert sind, führt dies dazu, dass alle Änderungen an Methoden einer Klasse als Modelländerung erkannt werden.

8.6 Laufzeitfehler: «Object 'TransactionBegin' is not a sequence object»

Der vollständige Fehler sieht so aus:

```
err 02.06.2022 09:17:03.537 TSQLExecuter.AnalyzeError:
80040E14 - Object 'TransactionBegin' is not a sequence object
```

Lösung: stellen Sie sicher, dass die SEQUENCE **dbo.TransactionBegin** in der Datenbank existiert,



und erzeugen Sie diese bei Bedarf.

```
CREATE SEQUENCE TransactionBegin AS bigint START WITH 1;
DELETE FROM Modifications;
DELETE FROM TransactionEnd
```

N.B.: auf dem SQL Server mit der Version <= 2008 wurde statt einer SEQUENCE eine Tabelle **dbo.TransactionBegin** verwendet, welche allenfalls vorgängig gelöscht werden muss:

```
DROP TABLE dbo.TransactionBegin
```

Stammt die Datenbank aus einer alten Installation, dann muss vermutlich auch der Tabellentyp **dbo.IdTable** erzeugt werden. Siehe dazu [Laufzeitfehler: «Der dbo.IdTable-Datentyp ... wurde nicht gefunden»](#)

8.7 Fehler BISRestore – Zugriffsverletzung

Log-Fehler:

```
`err 06.12.2022 15:37:25.344 Zugriffsverletzung bei Adresse 008ADFF4 in Modul 'BISRestore.exe'. Lesen von Adresse 00000054`
```

Mögliche Ursache: Fehlendes Schreibrecht auf die BIX-Datei, welche wiederhergestellt werden soll.

8.8 Abweichendes Datenbank-Schema (dbo)

Wird mit einem anderen Datenbankschema als dem üblichen dbo gearbeitet, so muss folgendes berücksichtigt werden:

- Default-Schema bei den Benutzern muss ebenfalls auf das entsprechende Schema angepasst werden.
- Wird mit Benutzern gearbeitet mit unterschiedlichen Zugangsrechten (DB-Admin, DB-User), so brauchen beide dasselbe DB-Schema.

8.9 Fehler «Eine vorhandene Verbindung wurde vom Remotehost geschlossen»

Fehlermeldung:

```
TCP Provider: Eine vorhandene Verbindung wurde vom Remotehost geschlossen
```

Dieser Fehler tritt auf, wenn ein SQL Server 16 (Server 2022) zusammen mit einer älteren Version der SQL Server Native Client Library verwendet wird (i.e. 11.2 oder älter).

Lösung: Verwenden Sie die neueste Version des SQL Server Native Client

8.10 Welche Datenbank-Indices sollen erzeugt werden?

Für einfache Anwendungen werden alle nötigen Indices bereits durch Byron/BIS erzeugt.

Indices zum Optimieren der Verwendung indizierter Assoziationen

Werden in der Anwendung häufig indizierte Assoziationen verwendet und diese auch in Abfragen genutzt, dann können folgende zusätzliche Indices eine Performancesteigerung erzielen.

```
CREATE NONCLUSTERED INDEX [Index_S0_Assoc_RoleId_To] ON [Index_S0_Assoc] (
    [roleId] ASC,
    [toObj] ASC
)
INCLUDE([fromObj])
```

und

```
CREATE NONCLUSTERED INDEX [Index_S0_Assoc_RoleId_From] ON [dbo].[Index_S0_Assoc] (
    [roleId] ASC,
    [fromObj] ASC
)
INCLUDE([toObj])
```

Beispiel für eine Abfrage, welche indizierte Attribute und Assoziationen nutzt:

```
START VALUE "rep_Solltermin" >= 41913 <= 41943.9993
ANDVALUE "rep_LeIntern" <> 20
ANDREFERENCES "rep_ActivityToPerson" (
```

```

        START "{FCE67B00-4127-6915-929E-0080C89589E7}"
    )

```

Indices zum Optimieren der Verwendung indizierter Attribute

Folgende Optimierungen der Indices der SQL Datenbank werden empfohlen.

```

DROP INDEX [Index_S0_Assoc_To] ON [dbo].[Index_S0_Assoc];
GO
CREATE NONCLUSTERED INDEX [Index_S0_Assoc_To]
ON [dbo].[Index_S0_Assoc] (toObj ASC, roleId ASC)
INCLUDE(fromObj);
GO

```

```

DROP INDEX [Index_S0_Assoc_From] ON [dbo].[Index_S0_Assoc];
GO
CREATE NONCLUSTERED INDEX [Index_S0_Assoc_From]
ON [dbo].[Index_S0_Assoc] (fromObj ASC, roleId ASC)
INCLUDE(toObj);
GO

```

Weiter können einzelne **Indextabellen** (mit dem Namen Index_S0_Innnn), welche häufig verwendet werden optimiert werden wie folgt. Die **Indexnummer** muss entsprechend ersetzt werden:

```

DROP INDEX [Index_S0_I875_Key] ON [dbo].[Index_S0_I875];
GO

CREATE NONCLUSTERED INDEX [Index_S0_I875_Key]
ON [dbo].[Index_S0_I875] ([theKey] ASC)
INCLUDE (theValue);
GO

```

Bei welchen Indextabellen diese Änderung sinnvoll ist, kann mit Hilfe des [Query Stores](#) des SQL Servers ermittelt werden – Abfragen über Indices, welche häufig aufgerufen werden und / oder teuer sind.

Neu erzeugte Indextabellen verwenden ab Byron/BIS Version 5.9.2 die vorgeschlagenen Indexdefinitionen.

Wenn die Auswertungen der Abfragen [RID Lookups](#) auf den Indextabellen zeigen, dann kann auch noch der Index *Tabellenname_Value* geändert werden.

```

DROP INDEX [Index_S0_I875_Value] ON [dbo].[Index_S0_I875];
GO

CREATE NONCLUSTERED INDEX [Index_S0_I875_Value]
ON [dbo].[Index_S0_I875] ([theValue] ASC)
INCLUDE (theKey);
GO

```

Indices zum Optimieren der Objekthistorie

Um Abfragen aus dem Trace zu optimieren können zwei Indices auf der Tabelle [dbo].[Trace] angelegt werden, sofern sie noch nicht existieren.

```

CREATE CLUSTERED INDEX Trace_Pk ON [dbo].[Trace] (wann ASC);
CREATE INDEX Trace_Id ON [dbo].[Trace] (objekt ASC);
CREATE INDEX Trace_What ON [dbo].[Trace] (was ASC);

```