Wartung GEOS Pro - Datenbanken

Empfehlungen für SQL-Server

Version vom 27.03.2020

Hexagon Technology Center GmbH
Octave | Niederlassung Zürich
Flurstrasse 55
8048 Zürich
Schweiz
+41 (0)43 322 46 46
info.switzerland.sig@octave.com
www.octave.com



Inhaltverzeichnis

1 Einleitung 3

2 Abgrenzung 3

3 Empfehlung 3

3.1 Tabelle ModificationLog 3

3.1.1 Hintergrundinformationen 3

3.1.2 Empfehlung 4

3.2 Databank verkleinern 4

3.2.1 Hintergrundinformationen 4

3.2.2 Empfehlung 4

3.3 Indexe pflegen 6

3.3.1 Hintergrundinformationen 6

3.3.1.1 Indes neu aufbauen 6

3.3.1.2 Indes reoganisieren 6

3.3.2 Empfehlung 7

3.4 Indexe pflegen 8

3.4.1 Hintergrundinformationen 8

3.4.1.1 Index Statistiken 8

3.4.1.2 Spalten Statistiken 8

3.4.2 Empfehlungen 8

3.5 Weitere Massnahmen 9

4 Reihenfolge der empfohlenen Massnahmen 10

5 Anhang 1: Wartungsskript weekly 10

6 Anhang 1: Wartungsskript half-yearly 11

1 Einleitung

GEOS Pro ist eine Applikation, bei welcher relationale Datenbanken zugrunde liegen. Für eine gute Performance hat somit nicht nur die Applikation zu sorgen, sondern auch die zugrundeliegende Datenbank. Nur wenn alle Tabellen, Indexe und physische Speicherbelegung in einem guten «Gesundheitszustand» vorliegen, können die entsprechenden Datenbank-Funktionen performant arbeiten.

Dieses Dokument beschreibt Empfehlungen von Seiten Octave für die regelmässige Wartung von GEOS Pro – SQL-Datenbanken.

2 Abgrenzung

Dieses Dokument enthält Empfehlungen nur für SQL-Server Datenbanken. Andere Datenbanken haben zum Teil ähnliche Wartungsarbeiten nötig, das Vorgehen unterscheidet sich jedoch häufig.

Die Nutzung der nachfolgend empfohlenen Wartungsmethoden und -skripte liegt in der Verantwortung des Anwenders. HxGN Schweiz übernimmt keine Haftung.

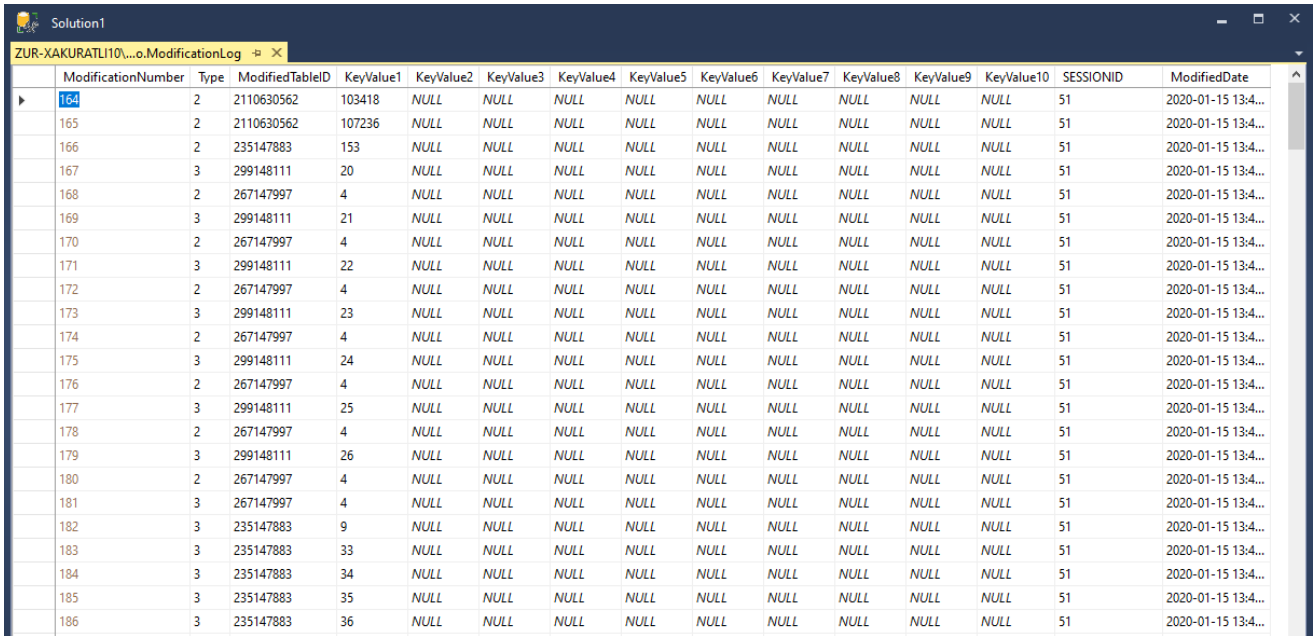
Die empfohlenen Wartungsmethoden und -skripte sind häufig an die kundenseitige Systeminfrastruktur und Datenverwaltungskonzepte anzupassen.

3 Empfehlung

3.1 Tabelle ModificationLog

3.1.1 Hintergrundinformationen

Die ModificationLog-Tabelle erfasst alle Modification (Insert, Update, Delete), welche GeoMedia in den einzelnen Datenbank-Tabellen vornimmt. Der Inhalt dieser Tabelle kann somit, abhängig von der Editier-Tätigkeit eines GEOS Pro – Operates, schnell anwachsen. Wenn diese Tabelle viele Datenbank-Records enthält, hat dies jedoch einen negativen Einfluss auf die Performance einzelner GeoMedia-Funktionen.

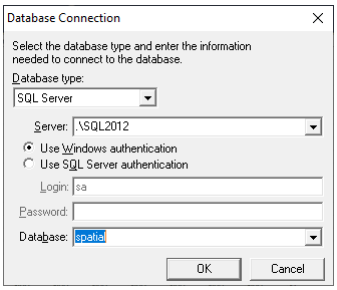

Solution1

ZUR-XAKURATLI10\...o.ModificationLog

ModificationNumber	Type	ModifiedTableID	KeyValue1	KeyValue2	KeyValue3	KeyValue4	KeyValue5	KeyValue6	KeyValue7	KeyValue8	KeyValue9	KeyValue10	SESSIONID	ModifiedDate
164	2	2110630562	103418	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
165	2	2110630562	107236	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
166	2	235147883	153	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
167	3	299148111	20	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
168	2	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
169	3	299148111	21	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
170	2	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
171	3	299148111	22	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
172	2	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
173	3	299148111	23	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
174	2	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
175	3	299148111	24	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
176	2	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
177	3	299148111	25	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
178	2	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
179	3	299148111	26	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
180	2	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
181	3	267147997	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
182	3	235147883	9	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
183	3	235147883	33	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
184	3	235147883	34	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
185	3	235147883	35	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...
186	3	235147883	36	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	51	2020-01-15 13:4...

Abbildung 1: ModificationLog-Tabelle von GeoMedia

3.1.2 Empfehlung

Massnahme	Regelmässiges Löschen des Tabelleninhaltes
Vorgehen (manuell)	1. Datenbank-Dienstprogramm von GeoMedia öffnen  2. Funktion «Clear Modification Log…» ausführen

Vorgehen (Skript)	`USE [<database>]`
	
	`TRUNCATE TABLE dbo.ModificationLog`

Häufigkeit	wöchentlich

3.2 Databank verkleinern

3.2.1 Hintergrundinformationen

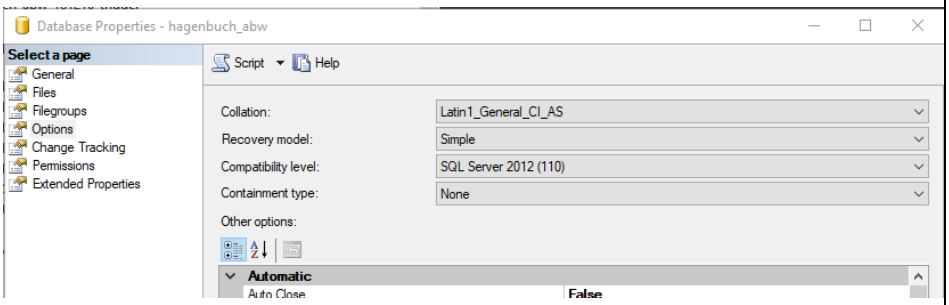
Der Microsoft SQL-Server speichert alle Änderungen der Datenbank in einer "Log-Datei", damit die Änderungen jederzeit wieder rückgängig gemacht werden können. Je nach Einstellung wird die Log-Datei dabei immer größer. Mit der Zeit kann es sein, dass die Datenbank-Datei z.B. **2 GB** groß ist, die Log-Datei aber **20 GB**! Das belastet den Server unnötig und verlangsamt auch die Zugriffe auf die Daten teilweise.

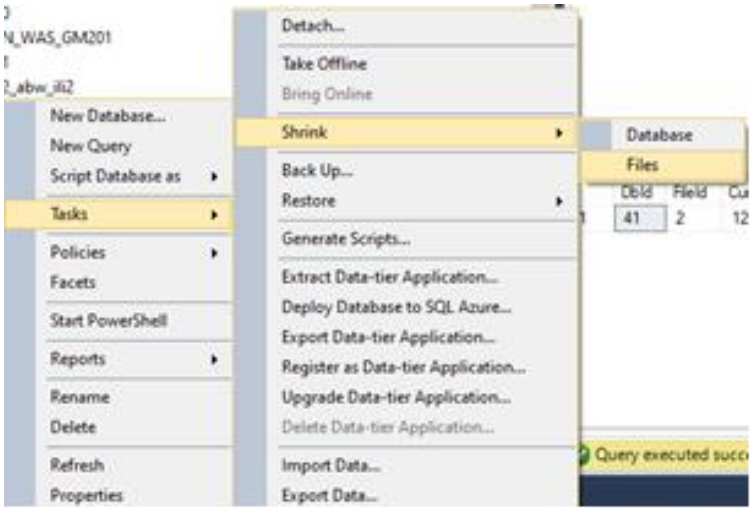
Folgende Einstellungen beeinflussen die Grösse einer Datenbank

- Datenvolumen / Anzahl Inserts
- Initial-Grösse der Datenbank-Dateien
- Gewähltes Recovery-Modell (Full, Simple)

3.2.2 Empfehlung

Massnahme	Recovery-Modell auf “Simple” ändern
Vorgehen (manuell)	1. SQL Server Management Studio öffnen und die Eigenschaften der Datenbank öffnen.

	2. Unter der Page «Optionen» das Recovery model auf Simple ändern 
Vorgehen (Skript)	`ALTER DATABASE <database> SET RECOVERY SIMPLE`
Häufigkeit	einmalig

Massnahme	Datenbank-Grösse verkleinern
Vorgehen (manuell)	1. Im SQLServer Management Studio auf der entsprechenden Datenbank das Kontextmenu aufrufen und den Befehl für das Shrinken der Files aufrufen  2. Nacheinander den File type «Data» und «Log» shrinken

Vorgehen (Skript)	`USE [<database>]`
	`GO`
	`DBCC SHRINKFILE (N'<logicalname>', 0, TRUNCATEONLY)`
	`GO`
	`USE [<database>]`
	`GO`
	`DBCC SHRINKFILE (N'<logicalname>', 0, TRUNCATEONLY)`
	`GO`

	⇨ Der <logicalname> ist unter Properties > Files ersichtlich (1. Spalte). Bei diesem Skript sind beide Logical Name anzugeben.
Häufigkeit	wöchentlich

3.3 Indexe pflegen

3.3.1 Hintergrundinformationen

Datenbank-Indizes sind eine von der Datenstruktur getrennte Indexstruktur, welche die Suche und das Sortieren nach bestimmten Feldern beschleunigen. Lesende Datenbankzugriffe profitieren von einem Index, schreibende Zugriffe werden durch Indizes verlangsamt.

Bestehende Indexe werden durch die Nachführung an den Daten laufend nachgeführt. Neuere Indexwerte können jedoch nicht immer an der korrekten Speicherstelle verwaltet werden. Dadurch nimmt die «Zerstückelung» der Indexe (auch Fragmentierung genannt) laufend zu. Je nachdem wie stark die Indexe einer Tabelle fragmentiert sind, kann sich dadurch die Abfrage-Performance auf eine oder mehrere Tabellen sehr verschlechtern.

Den Fragmentierungsgrad der Indexe einer Datenbank kann über folgendes SQL-Statement angezeigt werden:

```
SELECT OBJECT_NAME(ind.OBJECT_ID) AS TableName,

ind.name AS IndexName, indexstats.index_type_desc AS IndexType,

indexstats.avg_fragmentation_in_percent

FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, NULL) indexstats

INNER JOIN sys.indexes ind

ON ind.object_id = indexstats.object_id

AND ind.index_id = indexstats.index_id

ORDER BY indexstats.avg_fragmentation_in_percent DESC
```

Die Fragmentierung von Indexen kann nun auf zwei Arten verbessert resp. eliminiert werden.

3.3.1.1 Indes neu aufbauen

Hierbei wird der Index gelöscht und komplett neu aufgebaut. So wird die Ordnung, die Fragmentierung wieder bereinigt, da alle Einträge eindeutig passend und hintereinander neu aufgebaut werden. Dadurch wird die Fragmentierung entfernt und freier Speicherplatz geschaffen, welcher im Anschluss wieder freigegeben werden kann. Diese Operation ist relativ «kostenintensiv», da erst der alte Index gelöscht werden muss, dann der Index anhand der Daten aus der Tabelle komplett neu erstellt werden muss.

3.3.1.2 Indes reoganisieren

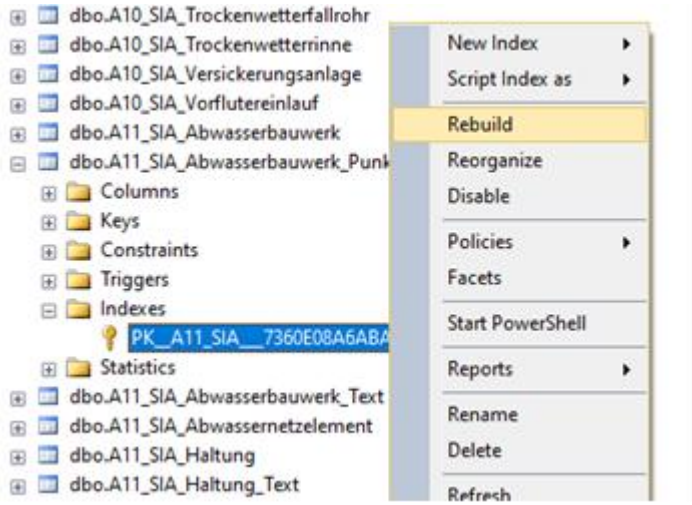
Das Neuorganisieren eines Indexes beansprucht minimale Systemressourcen, hierbei werden die Einträge überprüft, ob sie noch an der richtigen Position liegen und werden dann, wenn nötig umsortiert, um eine optimale und logisch richtige Reihenfolge wiederherzustellen.

Microsoft empfiehlt, die Indexe nur bei einer gewissen Fragmentierung neu aufzubauen resp. zu reorganisieren.

Fragmentierungsgrad	Massnahme
>5% und < =30%	Neu organisieren
>30%	Neu aufbauen

Tabelle 1: Empfehlung Indexwartung (Quelle: https://support.microsoft.com)

3.3.2 Empfehlung

Massnahme	Indexe nach Empfehlung von Microsoft neu organisieren oder aufbauen
Vorgehen (manuell)	1. Den Fragmentierungsgrad der Indexe einer Datenbank mit obenstehendem Skript anzeigen 2. Jeder Index mit einer Fragmentierung > 30% neu aufbauen. Hierzu ist auf dem entsprechenden Index das Kontextmenu zu öffnen und die Funktion «Rebuild» zu öffnen.  3. Mit OK wird der Neuaufbau gestartet 4. Jeder Index mit einer Fragmentierung > 5% und < 30% ist neu zu organisieren. Hierzu ist auf dem entsprechenden Index das Kontextmenu zu öffnen und die Funktion «Reorganize» zu öffnen. 5. Mit OK wird die Reorganisation gestartet

Vorgehen (Skript)	`USE [<database>]`
	`GO`
	`ALTER INDEX [<indexname>] ON [<tablename>] REBUILD`
	`GO`
	`ALTER INDEX [<indexname>] ON [<tablename>] REORGANIZE`
	`GO`

Häufigkeit	halbjährlich

⇨ Durch das Neuerstellen oder Reorganisieren von kleinen Indizes wird die Fragmentierung oft nicht oder nur geringfügig verringert. Dies ist durch die Organisation der physischen Speicherung mittels «Blöcken» zu erklären. Hohe Fragmentierungen bei kleinen Indexen haben keinen Einfluss auf die Performance.

⇨ Die Wartung der Indexe ist auch mittels einer SQL-Prozedur ausführbar, welche alle Indexe einer Datenbank, abhängig vom Fragmentierungsgrad, neu organisiert oder aufbaut. Eine solche Prozedur ist im Beispielskript im Anhang 2 ersichtlich.

3.4 Indexe pflegen

3.4.1 Hintergrundinformationen

Statistiken sind SQL-Server-Objekte, die Metriken über die Datenanzahl und -verteilung innerhalb einer Spalte enthalten. Diese werden herangezogen, um Abfragen optimal zu verarbeiten (Bestimmung des besten Ausführungsplanes). Es werden zwei Arten unterschieden:

3.4.1.1 Index Statistiken

Diese werden automatsch durch SQL-Server bei der Erstellung eines Indexes erstellt, und besitzen auch denselben Namen.

3.4.1.2 Spalten Statistiken

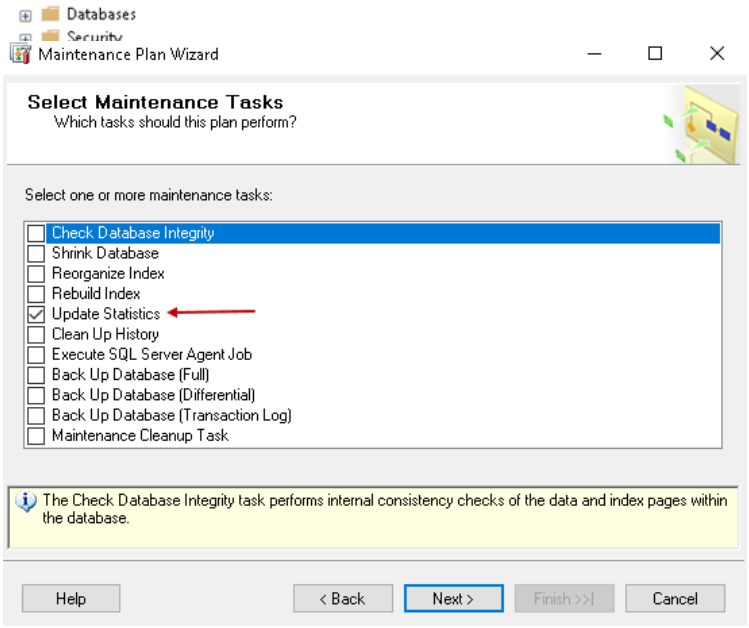
Diese werden automatisch für jede Spalte einer Tabelle erstellt, falls die Option «Auto Create Statistics» auf «True» gesetzt ist.

Wenn sich die Daten in einer Datenbank ändern, werden die Statistiken nicht dauernd aktualisiert und können somit nicht mehr aktuelle Informationen enthalten. Dies kann dazu führen, dass Abfragen nicht mehr mit den optimalen Ausführungsplänen bearbeitet werden. Veraltete Statistiken beeinträchtigen somit die Performance.

SQL-Server bietet hier zwar die beiden Optionen «Auto Create Statistics» und «Auto Update Statistics». Letztere Option veranlasst den SQL-Server, die Statistiken automatisch upzudaten, jedoch erst bei einer grösseren Diskrepanz zwischen Statistik und aktuellem Tabelleninhalt.

3.4.2 Empfehlungen

Massnahme	Statistiken mittels Wartungsskript aktualisieren
Vorgehen (manuell)	☞ Das manuelle Vorgehen ist in der Express-Edition von SQL-Server nicht möglich

	1. Im SQL-Server Management Studio den Maintenance Plan Wizard öffnen  2. Als Maintenance Task «Update Statistics" auswählen und den Wizard weiter befolgen.

Vorgehen (Skript)	`USE [<database>]`
	`GO`
	`EXEC sp_updatestats;`

Häufigkeit	halbjährlich

3.5 Weitere Massnahmen

- Nicht mehr benötigte Benutzer regelmässig löschen
- Nicht mehr benötigte Objektklassen in der Datenbank regelmässig löschen
- GRIVIS-GEOS Spalten-Sortierungen jeweils wieder aufheben

4 Reihenfolge der empfohlenen Massnahmen

Die empfohlenen Massnahmen entfalten Ihre optimale Wirkung nur, wenn diese auch in der richtigen Reihenfolge ausgeführt werden.

Wöchentlich

1. Erstellung Backup

   ⇨ Die Datensicherung ist zwingend am Anfang resp. vor den Wartungsskripten auszuführen

2. ModificatonLog und G5ERR leeren

3. Datenbank verkleinern

Halbjährlich

1. Indexe neu organisieren oder aufbauen

2. Statistiken neu rechnen

5 Anhang 1: Wartungsskript weekly

```
/*****************************************/
/*** WARTUNGSSKRIPT GEOS Pro Databanken ***/
/*** SQL Server                         ***/
/*** a/m/t software service ag          ***/
/*** Wartungsskript weekly              ***/
/*****************************************/
/*** die Verwendung dieses Skripts liegt***/
/*** in der Verantwortung des Anwenders ***/
/*****************************************/

/***Anleitung*************************************************************/
-- <database> mit korrektem DB-Namen ersetzten
-- <datapath> mit mit Pfad für Ablage Backup ersetzen
-- <backupname> mit Namen der gewünschten Backup-Bezeichnung ersetzen
-- <logicalname> mit dem logischen Namen der DB-Dateien ersetzen


-- Backup erstellen
BACKUP DATABASE [<database>] TO  DISK = N'<datapath>\<backupname>.bak'
WITH NOFORMAT, INIT, NAME = N'<database>-Full Database Backup', SKIP, NOREWIND, NOUNLOAD,
STATS = 10
GO

-- ModificationLog leeren
TRUNCATE TABLE [<database>].[dbo].[ModificationLog];
GO

-- A00_G5ERR leeren
TRUNCATE TABLE [<database>].[dbo].[A00_G5ERR];
GO

-- LOG-Datei verkleinern
USE [<database>];
GO
DBCC SHRINKFILE (N'<logicalname>_log' , 0, TRUNCATEONLY);
GO

-- Daten-Datei verkleinern
USE [<database>];
GO
DBCC SHRINKFILE (N'<logicalname>' , 0, TRUNCATEONLY);
GO
```

```
-- Datenbank verkleinern
DBCC SHRINKDATABASE(N'<database>' );
GO
```

6 Anhang 1: Wartungsskript half-yearly

```
/*****************************************/
/*** WARTUNGSSKRIPT GEOS Pro Databanken ***/
/*** SQL Server                         ***/
/*** a/m/t software service ag          ***/
/*** Wartungsskript half-yearly         ***/
/*****************************************/
/*** die Verwendung dieses Skripts liegt***/
/*** in der Verantwortung des Anwenders ***/
/*****************************************/

/***Anleitung*************************************************************/
-- <database> mit korrektem DB-Namen ersetzten



-- Indexe neu organisieren/aufbauen

/*** Declaration ***/
DECLARE @Database VARCHAR(255)
DECLARE @Table VARCHAR(255)
DECLARE @cmd NVARCHAR(500)
DECLARE @fillfactor INT

DECLARE @FragValue INT
DECLARE @ID_DB AS SMALLINT;
DECLARE @ID_TABLE AS INT;

SET @fillfactor = 90

/*** Defition cursor ***/
DECLARE DatabaseCursor CURSOR FOR
SELECT name FROM master.dbo.sysdatabases
WHERE name IN ('<database>')
ORDER BY 1

OPEN DatabaseCursor

FETCH NEXT FROM DatabaseCursor INTO @Database
WHILE @@FETCH_STATUS = 0
BEGIN

   SET @cmd = 'DECLARE TableCursor CURSOR FOR SELECT ''['' + table_catalog + ''].['' +
table_schema + ''].['' +
  table_name + '']'' as tableName FROM [' + @Database + '].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 @ID_DB = DB_ID(@Database);
               SET @ID_TABLE = OBJECT_ID(@Table);
               SET @FragValue = 0;

               /*** Calc. frag.-level ***/
               SET @FragValue = CONVERT(INT, (SELECT TOP(1) A.avg_fragmentation_in_percent
FROM sys.dm_db_index_physical_stats
                     (@ID_DB, @ID_TABLE, NULL, NULL , 'DETAILED') AS A WHERE ([record_count]
<> 0) ORDER BY [avg_fragmentation_in_percent] DESC))

               /*** frag.-level<30%=REORGANIZE // frag.-level>30%=REBUILD ***/
               IF (@FragValue > 30)
                      BEGIN
                             PRINT 'REBUILD='+CONVERT(NVARCHAR(10),@FragValue)+' Table:'
+@Table
```

```
                                SET @cmd = 'ALTER INDEX ALL ON ' + @Table + ' REBUILD WITH
(FILLFACTOR = ' + CONVERT(VARCHAR(3),@fillfactor) + ')'
                        END

                ELSE IF (@FragValue > 5)
                        BEGIN
                                PRINT 'REORGANIZE='+CONVERT(NVARCHAR(10),@FragValue)+' Table:'
+@Table
                                SET @cmd = 'ALTER INDEX ALL ON ' + @Table + ' REORGANIZE'
                        END

                /*** Execute command ***/
                --EXEC (@cmd)

        FETCH NEXT FROM TableCursor INTO @Table
   END

   CLOSE TableCursor
   DEALLOCATE TableCursor

   FETCH NEXT FROM DatabaseCursor INTO @Database

END
CLOSE DatabaseCursor
DEALLOCATE DatabaseCursor


-- Statistiken neu rechnen

USE [<database>];
GO
EXEC sp_updatestats;
```

Über Octave

Octave bietet geschäftskritische Software, die Unternehmen befähigt, in jeder Phase des Asset-Lebenszyklus – bei Design, Entwicklung, Betrieb und Schutz – fundierte Entscheidungen zu treffen. Dies gilt besonders, wo Leistung, Sicherheit und Zuverlässigkeit unverzichtbar sind und Misserfolg keine Option ist. Durch die Umwandlung komplexer operativer Daten in nutzbare Mehrwerte verbindet Octave tiefgreifende Expertise, reale Bedingungen und unternehmensweite Einblicke, um Leistung, Resilienz und Reaktionsfähigkeit zu verbessern.

Im Jahr 2025 leitete Hexagon AB eine Ausgliederung seiner Divisionen Asset Lifecycle Intelligence sowie Safety, Infrastructure & Geospatial ein. Zusammen mit ETQ und Bricsys entstand aus dieser Ausgliederung Octave, das künftig als eigenständiges Unternehmen agieren wird. Octave stützt sich dabei weiterhin auf fundiertes Fachwissen, langjährige Erfahrung und Beziehungen, die über Jahrzehnte hinweg aufgebaut wurden.

Octave beschäftigt mehr als 7.000 Mitarbeiter in 45 Ländern. Erfahren Sie mehr unter **octave.com** and folgen Sie uns auf **LinkedIn**.

©2026 Intergraph Corporation und/oder ihre verbundenen Unternehmen. Alle Rechte vorbehalten. Das Octave-Logo ist eine Marke der Intergraph Corporation. Andere Marken und Produktnamen sind registrierte Warenzeichen der jeweiligen Eigentümer.

Intergraph Corporation
305 Intergraph Way
Madison, AL 35758
USA
T: +1 256 730 2000
www.octave.com

Hexagon Technology Center GmbH
Flurstrasse 55
8048 Zürich
Schweiz
T: +41 43 322 46 46
www.octave.com