Maintenance des bases de données de GEOS Pro

Recommandations pour PostgreSQL et SQL Server

Ce document décrit les mesures de maintenance recommandées pour les bases de données GEOS Pro PostgreSQL et Microsoft SQL Server. Une maintenance régulière des bases de données permet d'éviter les baisses de performances et la croissance incontrôlée des données en environnement de production.

Version du 16.11.2026

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



Table des matières

Introduction 3
Périmètre d'application 3
Vider la table ModificationLog 4
Gestion de la croissance de la base de données (PostgreSQL) 7
Entretien des index (SQL Server) 8
Entretien des index (PostgreSQL) 11
Mise à jour des statistiques (SQL Server) 12
Mise à jour des statistiques (PostgreSQL) 14
Autres mesures 16
Ordre des mesures recommandées 16
Annexe 1 : Script de maintenance hebdomadaire 17
Annexe 2 : Script de maintenance semestriel 19

Introduction

GEOS Pro est une application qui repose sur des bases de données relationnelles. Pour garantir de bonnes performances, il ne suffit donc pas que l'application soit performante, mais la base de données sous-jacente doit également l'être. Ce n'est que lorsque toutes les tables, tous les index et l'allocation physique de la mémoire sont en « bon état de santé » que les fonctions correspondantes de la base de données peuvent fonctionner de manière performante.

Ce document présente les recommandations formulées par Hexagon Technology Center GmbH concernant la maintenance régulière des bases de données SQL de GEOS Pro.

Périmètre d'application

Ce document contient des recommandations pour les bases de données PostgreSQL et SQL Server. Bien que d'autres bases de données puissent nécessiter des tâches de maintenance similaires, les procédures spécifiques diffèrent souvent.

L'utilisation des méthodes et des scripts de maintenance recommandés ci-dessous relève de la responsabilité de l'utilisateur. Hexagon Technology Center GmbH décline toute responsabilité.

Les méthodes et scripts de maintenance recommandés doivent fréquemment être adaptés à l'infrastructure système et aux concepts de gestion des données du client.

Vider la table ModificationLog

Informations générales

La table ModificationLog dans SQL Server (ou la table modificationlog dans PostgreSQL) enregistre toutes les modifications (Insert, Update, Delete) effectuées par GeoMedia dans les différentes tables de la base de données. Le contenu de cette table peut donc augmenter rapidement, en fonction de l'activité d'édition d'un opérateur GEOS Pro. Cependant, lorsque cette table contient un grand nombre d'enregistrements, cela a un impact négatif sur les performances de certaines fonctions de GeoMedia.

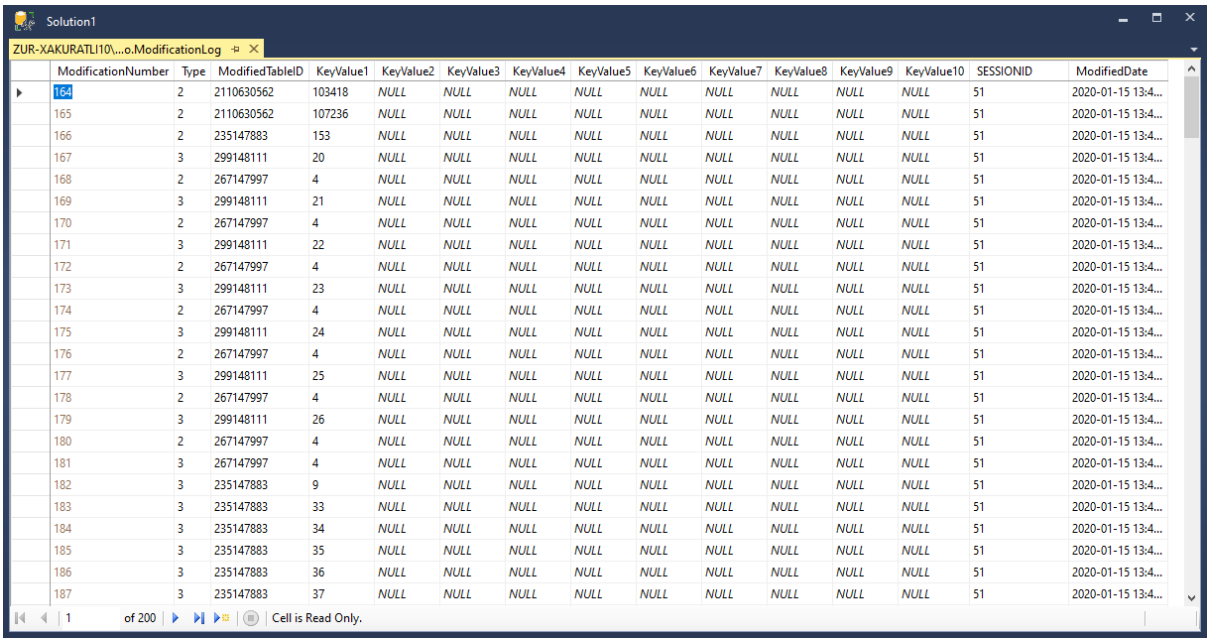
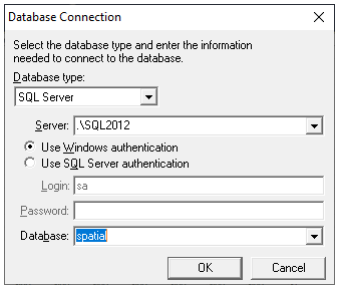


Figure 1 : Table ModificationLog de GeoMedia dans SQL Server Management Studio

Remarque : la table ModificationLog se trouve dans le schéma gdo de PostgreSQL

Recommandation

Vérification de sécurité : chaque opération de maintenance commence par une sauvegarde complète.

Mesure	Suppression régulière du contenu de Modification Log
Procédure (manuelle)	1. Ouvrir l'utilitaire de base de données de GeoMedia  2. Exécuter la fonction *Clear Modification Log*
Script SQL Server	`USE [<database>]` `TRUNCATE TABLE dbo.ModificationLog`
Script PostgreSQL	`TRUNCATE TABLE gdo.modificationlog`
Fréquence	Chaque semaine

Gestion de la croissance de la base de données (SQL Server)

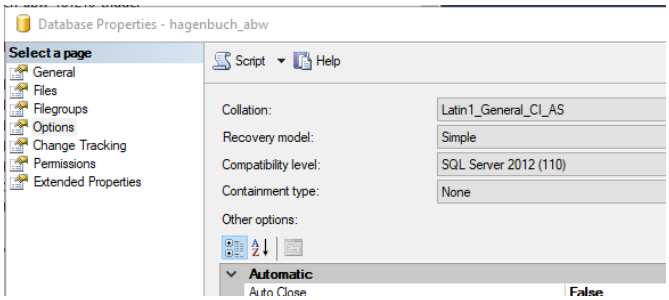
Informations générales

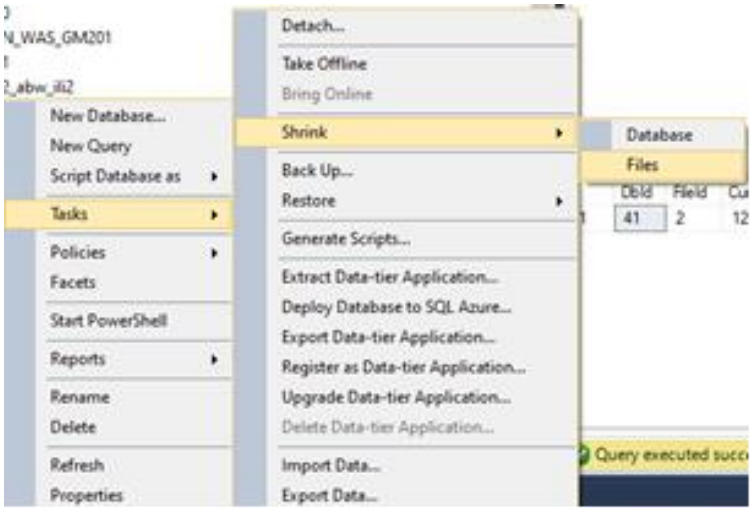
Microsoft SQL Server enregistre toutes les modifications apportées à la base de données dans un fichier journal, afin que ces modifications puissent être annulées à tout moment. Selon les paramètres définis, la taille du fichier journal augmente progressivement. Au fil du temps, il peut arriver que le fichier de la base de données atteigne par exemple 2 Go, tandis que le fichier journal pèse 20 Go ! Cela sollicite inutilement le serveur et ralentit parfois l'accès aux données.

Les paramètres suivants influencent la taille d'une base de données :

- Volume de données / nombre d'insertions
- Taille initiale des fichiers de base de données
- Modèle de récupération sélectionné (Full, Simple)

Recommandation

Mesure	Modifier le modèle de récupération sur Simple
Procédure (manuelle)	1. Ouvrez SQL Server Management Studio, puis accédez aux propriétés de la base de données. 2. Dans l'onglet Options, modifiez le modèle de récupération pour le régler sur *Simple*. 
Procédure (Script)	`ALTER DATABASE <database> SET RECOVERY SIMPLE`
Fréquence	Une fois

Mesure	Réduire la taille de la base de données
Procédure (manuelle)	1. Dans SQL Server Management Studio, ouvrez le menu contextuel de la base de données concernée et sélectionnez la commande permettant de réduire la taille des fichiers.  2. Réduire successivement les fichiers de type *Data* et *Log*

Procédure (Script)	`USE [<database>]`
	`GO`
	`DBCC SHRINKFILE (N'<logicalname>', 0, TRUNCATEONLY)`
	`GO`
	
	Le <logicalname> apparaît dans la section Properties Files (1re colonne). Dans ce script, les deux noms logiques doivent être indiqués.

Fréquence	Chaque semaine

Gestion de la croissance de la base de données (PostgreSQL)

Informations générales

Contrairement à SQL Server, où le fichier journal (.ldf) peut grossir indéfiniment sans intervention manuelle (Shrink/Simple Mode), PostgreSQL utilise un système de journalisation circulaire appelé WAL (Write Ahead Log). PostgreSQL écrase automatiquement les anciens journaux dès que les modifications sont enregistrées en toute sécurité dans la base de données. Le gonflement du disque dur par les fichiers journaux est ainsi évité de manière native.

Recommandation

Remarque : le démon Autovacuum est activé par défaut. Il nettoie en arrière-plan les enregistrements supprimés ou modifiés, sans interrompre le fonctionnement du système.

Mesure	Limitation de la taille du journal (WAL) - Définir le paramètre max_wal_size dans le fichier postgresql.conf
But	Détermine l'espace maximal que les journaux peuvent occuper avant que PostgreSQL ne libère de l'espace pour de nouvelles données.
Procédure (Script)	`ALTER SYSTEM SET max_wal_size = '2GB';` `SELECT pg_reload_conf();`
Fréquence	Lors de l'installation

Mesure	Libération d'espace disque
Procédure (Script)	`VACUUM FULL;` Cette commande verrouille temporairement la table. Elle n'est nécessaire qu'en cas de travaux de maintenance exceptionnels.
Fréquence	Très rarement (uniquement après des suppressions massives de données).

Entretien des index (SQL Server)

Informations générales

Les index de base de données constituent une structure d'index distincte de la structure des données, qui accélère la recherche et le tri selon certains champs. Les accès en lecture à la base de données bénéficient d'un index, tandis que les accès en écriture sont ralentis par les index.

Les index existants sont mis à jour en continu grâce à la mise à jour des données. Cependant, les nouvelles valeurs d'index ne peuvent pas toujours être gérées à l'emplacement de stockage approprié. De ce fait, la « fragmentation » des index ne cesse de s'accroître. Selon le degré de fragmentation des index d'une table, les performances des requêtes sur une ou plusieurs tables peuvent s'en trouver fortement dégradées.

Le degré de fragmentation des index d'une base de données peut être affiché à l'aide de l'instruction SQL suivante :

```
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
```

Il existe deux façons d'améliorer, voire d'éliminer, la fragmentation des index.

Reconstruire l'index

Cette opération supprime l'index et le reconstruit entièrement. Cela permet de rétablir l'ordre et de résorber la fragmentation, car toutes les entrées sont réorganisées de manière cohérente et consécutive. La fragmentation est ainsi éliminée et de l'espace disque libre est libéré, qui peut ensuite être réutilisé. Cette opération est relativement « coûteuse », car elle nécessite d'abord de supprimer l'ancien index, puis de reconstruire entièrement le nouvel index à partir des données de la table.

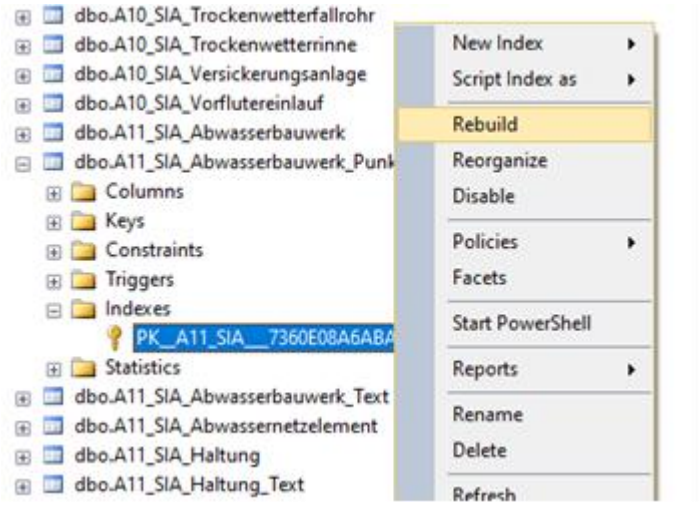
Réorganiser l'index

La réorganisation d'un index mobilise un minimum de ressources système ; elle consiste à vérifier si les entrées se trouvent toujours à la bonne position et, si nécessaire, à les réorganiser afin de rétablir un ordre optimal et logiquement correct.

Microsoft recommande de ne reconstruire ou réorganiser les index qu'en cas de fragmentation importante.

Degré de fragmentation	Mesure
>5% bis < =30%	Réorganiser
>30%	Reconstruire

Recommandation

Mesure	Réorganiser ou reconstruire les index conformément aux recommandations de Microsoft
Procédure (manuelle)	1. Afficher le niveau de fragmentation des index d'une base de données à l'aide du script ci-dessus 2. Reconstruire chaque index dont le niveau de fragmentation est supérieur à 30 %. Pour ce faire, ouvrez le menu contextuel de l'index concerné et sélectionnez la fonction *Rebuild*.  3. Cliquez sur OK pour lancer la reconstruction 4. Tout index présentant un taux de fragmentation > 5 % et < 30 % doit être réorganisé. Pour ce faire, ouvrez le menu contextuel de l'index concerné et sélectionnez la fonction *Reorganize*. 5. Cliquez sur OK pour lancer la réorganisation

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

Fréquence	Semestriel

- La création ou la réorganisation de petits index ne réduit souvent pas, ou très peu, la fragmentation. Cela s'explique par l'organisation du stockage physique au moyen de « blocs ». Une forte fragmentation des petits index n'a aucune incidence sur les performances.

- La maintenance des index peut également être effectuée à l'aide d'une procédure SQL qui réorganise ou reconstruit tous les index d'une base de données en fonction de leur degré de fragmentation. Une telle procédure est présentée dans le script d'exemple figurant à l'annexe 2.

Entretien des index (PostgreSQL)

Informations générales

- Les index (tels que l'index GiST pour les données géographiques) sont des structures distinctes de la structure des données, qui accélèrent la recherche et le filtrage dans GeoMedia. Chaque opération d'écriture (insertion/mise à jour/suppression) doit être répercutée dans l'index.
- Fragmentation (gonflement) vs. Autonettoyage Contrairement à SQL Server, qui souffre de fragmentation physique ou morcellement, PostgreSQL utilise le principe MVCC. Les versions obsolètes des données (lignes mortes) sont continuellement supprimées par le démon Autovacuum. Cela permet de maintenir l'index compact et performant, sans qu'aucune intervention manuelle ne soit nécessaire.
- Le mécanisme d'auto-réparation, grâce à Autovacuum, l'espace libéré au sein d'un index est immédiatement réutilisé. Il n'est donc pas nécessaire de surveiller les taux de fragmentation de 5 % ou 30 % pour décider d'une reconstruction manuelle.

Recommandation

Mesure	Maintenance automatisée des index
Procédure	Autovacuum gère la compacité des index de manière entièrement automatique en arrière-plan.
Procédure (manuelle)	Normalement pas nécessaire. Contrairement à SQL Server Express Edition, il n'est pas nécessaire de planifier des reconstructions tous les six mois.

Procédure (exception)	Uniquement après des suppressions massives de données (exemple : un nettoyage de 50 % de la base de données), afin de libérer de l'espace de stockage physique.
	`REINDEX TABLE CONCURRENTLY <nom_de_table>;`
	`REINDEX DATABASE CONCURRENTLY <nom_de_la_base_de_données>;`

Fréquence	Entièrement automatique. Une intervention manuelle n'est généralement nécessaire que tous les deux ou trois ans, ou à la suite de modifications structurelles importantes des données.

Mise à jour des statistiques (SQL Server)

Informations générales

Les statistiques sont des objets SQL Server qui contiennent des mesures relatives au nombre et à la répartition des données au sein d'une colonne. Elles sont utilisées pour optimiser le traitement des requêtes (détermination du meilleur plan d'exécution). On en distingue deux types :

Statistiques des index

Celles-ci sont créées automatiquement par SQL Server lors de la création d'un index et portent le même nom.

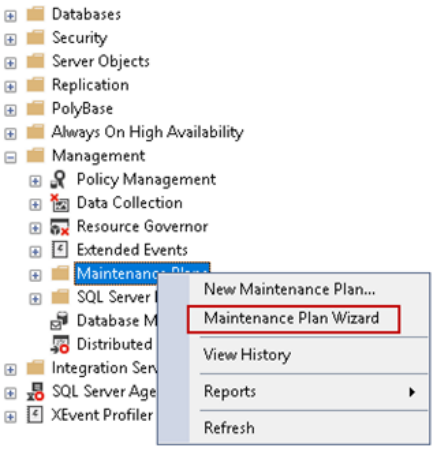
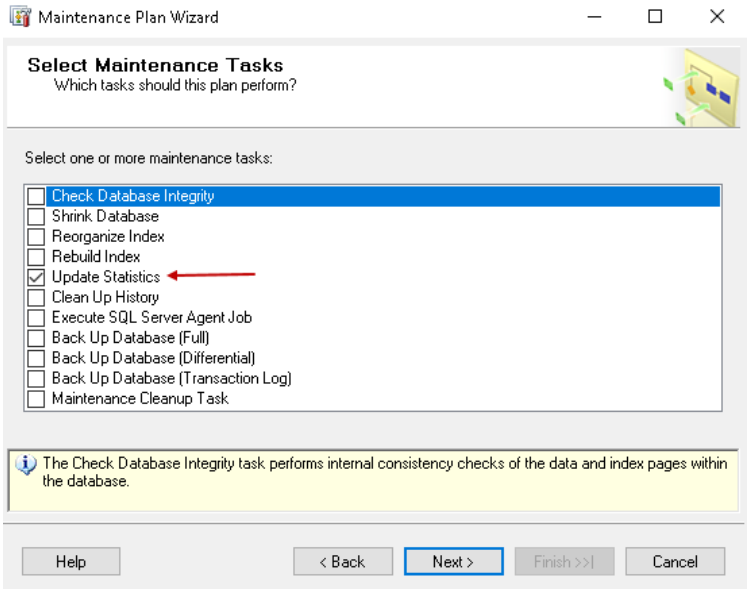
Statistiques des colonnes

Celles-ci sont créées automatiquement pour chaque colonne d'une table si l'option *Auto Create Statistics* est définie sur *True*.

Lorsque les données d'une base de données changent, les statistiques ne sont pas mises à jour en permanence et peuvent donc contenir des informations obsolètes. Cela peut avoir pour conséquence que les requêtes ne soient plus traitées selon les plans d'exécution optimaux. Des statistiques obsolètes nuisent donc aux performances.

SQL Server propose certes deux options : *Auto Create Statistics* et *Auto Update Statistics*. Cette dernière option demande à SQL Server de mettre à jour automatiquement les statistiques, mais uniquement en cas d'écart important entre les statistiques et le contenu actuel de la table.

Recommandation

Mesure	Mettre à jour les statistiques à l'aide d'un script de maintenance
Procédure (manuelle)	La procédure manuelle n'est pas disponible dans l'édition Express de SQL Server. 1. Ouvrir l'assistant de plan de maintenance dans SQL Server Management  Studio 2. Sélectionnez *Maintenance Task Update Statistics* puis suivez les instructions de l'assistant. 
Procédure (Script)	``` USE [<database>] GO EXEC sp_updatestats; ```
Fréquence	Semestriel

Mise à jour des statistiques (PostgreSQL)

Informations générales

Dans PostgreSQL, les statistiques sont des métadonnées essentielles utilisées par le planificateur de requêtes. Elles contiennent des informations sur la répartition des données dans les tables (par exemple, le nombre de bâtiments ou de biens immobiliers). Sur la base de ces données, PostgreSQL détermine s'il est préférable d'utiliser un index (tel que GiST) ou si un balayage séquentiel est plus rapide.

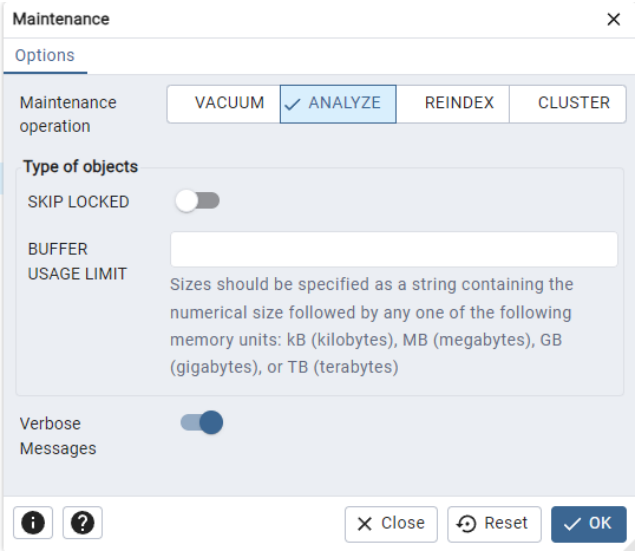
Statistiques d'index et statistiques de colonnes

PostgreSQL crée et met à jour ces statistiques de manière entièrement automatique pour chaque table et chaque index, dès que des données sont insérées, supprimées ou modifiées.

Problématique des statistiques obsolètes

Si, après une importation massive (par exemple, de nouvelles données cadastrales officielles concernant tout un canton), les statistiques ne sont pas mises à jour, la base de données « considère » que les tables sont encore vides ou de petite taille. Conséquence : GeoMedia devient extrêmement lent, car le planificateur choisit des chemins de recherche inefficaces.

Recommandation

Mesure	Maintenir les statistiques à jour avec ANALYZE
Procédure (manuelle)	Cliquez avec le bouton droit sur la base de données -> Maintenance -> Sélectionnez ANALYZE. 

Procédure (Script)	`--Pour l'ensemble de la base de données`
	`ANALYZE;`
	Ou
	`-- Spécifiquement pour une table après une importation`
	`ANALYZE <table>;`

Fréquence	Entièrement automatique grâce à Autovacuum. Il n'est recommandé de lancer manuellement cette opération qu'immédiatement après des importations massives de données, afin de bénéficier immédiatement de toutes les performances.

Autres mesures

- Supprimer régulièrement les utilisateurs qui ne sont plus nécessaires
- Supprimer régulièrement les classes d'objets qui ne sont plus nécessaires dans la base de données
- Désactiver systématiquement le tri des colonnes GRIVIS-GEOS

Ordre des mesures recommandées

Les mesures recommandées ne produiront leur effet optimal que si elles sont mises en œuvre dans le bon ordre.

SQL Server

Chaque semaine

1. Effectuer une sauvegarde des données
    a. La sauvegarde des données doit impérativement être effectuée au début, respectivement avant l'exécution des scripts de maintenance.
2. Vider les tables ModificationLog et G5ERR
3. Réindexer la base de données
4. Réduire la taille de la base de données (compactage de la base de données)

Semestriel

1. Réorganiser ou reconstruire les index
2. Recalculer les statistiques de la base de données

PostgreSQL

De nombreuses tâches ne nécessitent aucune intervention dans PostgreSQL et s'exécutent de manière entièrement automatique, comme l'Auto-analyse et l'Autovacuum.

Hebdomadaire

1. Effectuer une sauvegarde des données
    a. La sauvegarde des données doit impérativement être effectuée au début, respectivement avant l'exécution des scripts de maintenance.
2. Vider les tables ModificationLog et G5ERR

Semestriel :

1. Réindexer la base de données
2. Optimiser la base de données
3. Recalculer les statistiques

Annexe 1 : Script de maintenance hebdomadaire

SQL Server

Automatiser SQL Server au moyen de tâches SQL Server Agent
Automatiser SQL Server Express au moyen de fichiers batch

```
/****************************************/
/*** SCRIPT DE MAINTENANCE GEOS Pro     ***/
/*** SQL Server                         ***/
/*** Octave                             ***/
/*** Script de maintenance hebdomadaire ***/
/****************************************/
/*** L'utilisateur assume l'entière     ***/
/*** responsabilité de l'utilisation de ***/
/*** ce script.                         ***/
/****************************************/

-- <database> le nom de la base de données
-- <datapath> le chemin de stockage du backup
-- <backupname> le nom souhaité pour le backup
-- <logicalname> le nom des fichiers de la base de données
/****************************************/

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

-- Vide la table ModificationLog
TRUNCATE TABLE [<database>].[dbo].[ModificationLog];
GO

-- Vide la table A00_G5ERR
TRUNCATE TABLE [<database>].[dbo].[A00_G5ERR];
GO

-- Réindexe la base de données
USE [<database>];
GO
EXEC sp_MSforeachtable 'ALTER INDEX ALL ON ? REBUILD';
GO

-- Réduit la taille du fichier de données
USE [<database>];
GO
DBCC SHRINKFILE (N'<logicalname>' , 0, TRUNCATEONLY);
GO

-- Réduit la taille du fichier log
USE [<database>];
GO
DBCC SHRINKFILE (N'<logicalname>_log' , 0, TRUNCATEONLY);
GO
```

PostgreSQL

Intégrer le script dans un fichier batch

```
/*****************************************/
/*** SCRIPT DE MAINTENANCE GEOS Pro     ***/
/*** PostgreSQL                         ***/
/*** Octave                             ***/
/*** Script de maintenance hebdomadaire ***/
/*****************************************/
/*** L'utilisateur assume l'entière     ***/
/*** responsabilité de l'utilisation de ***/
/*** ce script.                         ***/
/*****************************************/

/*****************************************/
-- Vide Les tables log de GeoMedia
TRUNCATE TABLE gdo.modificationlog;
TRUNCATE TABLE public.a00_g5err;

-- Met à jour les statistiques et libère l'espace de stockage
VACUUM ANALYZE;
```

Annexe 2 : Script de maintenance semestriel

SQL Server

Intégration à une tâche SQL Server Agent ou à un fichier batch.

```
/****************************************/
/*** SCRIPT DE MAINTENANCE GEOS Pro     ***/
/*** SQL Server                         ***/
/*** Octave                             ***/
/*** Script de maintenance semestriel   ***/
/****************************************/
/*** L'utilisateur assume l'entière     ***/
/*** responsabilité de l'utilisation de ***/
/*** ce script.                         ***/
/****************************************/


/****************************************/
-- <database> le nom de la base de données
/****************************************/

-- Définition des variables
DECLARE @Database VARCHAR(255)
DECLARE @Table VARCHAR(255
DECLARE @FragValue INT
DECLARE @ID_DB AS SMALLINT
DECLARE @ID_TABLE AS INT
DECLARE @cmd NVARCHAR(MAX)
DECLARE @fillfactor INT = 90

-- Définition du curseur
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'''

   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;
      SELECT TOP(1) @FragValue = CONVERT(INT, 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
      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) + ')'
          EXEC (@cmd)
      END
      ELSE IF (@FragValue > 5)
      BEGIN
```

```
            PRINT 'REORGANIZE='+CONVERT(NVARCHAR(10),@FragValue)+' Table:' +@Table
            SET @cmd = 'ALTER INDEX ALL ON ' + @Table + ' REORGANIZE'
            EXEC (@cmd)
         END
         FETCH NEXT FROM TableCursor INTO @Table
      END
      CLOSE TableCursor
      DEALLOCATE TableCursor
      FETCH NEXT FROM DatabaseCursor INTO @Database
END
CLOSE DatabaseCursor
DEALLOCATE DatabaseCursor

-- Recalculer les statistiques
USE [<database>];
GO
EXEC sp_updatestats;
```

PostgreSQL

```
/****************************************/
/*** SCRIPT DE MAINTENANCE GEOS Pro     ***/
/*** PostgreSQL                         ***/
/*** Octave                             ***/
/*** Script de maintenance semestriel   ***/
/****************************************/
/*** L'utilisateur assume l'entière     ***/
/*** responsabilité de l'utilisation de ***/
/*** ce script.                         ***/
/****************************************/

/****************************************/
-- <database> le nom de base de données correct
-- <table> le nom de la table à analyser
/****************************************/

-- Réorganise / reconstruit les index.
REINDEX DATABASE "<database>";

-- Uniquement lorsque la base de données a fortement augmenté en taille
VACUUM FULL VERBOSE;

-- Analyse de la base de données.
ANALYZE VERBOSE;

-- Contrôle de l'intégrité PostGIS
-- Vérifie que les géométries sont conformes aux normes OGC.
SELECT gid, ST_IsValidReason(geom)
FROM <table>
WHERE ST_IsValid(geom) IS FALSE;
```

À propos d’Octave

Octave fournit des logiciels stratégiques qui permettent aux entreprises de prendre des décisions éclairées à chaque étape du cycle de vie des actifs, de la conception et du développement à l’exploitation et à la protection. Cela est particulièrement essentiel lorsque la performance, la sécurité et la fiabilité sont indispensables et que l’échec n’est pas une option. En transformant des données opérationnelles complexes en informations exploitables, Octave associe une expertise approfondie, une connaissance des conditions réelles et une vision globale de l’entreprise afin d’améliorer la performance, la résilience et la réactivité.

En 2025, Hexagon AB a lancé la scission de ses divisions Asset Lifecycle Intelligence ainsi que Safety, Infrastructure & Geospatial. Avec ETQ et Bricsys, cette opération a donné naissance à Octave, qui exercera désormais ses activités en tant qu’entreprise indépendante. Octave continue de s’appuyer sur une expertise reconnue, une vaste expérience et des relations solides construites au fil de plusieurs décennies.

Octave emploie plus de 7 000 collaborateurs dans 45 pays. Pour en savoir plus, rendez-vous sur **octave.com** et suivez-nous sur **LinkedIn.**

© 2026 Intergraph Corporation et/ou ses sociétés affiliées. Tous droits réservés. Le logo Octave est une marque d’Intergraph Corporation. Les autres marques et noms de produits sont des marques déposées de leurs propriétaires respectifs.

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

Hexagon Technology Center GmbH
Octave | Succursale de Zürich
Flurstrasse 55
8048 Zürich
Suisse
T: +41 43 322 46 46
www.octave.com

