

# SQL Server backuppen met Gerrit Mantel procedures

Handleiding SQL Server databases backuppen en restoren met Gerrit Mantel procedures



<https://www.gerritmantel.nl>

Datum laatste wijziging is 14-05-2026

# Inhoud

1	Inleiding.....	3
2	Inrichten database backups .....	4
2.1	Maintenance database.....	4
2.2	Aanmaak van jobs .....	4
2.3	Eigenschappen van de procedures .....	6
3	Herstellen databases.....	10
3.1	SQL Server crash recovery.....	10
3.1.1	Herinstallatie van SQL Server .....	10
3.1.2	Restore de master database .....	10
3.1.3	Restore de msdb database.....	10
3.1.4	Restore de model database .....	11
3.2	Genereren van een survive script .....	11
3.3	Restore user databases .....	11

# 1 Inleiding

Deze handleiding beschrijft een methode voor het maken van SQL Server database backups en het herstellen van databases m.b.v. Gerrit Mantel procedures.

Uitgangspunt is het hebben draaien van een Microsoft SQL server database omgevingen van Versie 2012 of hoger, en een Editie met een Job Agent (Standard-, Enterprise- of Developer Edition), draaiend op een Windows omgeving.

De in deze handleiding genoemde MSSQL procedures wordt als freeware aangeboden op de website van gerritmantel.nl (MSSQL Programmamodules).

Mijn backup strategie komt neer op het uitvoeren van 4 verschillende types backups. Deze worden in 4 aparte jobs met een bepaalde frequentie gestart:

1. Het maken van full backups van alle user databases op een bepaalde dag van de week.
2. Het maken van differential backups van alle user databases op alle andere dagen van de week.
3. Het 'frequent' maken van transaction log backups van alle user databases. Te denken valt aan elk uur (\*).
4. Het 'frequent' maken van full backups van de system databases. Te denken valt aan elke 4 uur (\*).

(\*) Om risico's te verkleinen kunnen log backups worden geïntensiveerd tot b.v. elk kwartier of half uur en system database backups tot elk uur of elke 2 of 3 uur

De backup disk moet een andere disk zijn dan waarop de MSSQL databestanden zich bevinden. Eventueel kan dit een netwerklocatie zijn, maar de bandbreedte van het netwerk kan een issue worden bij zeer grote databases. Een spreiding van backup tijdstippen en/of disks bij gebruik van meerdere MSSQL database servers kan nodig zijn.

Database backup bestanden behoren op hun beurt weer 'frequent' te worden 'gearchiveerd' naar b.v. een externe USB disk. Te denken valt aan elke dag. Een simpele oplossing hiervoor is te realiseren met het (Windows freeware) programma SyncFolders. Zie hiervoor "Windows PC backuppen met Syncfolders" op gerritmantel.nl – Windows.

Bij een calamiteit van de database omgeving kan een MSSQL Server crash recovery worden uitgevoerd, gebruikmakend van de laatst gemaakte system backup bestanden. De user databases kunnen worden hersteld met de m.b.v. de gerritmantel restore procedure. Deze procedure maakt gebruik van de laatst gemaakte ketens van backup bestanden op disk per database. Zorg er voor dat je een survive script klaar hebt liggen, zoals wordt beschreven in H3.2 "Genereren van een survive script". Hiermee maak je een database aan met de benodigde gerritmantel procedures, als de SQL Server omgeving (nog) leeg is..

## 2 Inrichten database backups

### 2.1 Maintenance database

De gerritmantel programma's (in de vorm van procedures en functies) moeten idealiter worden ondergebracht in een aparte database. Als voorbeeld neem ik de database **DBMAINT**. Maak database DBMAINT aan. Download de sourcecode van de volgende procedures op de gerritmantel.nl website en run de aanmaak scripts in de DBMAINT database:

- prc\_backup\_userdb\_full
- prc\_backup\_userdb\_diff
- prc\_backup\_userdb\_log
- prc\_backup\_systemdb
- prc\_check\_status\_databases
- prc\_create\_backup\_jobs
- prc\_restore\_database
- prc\_restore\_userdb\_script\_only
- prc\_dump\_survive\_script

Alle benodigde subprocedures en functions zit in deze sourcecode.

### 2.2 Aanmaak van jobs

Voor de volgende 5 procedures moet een job worden gemaakt met elk hun eigen job schedule:

- prc\_backup\_userdb\_full
- prc\_backup\_userdb\_diff
- prc\_backup\_userdb\_log
- prc\_backup\_systemdb
- prc\_check\_status\_databases

De procedure **prc\_create\_backup\_jobs** kan hiervoor de 5 jobs aanmaken. Er zijn 4 verplichte parameters, onderstaand voorbeeld is de meest simpele aanroep van deze procedure:

```
EXEC [dbo].[prc_create_backup_jobs]
@backup_path='C:\Data\MSSQL\backup',
@error_report_path='C:\Data\MSSQL\error_report',
@operator_name='SQL Beheerder',
@email_address='g.mantel@gmail.com';
```

Verplichte parameters:

Parameter	Type	Standaard waarde	Betekenis
@backup_path	VARCHAR(255)	"	Backup path.
@error_report_path	VARCHAR(255)	"	Error_report path.
@operator_name	NVARCHAR(128)	"	Naam van de agent Operator.
@email_address	NVARCHAR(100)	"	Email adres. Hier worden job notificaties en error rapporten naar

			toe verzonden.
--	--	--	----------------

Optionele parameters:

Parameter	Type	Standaard waarde	Betekenis
@subdir	BIT	1	Backup naar subdirectory met database naam.
@verify	BIT	0	Check na de backup of het backupbestand valide is. Dit wordt gedaan m.b.v. RESTORE VERIFYONLY na elke backup.
@split	BIT	0	Split full backup bestanden bij grote databases.
@keep_days	INT	90	Ruim backup bestanden ouder dan n dagen. 0 betekent niet opruimen.
@force_full_backup	BIT	0	Laat de log backup procedure een full backup maken als er geen full backup in de backuphistorie aanwezig is.
@full_backup_weekday	VARCHAR(3)	'SAT'	Job schedule weekday voor de full backup job ('SAT', 'SUN', 'MON', 'TUE', 'WED', 'THU' of 'FRI').
@full_backup_start_hour	INT	0	Job schedule start-uur voor de full- en diff backup jobs (0-23 uur).
@log_backup_interval_minutes	INT	60	Job schedule interval voor log backups (15, 30 of 60 minuten).
@system_backup_interval_hours	INT	4	Job schedule interval voor system database backups (1, 2, 4, 6, 8, 12 of 24 uur).
@option	VARCHAR(6)	'CREATE'	Opties. 'CREATE'=Creer jobs, 'LIST'=List backup jobs

De procedure prc\_create\_backup\_jobs zal de volgende zaken regelen:

- ➔ De Job Category 'Database Maintenance' wordt aangemaakt.
- ➔ De Operator met de naam @operator\_name en emailadres @email\_address wordt aangemaakt.
- ➔ De volgende 5 jobs met jobschedules worden aangemaakt:

Job Name	Schedule
DBMAINT - Backup Userdb Full	Weekly on @full_backup_weekday at @full_backup_start_hour hh:05:00
DBMAINT - Backup Userdb Diff	Weekly on every other weekday than @full_backup_weekday at @full_backup_start_hour hh:05:00
DBMAINT - Backup Userdb Log	Daily every @log_backup_interval_minutes minutes between 00:00:00 and 23:59:59
DBMAINT - Backup Systemdb	Daily every @system_backup_interval_hours hours

	between 00:55:00 and 23:59:59
DBMAINT - Check database backup	Daily every 1 hour between 00:50:00 and 23:59:59

In Job step Properties onder Advanced wordt de optie 'Log to table' aan gezet. Dit is handig bij foutopsporing. Namen van Jobsteps en Schedules zijn gelijk aan de jobnaam.

De jobs maken gebruik van de Job Notification – Email. Database Mail moet dus werkend zijn. Hoe je het SQL Server mailsysteem inricht, wordt uitgelegd in het document "SQL Server mail afleveren bij de gmail SMTP server", te vinden bij gerritmantel.nl – MSSQL – Index - MSSQL docs.

### Controle aangemaakte jobs

Er kan een overzicht (in de output grid) worden getoond van eigenschappen van de 5 aangemaakte jobs met:

```
EXEC [dbo].[prc_create_backup_jobs] @option = 'LIST';
```

## 2.3 Eigenschappen van de procedures

De [prc\\_backup\\_userdb\\_full](#) procedure voert full backups uit van alleen user databases en alleen databases die ONLINE staan. Overgeslagen worden logshipping secondary databases en HADR databases als ze secondary replica's zijn. Standaard schrijft de backup elke keer naar één (nieuw) bestand. Bij databases van 1 TB, 2 TB, 4 TB of meer groot is het mogelijk wenselijk om backupbestanden te splitsen. Dit is te realiseren m.b.v. @split=1. Het aantal backup bestanden dat ontstaat is afhankelijk van de totale data bestandsgrootte en het aantal databestanden binnen de database. De volgende regels worden gehanteerd:

- 8 of meer datafiles en totaal minimaal 4 TB -> split naar 8 delen
- 4 of meer datafiles en totaal minimaal 2 TB -> split naar 4 delen
- 2 of meer datafiles en totaal minimaal 1 TB -> split naar 2 delen
- 1 datafile -> niet gesplit

De [prc\\_backup\\_userdb\\_diff](#) procedure voert differential backups uit van alleen user databases en alleen databases die ONLINE staan. Overgeslagen worden logshipping secondary databases en HADR databases als ze secondary replica's zijn.

De procedure voert i.p.v. van een differential backup automatisch een full backup uit als er (nog) geen differential backup kan worden gemaakt. Er kan geen differential backup worden gemaakt in de volgende situaties:

- De differential\_base\_lsn is NULL. Dit is het geval bij nieuwe databases.
- De database staat in FULL of BULK\_LOGGED recovery model, en de last\_log\_backup\_lsn is NULL. Dit kan het geval zijn als het recovery model is gewisseld van SIMPLE naar FULL of BULK\_LOGGED.

Na een restore van een database onder een nieuwe naam, kan er technisch gezien wel een differential backup worden gemaakt van de nieuwe database. In de backuphistorie van de nieuwe database is er echter nog geen full backup aanwezig. Een differential backup komt in de keten van de oorspronkelijke database terecht, en niet in de keten

van de nieuwe database. Omdat dit leidt tot een onoverzichtelijk situatie, voert de procedure `prc_backup_userdb_diff` in zo'n situatie standaard i.p.v. een diff backup automatisch een full backup uit. Indien dit gedrag ongewenst is, kan dit worden onderdrukt met parameter `@force_full_backup=0`;

De `prc_backup_userdb_log` procedure voert transaction log backups uit van alleen user databases en alleen databases die ONLINE staan. Overgeslagen worden databases in SIMPLE recovery model, logshipping primary databases en logshipping secondary databases. Van logshipping primary databases wordt verwacht dat de log backup wordt afgehandeld in de (native) logshipping LS\_Backup jobs. Van HADR databases wordt een log backup gemaakt als de database de preferred backup replica is.

De procedure voert i.p.v. een log backup automatisch een full backup uit als er (nog) geen log backup kan worden gemaakt. Er kan geen log backup worden gemaakt als:

- De `last_log_backup_lsn` is NULL. Dit is het geval bij nieuwe databases, of als het recovery model is gewisseld van SIMPLE naar FULL of BULK\_LOGGED.

Na een restore van een database onder een nieuwe naam, kan er technisch gezien wel een log backup worden gemaakt van de nieuwe database. In de backuphistorie van de nieuwe database is er echter nog geen full backup aanwezig. Een log backup komt in de keten van de oorspronkelijke database terecht, en niet in de keten van de nieuwe database. Omdat dit leidt tot een onoverzichtelijk situatie, voert de procedure `prc_backup_userdb_log` in zo'n situatie standaard i.p.v. een log backup automatisch een full backup uit. Indien dit gedrag ongewenst is, kan dit worden onderdrukt met parameter `@force_full_backup=0`;

**LET OP!** Als er door de log backup procedure een full backup wordt gemaakt van een grote database, en de log backup frequentie is hoog, dan kan het gewenste beschermingsnivo van de andere databases mogelijk niet worden gehaald.

De `prc_backup_all_systemdb` procedure voert backups uit van alleen system databases. Overgeslagen wordt de tempdb. Mocht de model database in FULL of BULK\_LOGGED recovery model staan, dan wordt automatisch na de full backup een log backup gemaakt.

### **Naamgeving backupbestanden**

De namen van de backupbestanden zijn:

- `{dbname}_backup_{datetime}.{ext}` (bij niet gesplitte bestanden)
- `{dbname}_backup_{datetime}_{seq}.{ext}` (bij gesplitte bestanden)

`{datetime}` = 'yyyy\_mm\_dd\_hhMMss\_nanosec'

`{seq}` = family sequence number (1,2,3...)

`{ext}` = 'bak' voor full database backups, 'dif' voor differential backups en 'trn' voor transaction log backups.

De `prc_check_database_backup` procedure laat de database backup status zien. Deze procedure heeft 2 verplichte parameters, onderstaand voorbeeld is de meest simpele aanroep van deze procedure:

```
EXEC [dbo].[prc_check_database_backup]
@error_report_path='C:\Data\MSSQL\error_report',
@email_address='g.mantel@gmail.com';
```

Verplichte parameters:

Parameter	Type	Standaard waarde	Betekenis
@error_report_path	VARCHAR(255)	"	Hier wordt het backup check (error) rapport naar toe geschreven (in HTML formaat).
@email_address	NVARCHAR(100)	"	Email adres. Hier wordt het backup check (error) rapport naar toe verzonden.

Optionele parameters:

Parameter	Type	Standaard waarde	Betekenis
@log_backup_interval_minutes	INT	60	Gebruikte job schedule interval voor log backups (15, 30 of 60 minuten).
@system_backup_interval_hours	INT	4	Gebruikte job schedule interval voor system database backups (1, 2, 4, 6, 8, 12 of 24 uur).

Als er foutmeldingen zijn, worden deze verwerkt tot een Backup check (error) report. Dit rapport wordt weggeschreven als .html bestand naar het opgegeven pad (@error\_report\_path) en gemailed naar het opgegeven mailadres (@email\_address).

Het rapport bestaat uit een tabel met database namen en laatste backup datums. In de volgende situaties wordt het veld Error\_message gevuld.:

- Een database heeft geen zgn. "start backup" (dit betekent dat er geen begin is van de backupketen).
- De laatste gemaakte Full-, Differential of Transaction log backup is te oud.

Voor de berekening of de laatst gemaakte backup te oud is, wordt uitgegaan van de volgende regels:

- De wekelijks gemaakte Full backup mag niet ouder zijn dan 180 uur.
- De dagelijks gemaakte Diff backup mag niet ouder zijn dan 36 uur. Ook een full backup volstaat.
- De laatst gemaakte Log backup mag niet ouder zijn dan 1.5 keer de opgegeven @log\_backup\_interval\_minutes minuten. Ook een Full- of Diff backup volstaat.
- De laatst gemaakte systemdb (Full) backup mag niet ouder zijn van 1.5 keer de opgegeven @systemdb\_max\_hours uur.

### Database die niet ONLINE staan

Alleen databases die ONLINE staan doen mee in de backup check. Bijvoorbeeld, databases die door hun bijzondere grootte langdurig in RESTORING state staan, zijn dan wel qua log backup onbeschermd, maar dat wordt niet gemeld.

## **Databases in SIMPLE recovery model**

Van databases in SIMPLE recovery model worden geen log backups gemaakt. De backup check houdt hier rekening mee.

## **Logship databases**

Secondary logship databases en andere databases die Standby staan doen niet mee in de backup check. Primary logship databases wel. Let bij Primary Logship databases op dat de log backup frequentie van de LS backup job niet lager is dan die van de opgegeven `@log_backup_interval_minutes` van de backup check. In de praktijk hebben LS backup jobs een frequentie van 15 min, hetgeen ook de kleinste waarde is voor de `@log_backup_interval_minutes`.

## **HADR databases**

Van HADR databases wordt verwacht dat Full- en Differential backups worden gemaakt van de Primary replica, en de Log backups van de zgn. Backup\_preferred replica. De backup checks zijn dan ook hierop gebaseerd.

## **Model databases**

De system database 'model' mag in FULL of BULK\_LOGGED recovery model staan. Als dat het geval is, maakt de systemdb backup procedure na de full backup ook nog een log backup. Dit is om te voorkomen dat er op den duur te veel VLF's ontstaan. Voor het terugzetten van de model database is echter alleen de laatste full backup nodig. De backup check houdt voor de berekening van overschrijding van de log backup ouderdom gebruik van de opgegeven `@systemdb_max_hours`, en niet de `@log_backup_interval_minutes`.

## **READONLY Databases**

Van databases die READONLY staan kunnen full- en diff backups worden gemaakt. Als database ook nog in FULL of BULK\_LOGGED recovery model staat, kan ook een log backup worden gemaakt. Echter, deze log backups krijgen een COPY\_ONLY flag. Deze log backups zitten niet in de backupketen, en zijn daardoor niet te gebruiken voor een recover actie. De backup check geeft een foutmelding in zo'n situatie, en adviseert om de READONLY database in SIMPLE recovery model te zetten.

## 3 Herstellen databases

### 3.1 SQL Server crash recovery

Bij het uitvoeren van een SQL Server crash recovery kan gebruik worden gemaakt van de laatst gemaakte backups van de system databases.

Een SQL Server crash recovery bestaat uit de volgende stappen:

#### 3.1.1 Herinstallatie van SQL Server

Een her-installatie van SQL Server komt neer op de volgende stappen:

- a. Stop evt. nog lopende SQL services.
- b. Verwijder de map C:\Program Files\Microsoft SQL Server
- c. Verwijder de registersleutels  
Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server  
Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQLServer
- d. Herstart na registerwijzigingen altijd de machine.
- e. Installeer SQL Server met dezelfde versie, editie en patchlevel.

#### 3.1.2 Restore de master database

Hoe je de master database uit een backup terugzet wordt uitgelegd in het Microsoft artikel:

<https://learn.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-the-master-database-transact-sql?view=sql-server-ver16>

De restore van de master database komt neer op 4 stappen:

- a. Stop de SQL Server service en start de instance m.b.v. sqlsvr.exe handmatig op in single user-mode met speciale parameters.
- b. Connect SQL Server met de sqlcmd tool
- c. Run een restore commando met WITH REPLACE in sqlcmd, b.v.:  
**RESTORE DATABASE [master] FROM DISK='...' WITH REPLACE;**  
Gebruik het laatste (full) backup bestand.  
Als de master is gerestored wordt SQL Server automatisch gestopt en sqlcmd beëindigd.
- d. Start de SQL Server service normaal op.

#### 3.1.3 Restore de msdb database

De restore van de msdb database komt neer op 3 stappen:

- a. Stop de SQL Server agent service.
- b. Run een restore commando met WITH REPLACE in de SQL Server Management studio (onder sa), b.v.:  
**RESTORE DATABASE [msdb] FROM DISK='...' WITH REPLACE;**  
Gebruik het laatste (full) backup bestand.
- c. Start de SQL Server agent service.

### 3.1.4 Restore de model database

De restore van de model database kan evenals de msdb in de SQL Server management studio worden uitgevoerd.

- a. Run een restore commando met WITH REPLACE in de SQL Server Management studio (onder sa), b.v.:

```
RESTORE DATABASE [model] FROM DISK='...' WITH REPLACE;
```

Gebruik het laatste (full) backup bestand.

## 3.2 Genereren van een survive script

Te voorbereiding op toekomstige herstel acties, is het zinvol om een script bij de hand te hebben die een database (TMP\_SURVIVE) kan aanmaken waarin de benodigde restore procedures zitten.

Voer het volgende commando uit nadat je gerritmantel procedures hebt geïnstalleerd of hebt bijgewerkt:

```
EXEC dbo.prc_dump_survive_script  
@dump_path='C:\Data\MSSQL\survive'
```

Het dump path is een voorbeeld en moet van te voren bestaan. Bewaar het gegenereerde script (survive\_script.sql) goed. Je hebt dit script nodig als je user databases uit hun backup bestanden wilt terugzetten na een server crash.

## 3.3 Restore user databases

De user databases kunnen (per database) worden teruggezet door middel van de prc\_restore\_database procedure. Er is geen procedure die integraal voor alle user databases de restores meteen uitvoert, dat zou in mijn ogen gevaarlijk zijn. Het is de bedoeling dat er een script wordt gegenereerd en dat daarna het script aandachtig wordt beoordeeld voordat de restores van de user databases wordt uitgevoerd.

Voordat je dit kunt doen moet je eerst het kip-ei probleem verhelpen. Je hebt geen user databases, en (dus) ook geen DBMAINT database (b.v.) waarin de gerritmantel procedures zitten. Maak nu gebruik van het survive script (survive\_script.sql) (zie H 3.2 “Genereren van een survive script”). Run het survive script in de context van de master database. De database TMP\_SURVIVE wordt aangemaakt (inclusief restore procedures) en je context wordt verlegd naar de TMP\_SURVIVE database. Ga hieruit vandaan verder met de volgende stap.

Maak het restore script (m.b.v. de Management Studio) aan m.b.v. het volgende commando (voorbeeld):

```
EXEC [dbo].[prc_restore_userdb_script_only]  
@backup_path='D:\MSSQL\backup';
```

Op basis van aanwezige backup bestanden worden de scriptregels aangemaakt. Of de backup bestanden al of niet in een subdirectory staan wordt automatisch gedetecteerd. De scriptregels verschijnen in de output grid en zijn te kopiëren naar een nieuwe Query

Window op de TMP\_SURVIVE database. Het script zal per database de volgende regel bevatten:

```
EXEC dbo.prc_restore_database @dbname='...', @backup_path='...',  
@subdir=...;
```

**LET OP!** Backupbestanden die nog aanwezig zijn van niet-meer-bestaande databases resulteren ook scriptregels. Remark deze scriptregels om te voorkomen dat oude databases worden teruggezet.

De procedure maakt geen gebruik van informatie uit de msdb database, zodat ook user databases hersteld kunnen worden na een herinstallatie van SQL Server zonder SQL Server crash recovery.

Run de regels (één voor één, of meerdere tegelijk). Bij het uitvoeren van de prc\_restore\_database commando's, zullen de laatst aanwezige backupketens (bestaande uit full-, differential- en logbackup bestanden van de database) gebruiken om de database terug te zetten. De restore wordt afgerond met WITH RECOVERY, d.w.z. de gerestorede database is open en klaar voor gebruik.

Als prc\_restore\_database procedure zonder @option parameter wordt aangeroepen, wordt @option='RESTORE' aangenomen, en wordt de database restore meteen uitgevoerd. Er zijn twee andere opties die niet restoren maar alleen informatie geven:

@option='SCRIPT' -> maak restore scriptregels voor die database aan.

@option='REPORT' -> maak een lijst met backup bestanden voor die database aan (in de output grid). De backupbestanden die gebruikt zullen worden voor de restore zijn gemarkeerd.

**LET OP!** Gooi de TMP\_SURVIVE database weer weg, als de restore akties zijn afgerond.