tellora

Administration

SQL Server

2012

Index et statistiques

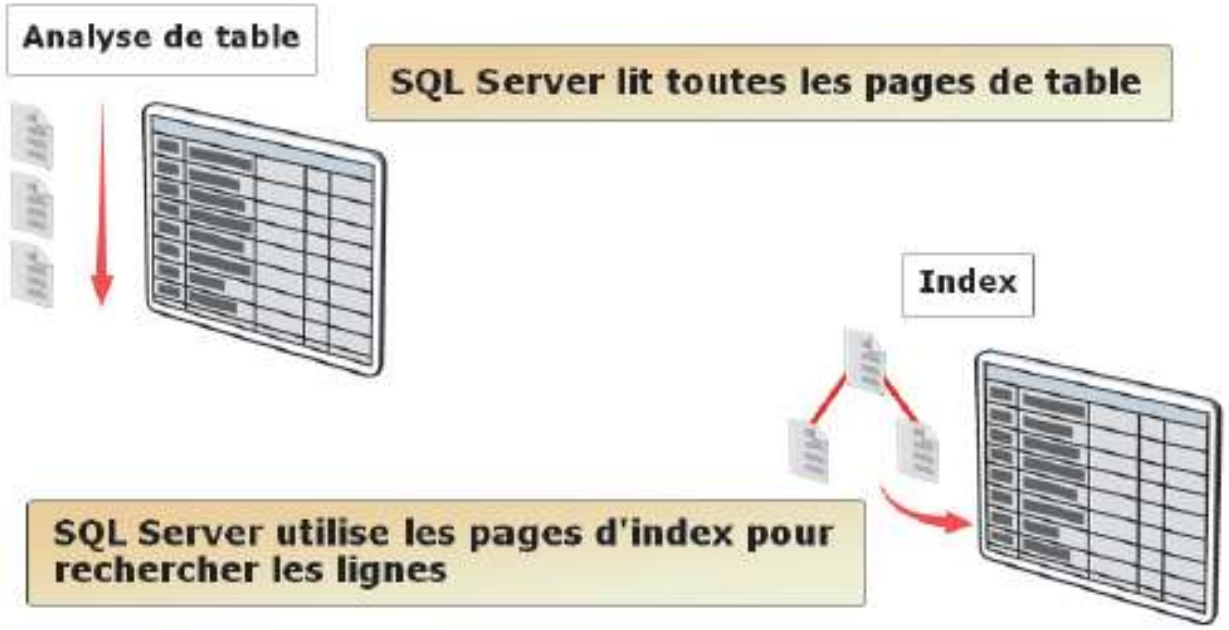
Index

- Pour accéder aux données stockées dans les tables SL Server peut :
  - Lire toutes les pages de la table
  - Utiliser un ou plusieurs index
- Il est plus lent de lire toutes les pages d'une table que d'utiliser un index
  - Les index accélères les traitements SQL SELECT
  - Mais augmentent les temps de réponse en Mise à Jour
- Occasionnellement SQL Server créé ses propres index temporaires pour améliorer la performance des requêtes comme cette action relève de l'optimiseur, ces index temporaires ne sont ne sont utilisés que pour améliorer les plan de requête si besoin

Index

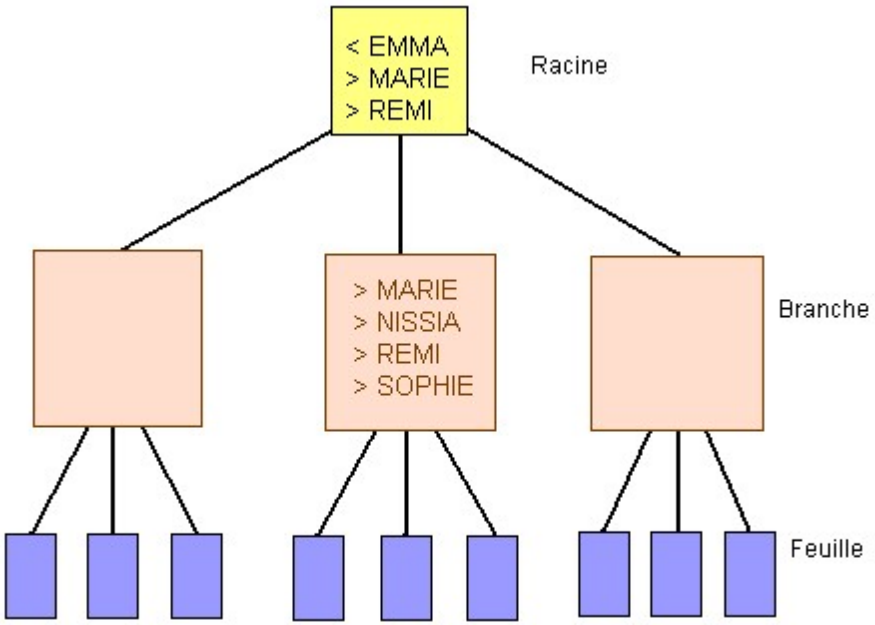
- SQL Server comprend d'autres types d'index :
  - la recherche en texte intégral intégrée (iFTS) utilise un type spécial d'index qui assure une recherche de texte flexible ;
  - les index spatiaux sont utilisés avec les types de données géométrie et géographie
  - les index XML primaires et secondaires contribuent à l'interrogation de données XML ;
  - les index columnstore sont généralement utilisés dans les grands entrepôts de données.
    - Les tables sont essentiellement en lecture seule lorsqu'il existe des index columnstore
  - les index cluster et non cluster (abordés dans ce cours)

Index


Analyse de table
SQL Server lit toutes les pages de table
Index
SQL Server utilise les pages d'index pour rechercher les lignes

Les index

- Organisation physique des index cluster et non cluster
  - Parcours de l'index de haut en bas et les branche d'un même niveau
  - L'index cluster possède des pages de données au niveau des feuilles
  - L'index non cluster possède un pointeur vers les lignes de données (tables) au niveau des feuilles



Index organisés en B-Tree*

Les index cluster

- Plutôt que d'inscrire les lignes d'une donnée en tant que segment, il est possible de concevoir les tables avec un ordre logique interne
  - Il s'agit d'un index cluster
- Une table dotée d'un index cluster possède un ordre prédéfini des lignes dans une page et des pages au sein de la table
  - Cet ordre est fondé sur une clé composée d'une ou plusieurs colonnes
  - On appelle communément cette clé une clé de clustering
- Comme les lignes d'une table ne peuvent être que dans un seul ordre, il ne peut exister qu'un seul index cluster sur une table
  - Les index cluster ont toujours un ID d'index = 1
  - SQL Server essaie d'aligner l'ordre physique et l'ordre logique lors de la création d'un index cluster, mais il peut se créer du désordre à mesure que les données sont modifiées
    - L'index et les pages de données sont liés selon une hiérarchie logique et également doublement liés sur toutes les pages au même niveau de la hiérarchie afin de faciliter l'analyse d'un index

Les index non cluster

- Un index non-cluster est un type d'index qui n'affecte pas la disposition des données dans la table à la manière dont le fait un index cluster
- Si la table sous-jacente est un segment (autrement dit, elle ne comporte pas d'index cluster), le niveau feuille d'un index non cluster contient des pointeurs vers l'endroit où sont stockées les lignes de données
  - Les pointeurs incluent un numéro de fichier, un numéro de page, et un numéro d'emplacement dans la page
- Si la table sous-jacente a un index cluster (autrement dit, les pages et les données sont logiquement liées dans l'ordre d'une clé de clustering), le niveau feuille d'un index non-cluster contient la clé de clustering
  - qui permet de faire une recherche dans les pages de l'index cluster pour localiser les lignes souhaitées

Les index

- SQL Server propose 2 types d'index
  - Les index organisés en cluster
  - Les index non organisés ou non cluster
- L'index organise ou non physiquement les données stockées dans les tables
  - L'index organisé organise les lignes de la table à laquelle il est rattaché
  - L'index non organisé n'a aucune incidence sur la structure physique de la table à laquelle il est rattaché
  - Ainsi dans un premier temps on définit les index organisés et dans un deuxième temps les index non organisé

Les index

- Les index ordonnés (suite)
  - Syntaxes
    - Alter table nomtable
      add constraint nomcontrainte PRIMARY KEY
      [clustered|nonclustered] (liste des colonnes) ;

    - Create [unique] [clustered|nonclustered] index
      nomindex
      on nomtable (liste des colonnes)
      [ON nomgroupede fichier] ;

Les index

- Les index ordonnées
  - Organisent physiquement les données de la table
  - Créé par défaut lorsqu'une contrainte de clé primaire est définie sur une table
    - L'option NONCLUSTERED permet de ne pas créer l'index
  - Sont constitués d'un arbre dans lesquels les pages de niveau feuille contiennent les données de la table sous-jacente
  - Les niveaux supérieurs de l'arbre permettent d'ordonner les informations par rapport à la valeur indexée
  - Ainsi lors de l'ajout d'une ligne celle-ci est insérée en fonction de la valeur de sa clé
    - Il faudra baser l'index sur une valeur de clé stable (la clé primaire)

Index

- Fragmentation des index
  - Pour les opérations qui lisent les données, les index offrent les meilleures performances lorsque chaque page de l'index est aussi complète que possible.
  - Même si les index sont complets au démarrage (ou relativement complets), les modifications apportées aux données peuvent provoquer le fractionnement des pages d'index.
  - L'ajout d'un nouvelle d'index à la fin d'un index est simple ; cependant, ce processus est plus compliqué si l'entrée doit être effectuée en milieu de page d'un index complet existant

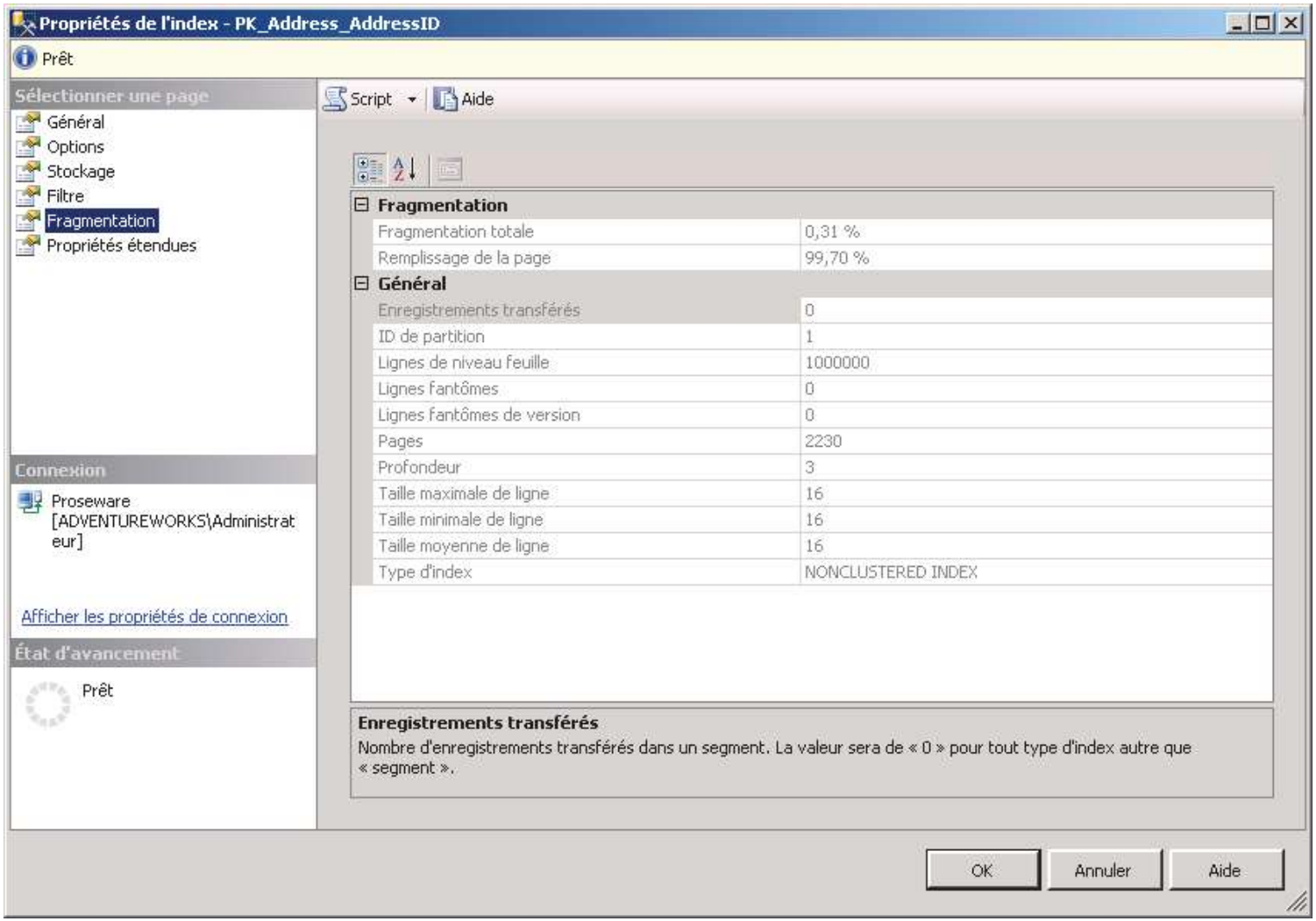
Index

- Fragmentation interne
  - Se produit lorsque les pages ne contiennent pas autant de données qu'elles le peuvent.
    - Cela se produit souvent lorsque la page est fractionnée pendant une opération d'insertion, et cela peut également se produire lorsqu'une opération de mise à jour entraîne le déplacement d'une ligne vers une autre page.
  - Dans l'un ou l'autre cas, il y a de l'espace laissé vide dans les pages.
- La fragmentation externe se produit lorsque des pages qui sont logiquement organisées ne sont pas conservées dans des numéros de page organisés.
  - Si une nouvelle page d'index doit être allouée, elle est logiquement insérée dans l'emplacement approprié dans la liste des pages, mais elle peut aussi bien être placée à la fin de l'index.
    - Cela implique qu'un processus qui doit lire les pages d'index dans l'ordre doit suivre les pointeurs pour localiser les pages ;
    - le processus implique alors d'accéder à des pages non séquentielles dans la base de données

Index

- Détection de la fragmentation
  - SQL Server fournit une mesure utile dans la colonne avg_fragmentation_in_percent de la vue de gestion dynamique sys.dm_db_index_physical_stats
  - SQL Server Management Studio fournit également des informations de fragmentation de l'index dans la page des propriétés pour chaque index, comme le montre la capture d'écran suivante de la base de données AdventureWorks

Index


Propriétés de l'index - PK_Address_AddressID
Prêt
Sélectionner une page
Général
Options
Stockage
Filtre
Fragmentation
Propriétés étendues
Script
Aide
Fragmentation
Fragmentation totale 0,31 %
Remplissage de la page 99,70 %
Général
Enregistrements transférés 0
ID de partition 1
Lignes de niveau feuille 1000000
Lignes fantômes 0
Lignes fantômes de version 0
Pages 2230
Profondeur 3
Taille maximale de ligne 16
Taille minimale de ligne 16
Taille moyenne de ligne 16
Type d'index NONCLUSTERED INDEX
Connexion
Proseware [ADVENTUREWORKS\Administrateur]
Afficher les propriétés de connexion
État d'avancement
Prêt
Enregistrements transférés
Nombre d'enregistrements transférés dans un segment. La valeur sera de « 0 » pour tout type d'index autre que « segment ».
OK
Annuler
Aide

Index

- L'espace libre disponible dans une page d'index peut influer grandement sur les performances des opérations de mise à jour des index.
  - Si un enregistrement d'index doit être inséré et qu'il n'y a pas d'espace libre, il est nécessaire de créer une nouvelle page d'index et de répartir le contenu de l'ancienne page sur les deux pages.
    - Les performances peuvent s'en trouver pénalisées si cette action se produit trop souvent
  - L'impact sur la performance des fractionnements de page peut être atténué en laissant de l'espace vide sur chaque page lors de la création d'un index, notamment un index cluster

Index

- Des espaces libres peuvent être laissés dans les index y compris les index cluster
- Cela s'effectue en spécifiant une valeur de FILLFACTOR. La valeur par défaut de FILLFACTOR est 0, ce qui signifie « remplit à 100 % ».
  - Toute autre valeur (y compris 100) correspond au pourcentage de remplissage souhaité pour chaque page
    - FILLFACTOR (niveau feuille uniquement)
    - PAD_INDEX (niveau intermédiaire et racine)
    - ```
      ALTER TABLE client ADD constraint PK_client primary
      key clustered
      (  contactID     ASC  )
       With (PAD_INDEX = OFF, FILLFACTOR=70);
      GO
      ```
  - Si l'on reprend l'exemple ci-dessus, cela signifie complet à 70 % et 30 % d'espace libre sur chaque page
  - FILLFACTOR s'applique uniquement aux pages de niveau feuille dans un index. PAD_INDEX est une option qui, lorsqu'elle est activée, entraîne l'allocation du même espace libre dans les niveaux non-feuille de l'index.
      ```

Maintenance des Index

- Comme les index sont mis à jour lors des modifications de données, ils peuvent être fragmentés au fil du temps.
- SQL Server fournit deux options pour éliminer la fragmentation des index cluster et non-cluster :
  - reconstruire ;
  - réorganiser

Maintenance des index

- Reconstruction des index
  - supprime l'index et le recrée
  - Cela supprime la fragmentation, libère de l'espace disque en compactant les pages d'après le paramètre spécifié ou existant du facteur de remplissage, et réorganise les lignes d'index en pages contiguës.
    - Si l'option ALL est spécifiée, tous les index de la table sont supprimés et reconstruits en une seule opération. Si une partie de l'opération échoue, l'opération entière est annulée.
- Étant donné que les reconstructions sont exécutées en une seule opération et sont consignées,
  - une seule opération de reconstruction peut utiliser une grande quantité d'espace dans le journal des transactions
- Il est possible d'effectuer l'opération de reconstruction en l'enregistrant de manière minimale lorsque la base de données est dans le mode de récupération simple ou BULK_LOGGED
  - Une opération de reconstruction enregistrée de façon minimale consomme moins d'espace dans le journal des transactions et s'effectue plus rapidement
  - Il doit y avoir de l'espace libre lors de la reconstruction des index

Maintenance des index

- La réorganisation d'un index utilise au minimum les ressources système. Elle défragmente le niveau feuille des index cluster et non-cluster des tables :
  - les pages de niveau feuille sont réorganisées physiquement afin de correspondre à l'ordre logique de gauche à droite des nœuds feuille
- La réorganisation entraîne aussi une compression des pages d'index.
  - La compression est basée sur la valeur du facteur de remplissage existante.
  - Il est possible d'interrompre une réorganisation sans perdre le travail effectué jusqu'alors
    - Par exemple, cela implique qu'il est possible de réaliser une réorganisation partielle chaque jour sur un index volumineux.
  - Pour les index très fragmentés > (30%) la reconstruction est généralement l'option la plus appropriée
  - Les plans de maintenance de SQL Server comportent des options de reconstruction ou de réorganisation des index. Si les plans de maintenance ne sont pas utilisés, il est important de créer une tâche qui réalise la défragmentation des index de vos bases de données

Maintenance des index

- Avec SQL Server Enterprise les index peuvent être créés, reconstruits et supprimés en ligne
  - Autorise l'accès simultané d'utilisateurs à la table sous-jacente et aux index
  - Nécessite uniquement des verrous partagés à court terme au début et à la fin de l'opération et des verrous de schéma pendant l'opération
  - Attention aux temps de réponse qui seront impactés
    - ```
      ALTER INDEX client_IDX ON Client REBUILD
      With (PAD_INDEX=OFF,
      FILLFACTOR=80,
      ONLINE = ON,
      MAXDOP = 4) ;
      ```
      ```

Maintenance des index

- Quand il réalise une opération en ligne de reconstruction d'index, SQL Server crée un index de mappage temporaire qui suit les modifications de données qui surviennent pendant l'opération de reconstruction
  - Pour des raisons de cohérence, SQL Server met un verrou partagé de très courte durée sur l'objet au début de l'opération puis à la fin de l'opération.
- Pendant l'opération de reconstruction en ligne, des verrous de schéma sont mis en place pour empêcher la modification des métadonnées
  - Cela signifie que les utilisateurs ne peuvent pas modifier la structure de la table au moyen de commandes telles que ALTER TABLE pendant l'opération en ligne de reconstruction d'index
- En raison du travail supplémentaire impliqué, les opérations en ligne de reconstruction d'index sont généralement plus lentes que leurs équivalents hors connexion
  - Attention : Certains index ne peuvent pas être reconstruits en ligne, notamment les index cluster avec des données d'objet volumineuses ou les index non cluster qui contiennent des données d'objet volumineuses

Maintenance des index

- Attention afin de conserver des temps de réponse correctes, il faudra réorganiser les index régulièrement !
  - Prévoir des jobs pour les index
    - Plan de maintenance dans SQL Server

Réorganisation des index

- script utiliser pour le rebuild des indexes ; pour gagner de l'espace disque !!!
  - USE ePO4_SRVW3KEPO
    GO

    Print 'Selecting Index Fragmentation in the database.'

    SELECT
    DB_NAME(DPS.DATABASE_ID) AS [DatabaseName]
    ,OBJECT_NAME(DPS.OBJECT_ID) AS TableName
    ,SI.NAME AS IndexName
    ,DPS.INDEX_TYPE_DESC AS IndexType
    ,DPS.AVG_FRAGMENTATION_IN_PERCENT AS AvgPageFragmentation
    ,DPS.PAGE_COUNT AS PageCounts
    FROM sys.dm_db_index_physical_stats (DB_ID(), NULL, NULL , NULL, NULL) DPS --N'LIMITED') DPS
    INNER JOIN sysindexes SI
    ON DPS.OBJECT_ID = SI.ID
    AND DPS.INDEX_ID = SI.INDID
    and DPS.AVG_FRAGMENTATION_IN_PERCENT >50
    ORDER BY DPS.avg_fragmentation_in_percent DESC
    GO

    ===> On peut faire manuellement : click droit sur la table , puis index

Réorganisation des index

- script utiliser pour le rebuild des indexes ; pour gagner de l'espace disque !!!
- (suite)

  – -- reorg
  Print 'Rebuilding indexes on every table in the database.'

  USE ePO4_SRVW3KEPO
  EXEC sp_MSforeachtable @command1="print 'Rebuilding indexes for ?' ALTER INDEX ALL ON ? REBUILD WITH (FILLFACTOR = 20)"
  GO

  --
  Print 'Reorganizing indexes on every table in the database.'

  EXEC sp_MSforeachtable @command1="print 'Reorganizing indexes for ?' ALTER INDEX ALL ON ? REORGANIZE"
  GO

    - on est dans la boucle sp_Msforeachtable, et ? est remplacé par le nom la table.

Statistiques

- Lorsque les données changent, les statistiques de distribution deviennent obsolètes
  - Les statistiques peuvent être mises à jour automatiquement ou à la demande
  - La mise à jour automatique est définie à partir d'une option de base de données et doit être activée
    - AUTO_UPDATE_STATISTICS (activée par défaut)
  - 2 autres possibilités pour la mise à jour des statistiques
    - UPDATE STATISTICS : instruction permettant de mettre à jour les statistiques
    - Sp_updatestats : procédure permettant la mise à jour des statistiques d'une base de donnée

Statistiques

- SQL Server prend les décisions sur les index à utiliser en fonction des statistiques qu'il conserve sur la distribution des données dans l'index
  - Les statistiques
- Les statistiques sont principalement mises à jour automatiquement par SQL Server, et AUTO_UPDATE_STATISTICS est activée par défaut dans toutes les bases de données
  - Il est recommandé de ne pas désactiver cette option
  - Pour les tables volumineuses, l'option AUTO_UPDATE_STATISTICS_ASYNC demande à SQL Server de mettre à jour les statistiques de manière asynchrone au lieu de retarder l'exécution de la requête, car sinon SQL Server mettrait à jour des statistiques obsolètes requises pour la compilation de la requête

Statistiques

- Les statistiques peuvent être mises à jour à la demande
  - L'exécution de la commande UPDATE STATISTICS sur une table entraîne la mise à jour de toutes les statistiques de la table.
  - La procédure système stockée sp_updatestats peut servir à mettre à jour toutes les statistiques de la base de données

L’outil - DBCC CHECKDB

- DBCCCHECKDB est un utilitaire qui est fourni avec SQL Server et qui offre un grand nombre de fonctionnalités de gestion
  - vérifier la cohérence des bases de données
  - En version 2012 il est connu sous le nom de l'utilitaire de commandes de console de base de données afin de mieux refléter la grande variété des tâches pour lesquelles il peut être utilisé
    - L'option CHECKDB de l'utilitaire DBCC effectue un contrôle particulièrement approfondi de la structure d'une base de données afin de détecter presque toutes les formes d'altération potentielle

L’outil - DBCC CHECKDB

- Les fonctions contenues dans la commande DBCC CHECKDB sont également disponibles en tant qu'options pouvant être exécutées indépendamment si nécessaire
- Les plus importantes sont :
  - DBCC CHECKALLOC = Vérifie la cohérence des structures d'allocation de l'espace disque pour une base de données spécifiée
  - DBCC CHECKTABLE = Vérifie les pages associées à une table spécifiée et les pointeurs entre les pages qui sont associées à la table. DBCC CHECKDB exécute DBCC CHECKTABLE pour chaque table de la base de données
  - DBCC CHECKCATALOG = Vérifie le catalogue de la base de données en effectuant des vérifications de cohérence logique dans les tables de métadonnées de la base de données. DBCC CHECKCATALOG ne vérifie pas les tables utilisateur.

L’outil - DBCC CHECKDB

- Bien que DBCC CHECKDB propose des options de réparation, il n'est pas toujours possible de réparer une base de données sans perte de données
  - l'exécution de DBCC CHECKDB doit être synchronisée avec votre stratégie de rétention de sauvegarde !
- C’est un outil gourmand en espace disque (temp et autre, ainsi qu’en ressource d'E/S et de processeur et peut prendre beaucoup de temps pour s'exécuter)
  - A exécuter en dehors du travail des utilisateurs !
  - Pour fournir une estimation de la quantité d'espace nécessaire dans tempdb, DBCC CHECKDB propose l'option ESTIMATEONLY
- Il est recommandé d'exécuter DBCC CHECKDB sur une base de données avant d'en réaliser la sauvegarde.
  - Ce contrôle permet de garantir que la sauvegarde contient une version cohérente de la base de do

L'outil – DBCC CHECKDB

- Les options de DBCC CHECKDB
  - PHYSICAL_ONLY : souvent utilisée sur les systèmes de production car elle réduit considérablement le temps nécessaire pour exécuter DBCC CHECKDB sur des bases de données volumineuses
    - Il faudra quand même exécuter la version complète
  - NOINDEX : indique qu'il ne faut pas exécuter de contrôle intensif des index non cluster pour les tables utilisateur
    - réduit également la durée totale d'exécution
  - EXTENDED_LOGICAL_CHECKS : (sur des version supérieure à 2008) effectuer des contrôles détaillés de la structure interne des objets tels que les types de données CLR (Common Language Runtime) définis par l'utilisateur et les types de données spatiales
  - TABLOCK : utilisée pour faire en sorte que DBCC CHECKDB mette un verrou de table sur chacune lors des vérifications de cohérence, au lieu d'utiliser les captures instantanées internes de la base de données. Cela réduit l'espace disque nécessaire, avec comme contrepartie le fait d'empêcher d'autres utilisateurs de mettre les tables à jour.

L'outil – DBCC CHECKDB

- Remarques :
  - La base de données doit être en SINGLE_USER
  - Le résultat indique l'option minimum pour la récupération
    - REPAIR_REBUILD pour les réparations qui peuvent être effectuées sans perte de données
    - REPAIR_ALLOW_DATA implique la perte de données
      - Dans ce cas pensez à restaurer la base de données au lieu de perdre des données avec DBCC CHECKDB