utiliser la vue système sys.dm\_db\_index\_physical\_stats pour surveiller la fragmentation des index.

Voici un script pour identifier la fragmentation des index d'une base de données :

```sql

USE [NomDeVotreBaseDeDonnées];

GO

SELECT

DB\_NAME() AS DatabaseName,

t.name AS TableName,

i.name AS IndexName,

ips.index\_type\_desc AS IndexType,

ips.avg\_fragmentation\_in\_percent AS FragmentationPercentage,

ips.page\_count AS PageCount

FROM

sys.dm\_db\_index\_physical\_stats (NULL, NULL, NULL, NULL, 'SAMPLED') AS ips

INNER JOIN

sys.tables AS t ON ips.object\_id = t.object\_id

INNER JOIN

sys.indexes AS i ON ips.object\_id = i.object\_id AND ips.index\_id = i.index\_id

WHERE

ips.avg\_fragmentation\_in\_percent > 10 -- Seuil pour déterminer la fragmentation

AND ips.page\_count > 1000 -- Ignorer les index trop petits

ORDER BY

ips.avg\_fragmentation\_in\_percent DESC;

GO

Lorsque tu détectes de la fragmentation, tu peux soit réorganiser l'index (si la fragmentation est modérée), soit reconstruire l'index (si la fragmentation est plus sévère).

- \*Réorganiser les index\* : C'est une opération moins coûteuse qui consiste à réorganiser les pages de l'index.

- \*Reconstruire les index\* : Cette opération supprime et recrée l'index, ce qui est plus coûteux en termes de ressources, mais elle est plus efficace pour les index fortement fragmentés.

Voici un script qui te permet de réorganiser ou reconstruire les index en fonction du niveau de fragmentation :

sql

USE [NomDeVotreBaseDeDonnées];

GO

DECLARE @TableName NVARCHAR(255), @IndexName NVARCHAR(255), @SQL NVARCHAR(MAX);

-- Réorganiser ou reconstruire les index selon la fragmentation

DECLARE index\_cursor CURSOR FOR

SELECT

t.name AS TableName,

i.name AS IndexName,

ips.avg\_fragmentation\_in\_percent AS FragmentationPercentage

FROM

sys.dm\_db\_index\_physical\_stats (NULL, NULL, NULL, NULL, 'SAMPLED') AS ips

INNER JOIN

sys.tables AS t ON ips.object\_id = t.object\_id

INNER JOIN

sys.indexes AS i ON ips.object\_id = i.object\_id AND ips.index\_id = i.index\_id

WHERE

ips.avg\_fragmentation\_in\_percent > 10 -- Seuil de fragmentation

AND ips.page\_count > 1000 -- Ignorer les index trop petits

ORDER BY

ips.avg\_fragmentation\_in\_percent DESC;

OPEN index\_cursor;

FETCH NEXT FROM index\_cursor INTO @TableName, @IndexName, @FragmentationPercentage;

WHILE @@FETCH\_STATUS = 0

BEGIN

IF @FragmentationPercentage > 30

BEGIN

-- Reconstruire l'index si la fragmentation est supérieure à 30%

SET @SQL = 'ALTER INDEX [' + @IndexName + '] ON [' + @TableName + '] REBUILD';

END

ELSE

BEGIN

-- Réorganiser l'index si la fragmentation est entre 10% et 30%

SET @SQL = 'ALTER INDEX [' + @IndexName + '] ON [' + @TableName + '] REORGANIZE';

END

-- Exécution du script dynamique

EXEC sp\_executesql @SQL;

FETCH NEXT FROM index\_cursor INTO @TableName, @IndexName, @FragmentationPercentage;

END

CLOSE index\_cursor;

DEALLOCATE index\_cursor;

GO

Voici un script pour actualiser les statistiques de manière proactive pour toutes les tables d'une base de données :

sql

USE [NomDeVotreBaseDeDonnées];

GO

-- Actualiser les statistiques pour toutes les tables et index

EXEC sp\_updatestats;

GO

Cette commande met à jour les statistiques de toutes les tables de la base de données. Si tu souhaites actualiser les statistiques d'une table ou d'un index spécifique, tu peux utiliser :

sql

UPDATE STATISTICS [NomDeLaTable] [NomDeL'Index];

GO