



Formation SQL Server  
Optimisations et amélioration des performances  
Versions 2016 à 2008

Cédric Del Nibbio

1

2

Formation SQL Server - Optimisations et amélioration des performances  
Versions 2016 à 2008

## INTRODUCTION



3

## Présentation

- Nom
- Affiliation à une société
- Titre / Fonction
- Responsabilité professionnelle
- Expérience en administration de bases de données
- Expérience en développement SQL Server
- Attentes vis-à-vis du cours

4

## Pré-requis

- Expérience d'administration SQL Server
- Connaissance empirique du Transact-SQL
- Connaissances basiques de Microsoft Windows

5

## Objectifs

- Décrire l'architecture SQL Server et ses composants
- Décrire les modèles d'exécutions, le système d'attente des requêtes
- Décrire les concepts principaux liés aux I/O
- Décrire les concepts et « best practices » liés à la gestion des fichiers de données pour les bases de données utilisateur et TempDB
- Décrire les concepts liés à la concurrence, les transactions et les niveaux d'isolation
- Décrire les concepts de l'optimiseur et comment identifier et résoudre les problèmes liés au plan d'exécution
- Décrire et comprendre les concepts liés au Plan Cache afin d'anticiper et de résoudre les problèmes
- Décrire les stratégies d'analyse et les scénarios afin d'utiliser au mieux les événements étendus
- Comprendre les techniques utilisées afin d'identifier et résoudre les goulots d'étranglements pour améliorer les performances globales

6

## Plan du cours

- **Module 1**

Architecture Client/serveur, Composants SQL Server, Planification, attentes

- **Module 2**

SQL Server I/O

- **Module 3**

Structures de base de données

- **Module 4**

Gestion de la mémoire dans SQL Server

- **Module 5**

Concurrence et transactions

7

## Plan du cours (suite)

- **Module 6**

Statistiques et index

- **Module 7**

Exécution des requêtes et analyse du plan d'exécution

- **Module 8**

Cache des plans d'exécution et recompilation

- **Module 9**

Événements étendus (Extended Events)

- **Module 10**


Monitoring et analyse de traces


- **Module 11**


Conception Objets SQL


8


# Logistique


Heure de cours


Horaires d'ouverture des locaux


Téléphones


Stationnement

Messages

Toilettes

Zones fumeurs


Repas

Recyclage

9

Architecture client/serveur, composants SQL Server, Planification, attentes

# MODULE 1



EXPACEO  
L'ESPACE DES EXPERTS

10

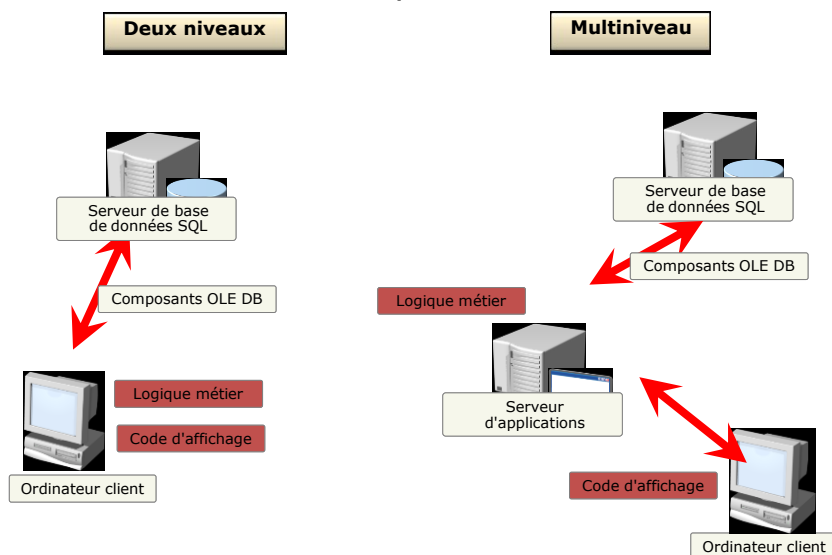
5

## Aperçu du Module

- Vue d'ensemble de l'architecture client/serveur
- Protocoles de connexion
- Moteur de base de données
- SQLOS
- Cycle de de vie d'une requête
- Monitorer le comportement du moteur
- Modèle d'exécution SQL Server
- Cycle de vie d'une requête
- Attentes et files d'attente

11

## Vue d'ensemble de l'architecture client/serveur



12

## Vue d'ensemble de l'architecture client/serveur

Composants serveur	Description
• Le moteur de base de données SQL Server	Service principal pour le stockage et le traitement des données
• Analysis Services	Outils de création et de gestion du traitement analytique
• Reporting Services	Composants de création et de déploiement de rapports
• Integration Services	Outils de déplacement, de copie et de transformation de données

**Le moteur de base de données comprend aussi ces composants :**

• La recherche en texte intégral

• La réplication

• Service Broker

• Notification Services

13

## Vue d'ensemble de l'architecture client/serveur

Composants	Description
• Protocoles	Manières d'implémenter l'interface externe vers le serveur SQL Server
• Moteur relationnel	Interface vers le moteur de stockage, composée de services qui permettent d'interagir avec les composants et fonctionnalités de stockage de la base de données sous-jacente
• Moteur de stockage	Au cœur de SQL Server, service hautement évolutif et disponible pour le stockage, le traitement et la sécurité des données
• SQLOS	Système d'exploitation disposant d'une puissante API, qui réunit tous les composants système pour permettre les innovations de SQL Server en matière d'évolutivité et de performances et pour fournir des fonctionnalités qui facilitent l'administration et la prise en charge






14

# Vue d'ensemble de l'architecture client/serveur

Objets	Commentaires
• Tables	Contiennent les données des bases de données SQL Server
• Vues	Agissent comme des tables virtuelles ou des requêtes stockées
• Index	Permettent une récupération rapide, créés à partir d'une ou de plusieurs colonnes d'une table ou d'une vue
• Déclencheurs	Exécutent un code SQL par lot lorsqu'une commande INSERT, UPDATE ou DELETE est exécutée sur une table spécifique
• Procédures	Acceptent des paramètres, contiennent des instructions et retournent des valeurs
• Contraintes	Empêchent le placement dans une colonne de données incohérentes
• Règles	Spécifient les valeurs acceptables pouvant être insérées dans une colonne

15

# Vue d'ensemble de l'architecture client/serveur

Outils d'administration	Description
 SQL Server Management Studio	Environnement intégré qui permet d'accéder à des composants serveur SQL Server, et de les configurer, les gérer et les administrer
 Gestionnaire de configuration SQL Server	Interface qui permet de gérer les services SQL Server, les protocoles et les alias clients
 SQL Server Profiler	Outil GUI permettant de profiler et de suivre le moteur de base de données et Analysis Services
 Assistant Paramétrage du moteur de base de données	Application qui permet de créer des jeux d'index, de vues indexées et de partitions optimaux
 Business Intelligence Development Studio	IDE qui permet de créer des solutions Analysis Services, Reporting Services et Integration Services

16

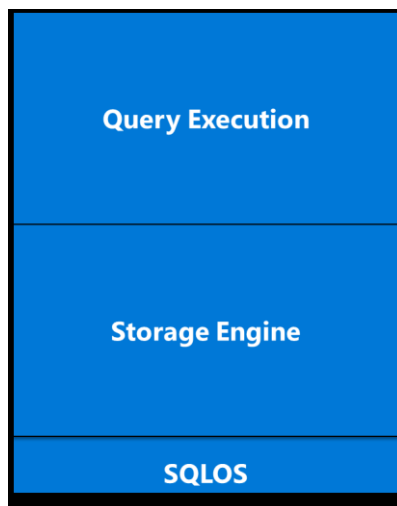


## Protocoles de connexion

- 3 protocoles de connexion supportés:
  - Shared Memory
  - Named Pipes
  - TCP/IP
- La communication dans les couches applicatives se fait grâce au protocole Tabular Data Stream (TDS)

17

## Moteur de base de données



- Couche d'exécution des requêtes
  - Analyse et optimise les requêtes, gère le cache et l'exécution des plans d'exécutions
- Moteur de stockage
  - Gère les pages du buffer, les I/O des fichiers physiques, verrous et les transactions
- Couche SQLOS
  - Donne un niveau d'abstraction des fonctions de l'OS, des tâches et de la gestion de la mémoire

18

## SQLOS

- Les besoins de SQL Server pour les ressources bas-niveaux sont complexes
- Beaucoup de services dans le moteur ont besoin d'avoir accès à ces ressources bas-niveaux
- Les composants pour donner cet accès sont groupés dans une seule fonctionnalité appelée SQLOS
- Les composants SQL Server font des appels au SQLOS
- SQLOS fournit beaucoup de fonctionnalité, pour gérer la mémoire, la planification, gérer les I/O, etc...

19

## Cycle de vie d'une requête

- Interfaces réseaux SQL Server
  - Reçoit la requête du client
- Moteur relationnel
  - Décompose la requête et la lie aux objets de la base de données
  - Compile le plan de requête
  - Exécute le plan de requête
  - Génère les résultats
- Moteur de stockage
  - Utilisé par le moteur relationnel pour accéder aux données et aux métadonnées
- SQLOS
  - Fournit des fonctions bas-niveaux pour les moteurs de stockage et relationnel

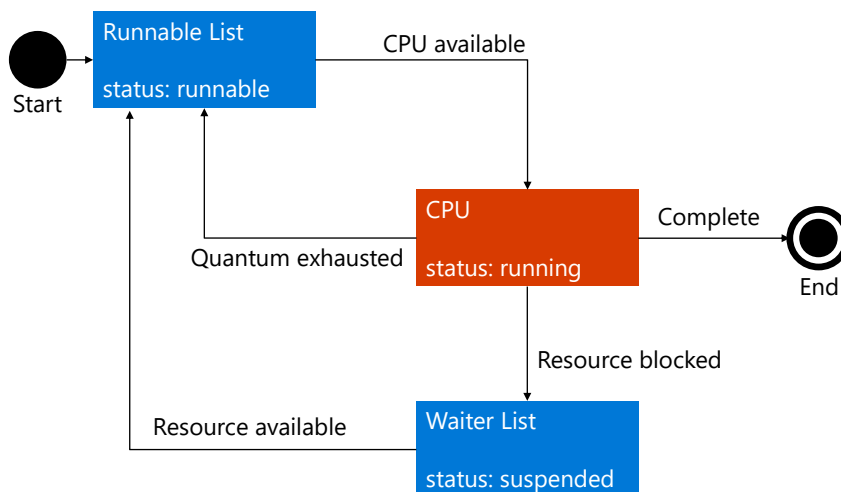
20

## Monitorer le comportement du moteur

- Windows System Monitor
- Activity Monitor
- Performance Monitor
- DMVs et DMFs
  - dm\_exec\_\* (sys.dm\_exec\_sessions)
  - dm\_os\_\* (sys.dm\_os\_schedulers)
  - dm\_tran\_\* (sys.dm\_tran\_locks)
  - dm\_io\_\* (sys.dm\_io\_virtual\_file\_stats)
  - dm\_db\_\* (sys.dm\_db\_index\_physical\_stats)

21

## Modèle d'exécution SQL Server



22

## Cycle de vie d'une requête utilisateur

- Connexion établie: sys.dm\_exec\_connections
- Session ID assignée: sys.dm\_exec\_sessions
- Requête créée: sys.dm\_exec\_requests
- Tâche créée: sys.dm\_os\_tasks
- Tâche assignée: sys.dm\_os\_workers
- Tâche s'exécute sur un thread CPU: sys.dm\_os\_threads
- Vue des tâches planifiées: sys.dm\_os\_schedulers

23

## Attentes et files d'attentes

- Une requête ou une tâche est en attente si une ressource requise n'est pas disponible
- SQL Server suit les ressources en attente. On distingue 2 types principaux:
  - Attentes de ressources (Suspended)
  - Attente de CPU (Runnable)
- Fils d'attentes
  - Ressources système et utilisation, mesurées grâce au Performance Monitor, DMV et d'autres outils...

24

## Attentes et files d'attentes

- Voir les statistiques d'attentes
  - Les statistiques d'attentes sont agrégées dans la DMV **sys.dm\_os\_wait\_stats**
  - La DMV **sys.dm\_os\_waiting\_tasks** montre les statistiques d'attentes en cours
  - Les statistiques pour les sessions actives sont agrégées dans la DMV **sys.dm\_exec\_session\_wait\_stats** (SQL Server 2016 et supérieur)

25

## Attentes et files d'attentes

- Les types d'attentes LCK\_M\_\*
  - Une tâche est en attente sur un verrou incompatible
  - Les causes
    - Une mise à jour importante ou un balayage de table provoquant un verrou en cascade
    - Un verrou partagé sur une donnée en accès
  - Les solutions possibles
    - Utiliser **sys.dm\_os\_waiting\_tasks** pour trouver le principal type d'attente bloquant
    - Considérer d'utiliser un niveau d'isolation différent
    - Faire une passe sur les index et optimiser les requêtes

26

## Attentes et files d'attentes

- Les types d'attentes PAGELATCH\_\*
  - Une tâche est en attente de verrou sur une page dans le tampon du buffer
  - Les causes
    - Une contention sur les pages les plus utilisées
    - Une contention sur l'allocation des pages de fichiers
  - Les solutions possibles
    - Utiliser une nouvelle stratégie d'indexation ou de partitionnement
    - Utiliser plus d'objets temporaires ou ajouter des fichiers de données

27

## Attentes et files d'attentes

- Les types d'attentes PAGEIOLATCH\_\*
  - Une tâche est en attente de recuperation d'une page de donnée du disque vers la mémoire
  - Les causes
    - Une contention I/O
    - Une mauvaise performance de la requête
  - Les solutions possibles
    - Revoir la configuration I/O
    - Revoir et mettre à jour les statistiques des indexes
    - Identifier les requêtes comportant des balayages parallèles et des conversion implicites dans le plan de requête
    - Réduire la division des pages

28

## Attentes et files d'attentes

- Les types d'attentes CXPACKET
  - Une tâche participante dans une requêtes parallélisée se complete avant les autres tâches
  - Les causes
    - Un mauvais plan de requête utilisé
    - Une mauvaise repartition du travail entre les taches
    - Des taches parallèles ralenties par d'autres types d'attentes
  - Les solutions possibles
    - Mettre à jour les statistiques
    - Mettre à jour la valeur configurée du MAXDOP
    - Augmenter le seuil de coût du parallélisme

29

## Attentes et files d'attentes

- Les types d'attentes WRITELOG
  - Une tâche attend qu'une transaction de log soit écrite sur le disque
  - Les causes
    - Disque de log mal dimensionné
    - Beaucoup de petites transactions
    - Des index inutiles
    - Division de pages fréquente
  - Les solutions possibles
    - Déplacer les fichiers de logs sur un disque plus performant
    - Utiliser moins de transactions
    - Revoir les index

30

## Attentes et files d'attentes

- D'autres types d'attentes communément rencontrés
  - PREEMPTIVE\_OS\_\*
  - BACKUPTHREAD
  - SOS\_SCHEDULER\_YIELD
  - THREADPOOL
  - ASYNC\_NETWORK\_IO
  - RESOURCE\_SEMAPHORE
  - LOGBUFFER
  - ASYNC\_IO\_COMPLETION
  - IO\_COMPLETION
  - CMEMTHREAD

31

SQL Server I/O

## MODULE 2



32



## Aperçu du Module

- Concepts principaux des I/O
- Solutions de stockage
- Compression des données

33

## Concepts principaux des I/O

- Opérations d'entrées/sorties par seconde
  - Mesure du nombre d'opérations qu'un système de stockage peut supporter en une seconde
  - Peut être ensuite divisé entre lecture/écriture et aléatoire/séquentiels IOPS:
    - Ecriture aléatoire
    - Lecture aléatoire
    - Ecriture séquentielle
    - Lecture séquentielle
  - Les différents fichiers de bases de données SQL Server ont des caractéristiques lecture/écriture différentes

34

## Concepts principaux des I/O

- Débit
  - Mesure de quantité de data qu'un système peut transférer en un temps fixe
- Facteur de latence
  - Mesure de complétion d'une requête de lecture ou d'écriture de donnée sur un support de stockage
  - Microsoft recommande que cet indicateur soit le plus bas possible pour le journal de transactions

35

## Solutions de stockage

- Stockage directement attaché
- Storage Area Network
- Windows Storage Spaces
- SQL Server Data Files dans Microsoft Azure

36

## Solutions de stockage

- Stockage directement attaché
  - Un ou plusieurs support de stockage est directement attaché au serveur
    - Dédié à un serveur
    - Avantages
      - Facilement et rapidement approvisionné
      - Facile à monitorer et à dépanner
    - Désavantages
      - Expansions limitées par la partie hardware
      - Manque de flexibilité, l'espace non utilisé ne peut pas être partagé sur d'autres serveurs

37

## Solutions de stockage

- Storage Area Network (SAN)
  - Ensemble de disques liés par un réseau
    - Chaque serveur se connecte au réseau et peut partager des disques, caches et débit avec d'autres serveurs
    - Avantages
      - Augmente l'utilisation des disques et réduit la gérance
      - Beaucoup d'options de sécurité et de protection des données seulement possibles dans le cas de SAN
    - Désavantages
      - Performances non prévisibles, bande passante limitée, coût élevé

38

## Solutions de stockage

- Windows Storage Spaces
  - RAID géré par Microsoft
    - Les disques physiques sont groupés dans des stockages pool
    - Storage Spaces permet de créer des volumes à partir de la pool de stockage

39

## Solutions de stockage

- Fichiers de données SQL Server dans un Blob Storage Azure
  - Utilisables par des instances on-premises ou sur une machine virtuelle Azure
    - Avantages
      - Stockage illimité
      - Migration facilité
      - Stockage centralisé
      - Sauvegardes dans Azure par snapshot
    - Désavantages
      - Fichiers de maximum 1 To
      - Toutes les fonctionnalités SQL Server ne sont pas disponibles
      - Les coûts sont difficilement prévisibles

40

## Compression des données

- Pourquoi compresser les données
  - Sauvegarde de l'espace disque
  - Améliore les performances sur un système très sollicité au niveau des I/O, les requêtes accédant aux données compressées retournant moins de pages
  - Peut être ajouté aux objets suivants
    - Tables Heap
    - Tables stockées comme des index clustered
    - Index nonclustered
    - Vues indexées
    - Partitions individuelles
    - Index spatiaux

41

## Compression des données

- Compression de page
  - Ajoute une structure d'information de compression (CI) en dessous de l'en-tête de chaque page compressé
  - Comprime les données de trois façons
    - Compression de ligne: compression de page incorpore une compression de page
    - Compression de préfixe: remplace les valeurs de préfixe commune avec un identifiant plus petit, stockée dans la structure CI
    - Compression de dictionnaire: similaire au préfixe, mais remplace les valeurs répétées avec des identifiants plus petits, stockées aussi dans la structure CI

42

## Compression des données

- Compression de ligne
  - Sauvegarde de l'espace en stockant des types de données de longueur fixe en longueur variable, comme les types entiers
  - Ne fonctionne pas sur les types XML, images, text et ntext
  - Ajoute quatre bits supplémentaires à chaque colonne compressée pour stocker la longueur du type de donnée
    - Pour les valeurs NULL, les quatre bits est le seule espace consommé

43

## Compression des données

- Compression Unicode
  - La compression Unicode utilise l'algorithme Standard Compression Scheme for Unicode (SCSU)
  - Les types de données nchar(n) et nvarchar(n) sont automatiquement compressés avec SCSU
  - Nvarchar(MAX) peut être compressé en compression page mais pas ligne
  - Le ratio de compression varie entre les différents langage: l'anglais a 50 % de rendement, le japonais 15 %

44

## Compression des données

- Considérations sur la compression
  - Utilisez la compression pour sauvegarder de l'espace disque
    - Considérez les caractéristiques de la donnée pour décider si une table ou un index est un bon candidat pour de la compression
    - Considérez de compresser les tables qui ont beaucoup de colonnes avec des types de données fixes ou avec beaucoup de données redondantes
  - Utilisez la compression pour améliorer les performances
    - La compression peut réduire les I/O disque et augmenter le montant de donnée que SQL Server peut garder en mémoire
    - La compression augmente l'utilisation CPU
    - Il faut mettre en balance les bénéfices de performances contre l'utilisation augmentée du CPU

45

Structures de base de données

## MODULE 3



46

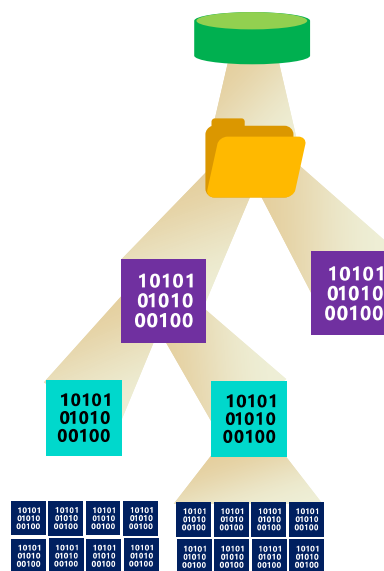
## Aperçu du Module

- Composants des bases de données
- Fichiers de données
- TempDB

47

## Composants des bases de données

- Base de données
- Groupe de fichiers
- Fichiers de données
- Etendues
- Pages (8 KB chacune)



48



## Composants des bases de données

- Groupe de fichiers et fichiers de bases de données
  - Groupe de fichier
    - Groupage logique des fichiers de données
    - Amène des bénéfices en terme d'administration et de performance
    - FILESTREAM ont leur proper groupe de fichier
  - Fichiers de base de données
    - Fichiers de données
      - Au minima 1 par base de données
      - Plusieurs fichiers peuvent aider aux performances et aider à la maintainibilité
    - Journal de transaction
      - De manière Générale, un par base de données, pas d'avantages en performance dans le cas ou il y a plusieurs fichiers

49

## Composants des bases de données

- Etendues
  - Comprend 9 pages physiques continues
    - Mélangées: partagées entre différents objets de la base de données
    - Uniforme: détenue par un seul objet de la base données

50

# Composants des bases de données

- Structure des pages et type de pages
  - Pages
    - 8 KB en taille
    - En-tête de 96 byte

Types de pages
Data (1)
Index (2)
Text (3 and 4)
Boot (13)
File Header (15)
PFS (11)
GAM (8)
SGAM (9)
IAM (10)
DIFF_MAP (16)
ML_MAP (17)

51

# Composants des bases de données

- Types d'enregistrements
  - Data records
  - Forwarding/forwarded records
  - Index records
  - Leaf level index records
  - Non-leaf level index records
  - Text records
  - Versioned records
  - Ghost records

52

## Composants des bases de données

- Bitmaps d'allocation et pages spéciales
  - Global Allocation Map (GAM)
  - Shared Global Allocation Map (SGAM)
  - Differential Change Map (DCM)
  - Bulk Change Map
  - Page Free Space (PFS)
  - Index Allocation Map (IAM)

53

## Composants des bases de données

- Pages d'allocation et unités d'allocation
  - Page d'allocation
    - Le processus d'alloué des Nouvelles pages à un objet
  - Unités d'allocation
    - IN\_ROW\_DATA
    - LOB\_DATA
    - ROW\_OVERFLOW\_DATA
  - Les pages d'allocations peuvent être analysées en utilisant les pages DBCC

54

## Fichiers de données

- Best practices pour la configuration des volumes
  - RAID 10 pour les journaux de transactions
  - RAID 10 ou RAID 5 pour les fichiers des bases de données
  - Stocker les données et les journaux de transactions sur deux volumes séparés physiquement
  - Défragmenter les disques fréquemment
  - Dimensionner correctement les fichiers de données et les journaux de transaction pour éviter les croissances auto non contrôlées

55

## Fichiers de données

- Nombre de fichiers de données et emplacement
  - Pour la plupart des bases de données un seul fichier de données par base de données est suffisant
    - Stocker les fichiers de données et les journaux de transaction sur un disque séparé physiquement
  - Pour améliorer les performances
    - Placer d'autres fichiers de données dans un groupe de fichier séparé du groupe de fichier primaire et marqué le nouveau groupe de fichier comme DEFAULT
    - Séparer plusieurs fichiers de données entre différents disques physiques
    - Placer les tables les plus utilisées et leurs index nonclustered sur un différent disque physique

56

## Fichiers de données

- I/O physiques vs I/O logiques
  - I/O logiques – Pages lues à partir du disque
  - I/O physiques – Pages lues à partir du cache
  - Buffer cache hit ratio – pourcentage des pages lues du cache qui n'ont pas eu besoin d'être lues à partir du disque

57

## Fichiers de données

- Allocation au sein des fichiers de données
  - L'allocation est simple dans un groupe de fichier qui à un seul fichier
  - Dans un groupe à plusieurs fichiers, SQL Server utilise:
    - Round robin allocation (allocation à tour de rôle)
    - Proportional fill (remplissage proportionnelle)

58

## Fichiers de données

- Instant File Initialization (IFI)
  - Améliore les performances en sautant la mise à zéro des pages de données pendant la création et la croissance
  - Désactivé par défaut: risque à la sécurité
  - Il est recommandé de l'activer

59

## Fichiers de données

- Auto croissance et auto rétrécissement
  - Auto croissance
    - Etend un fichier automatiquement quand l'espace disponible devient faible
    - Gérer manuellement la taille des fichiers pour de meilleures performances
    - Best practices pour l'auto croissance
      - Laisser activé pour éviter les indisponibilités dans le cas où le fichier arrive à sa taille limite
      - Configuré à une taille fixe plutôt qu'à un pourcentage
  - Auto rétrécissement
    - Rétréci un fichier de données automatiquement pour libérer de l'espace disque
    - Utiliser avec précaution
      - L'auto rétrécissement fragmente les index
      - Il n'est pas contrôlé et affecte les performances
      - Il peut amener à des cycles croissance/rétrécissement affectant grandement les performances

60

## Fichiers de données

- Rétrécissement des fichiers de données
  - Rétrécissement
    - Cause une fragmentation des index
    - Intensif en ressource
    - Dégrade les performances
  - Utilisé pour les cas suivant
    - Vider un fichier avant de l'enlever
    - Changer un fichier ou un groupe de fichier en lecture seule
    - Récupérer de l'espace disque après une opération de suppression importante
  - Alternatives
    - Déplacer tout les index dans un nouveau groupe de fichier
    - Déplacer des segments avec le rétrécissement. Les segments ne sont pas fragmentés après rétrécissement
    - Supprimer un ancien groupe de fichier

61

## Fichiers de données

- Monitorer les fichiers de bases de données
  - Configuration des fichiers de base de données
    - Propriété de la base de données dans SSMS
    - sp\_helpfile et sp\_helpfilegroup
    - sys.database\_files et sys.filegroups
    - sys.master\_files
    - sys.dm\_db\_file\_space\_usage
  - Activité des fichiers de base de données
    - SSMS Moniteur d'activité– Data File I/O
    - Wait statistics – PAGEIOLATCH\_\*, WRITELOG
    - sys.dm\_io\_virtual\_file\_stats

62

## TempDB

- Utilisation de TempDB
  - Stocke
    - Les objets internes
    - Version store
    - Certains objets utilisateur
  - Recrée au démarrage
  - Doit toujours avoir de l'espace libre
  - Doit être activement monitorer

63

## TempDB

- Version Store
  - Stocke toute les versions des enregistrements de toutes les bases de données
  - Unité d'allocation créée toute les minutes
  - Nettoyage toute les minutes
  - Opérations non logués

64



## TempDB

- Configuration TempDB
  - Configuration
    - Activer IFI
    - Dimensionner par rapport à l'usage
    - Isoler les fichiers de données de TempDB des autres
    - Utiliser plusieurs fichiers de données
    - Activer l'auto croissance
  - Il est interdit de
    - Faire un Backup / restore
    - Ajoute des groupes de fichiers
    - Changer la collation
    - Mettre hors-ligne ou en lecture seule

65

Gestion de la mémoire dans SQL Server

## MODULE 4



66

## Aperçu du Module

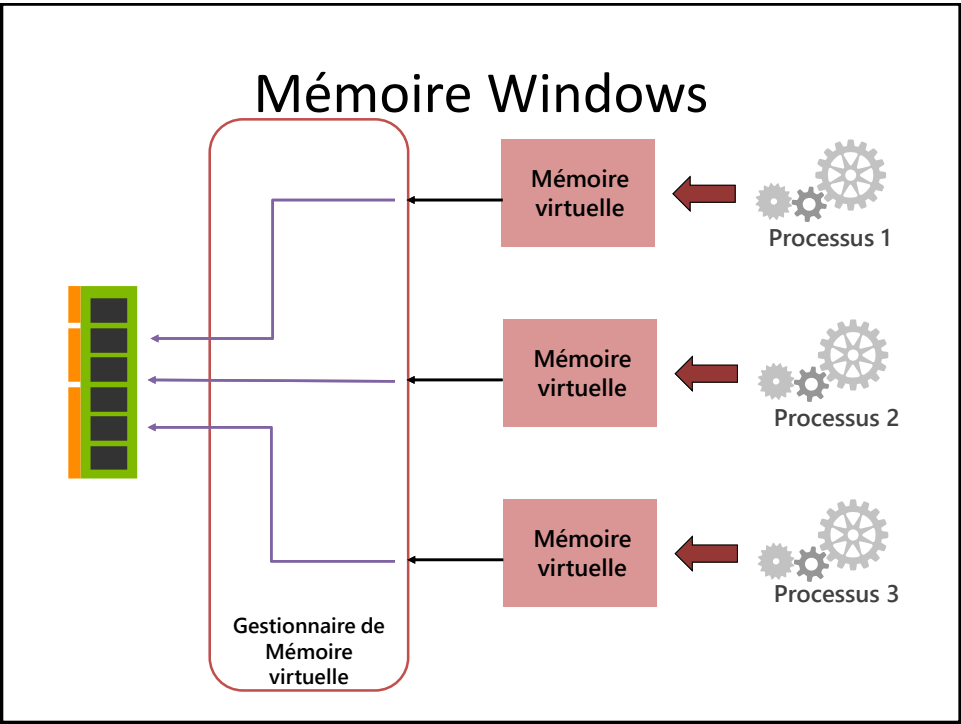
- Mémoire Windows
- Mémoire SQL Server
- In-Memory OLTP

67

## Mémoire Windows

- Virtual Address Space (VAS)
  - VAS fournit une couche d'abstraction entre une application et la mémoire physique
  - L'OS peut choisir la meilleure manière d'utiliser la mémoire physique entre les processus

68



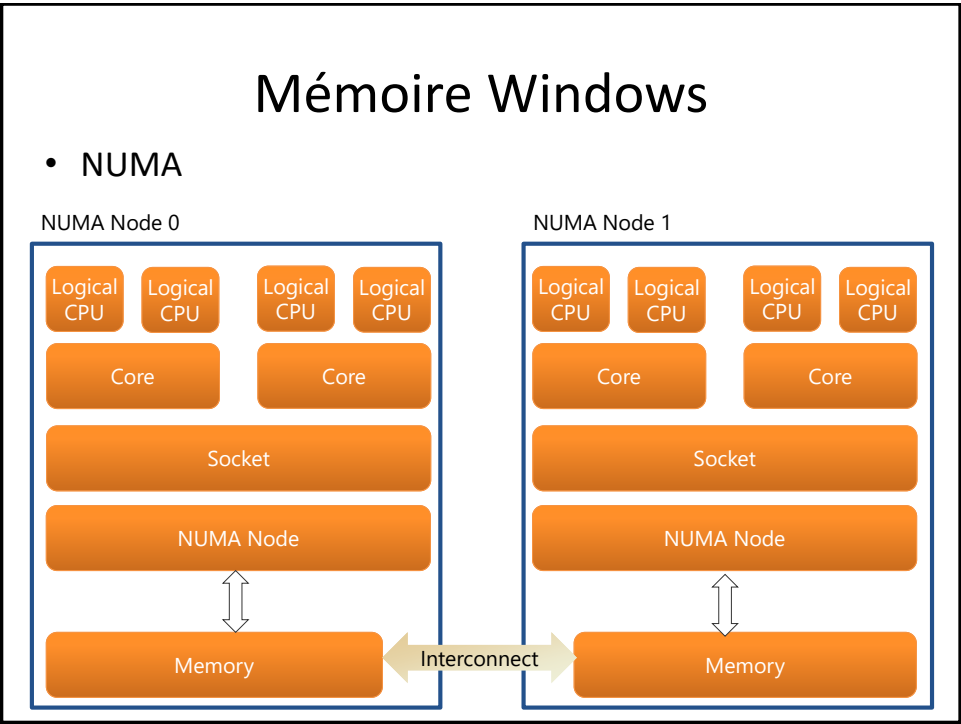
69

### Mémoire Windows

- 32-bit vs. 64-bit

32-bit	64-bit
$2^{32} = 4,294,967,295 = 4 \text{ GB}$	$2^{64} = 18,446,744,073,709,551,616 = 16 \text{ exabytes}$
Limited to using only 4 GB due to the architecture	Limited to using 256 TB by processor address space
<div>2 GB Kernel mode</div> <div>2 GB User mode</div>	<div>8 TB Kernel mode</div> <div>8 TB User mode</div>

70



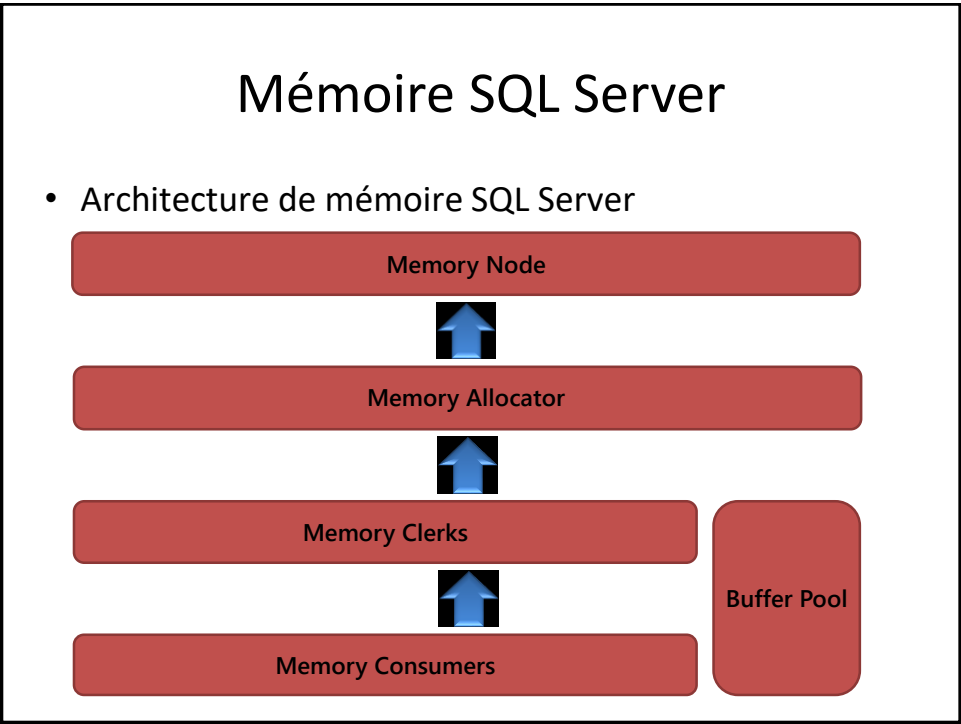
71

# Mémoire SQL Server

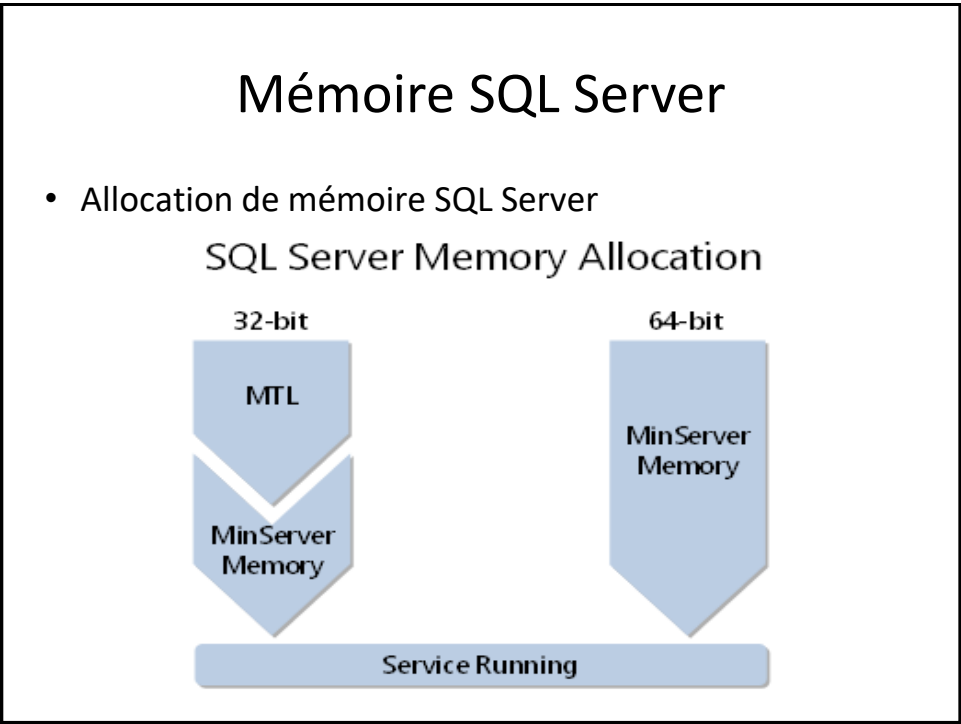
- Modèles de mémoire SQL Server

	Conventional	Locked	Large
Page Size	4/8 KB	4/8 KB	> = 2 MB
Static/Dynamic	Dynamic	Dynamic	Static. Memory committed at startup
Lock Pages in memory (LIPM)	Not Required	Required for Service account	Required for service account
Allocation API	VirtualAlloc	AWE APIs	VirtualAlloc with MEM_LARGE_PAGES option
Prerequisites	None	Enterprise Edition 64-bit	Enterprise Edition, Mem >= 8 GB, 64-bit, Trace Flag 834

72



73



74

## Mémoire SQL Server

- SQL Server NUMA
  - NUMA matériel
    - Détecté et utilisé automatiquement
  - NUMA logiciel
    - Configuré manuellement
    - Demande des changements dans le registre

75

## Mémoire SQL Server

- Configuration mémoire SQL Server
  - Minimum de mémoire du serveur
    - Niveau de mémoire qui, une fois atteint, ne sera pas relâché
  - Maximum de mémoire du serveur
    - Montant maximum de mémoire que SQL Server demandera
  - Recommandation
    - Si possible, laisser SQL Server gérer la mémoire dynamiquement

76

## In-Memory OLTP

- Définition
  - Tables In-memory
    - Tables transactionnelles stockées en mémoire
    - Accède de la même façon aux données qu'une table basée sur le disque
    - Complètement transactionnelle
    - Durable
    - Très faible latence

77

## In-Memory OLTP

- Tables Memory-Optimized
  - Tables Memory\_optimized durables
    - Conforme à la propriété ACID
    - Persistées sur le disque
  - Tables Mémoire\_Optimized non durables
    - Non conforme à la propriété ACID
    - Non persistante sur le disque (crash serveur= perte de données)
    - Pas d'I/O disque

78

## In-Memory OLTP

- Procédures stockées nativement compilées
  - Compilées à la création
  - Performances améliorées
  - Supporte les blocs atomiques et les paramètres de contraintes NOT NULL

79

Concurrence et transactions

## MODULE 5



80



## Aperçu du Module

- Concurrency et transactions
- Verrous

81

## Concurrency et transactions

- Modèles de concurrence
  - Concurrences pessimistes
    - Intégrité des données maintenue en utilisant des verrous
    - Seulement un utilisateur à la fois peut accéder aux données
  - Concurrences optimistes
    - Les données sont marquées pour changement avant une mise à jour
    - Minimum de verrous

82

## Concurrence et transactions

- Problèmes de concurrences
  - Dirty reads
    - Des données non commitées sont inclus dans le résultat
  - Lost Update
    - Deux mises à jour concurrents, le premier est perdu
  - Non-repeatable read
    - Les données changent entre deux SELECT identiques dans une transaction
  - Phantom read
    - La donnée est lue et supprimée avant la lecture complète
  - Double read
    - Une data dans un ensemble est lue deux fois quand la valeur de la clé de cet ensemble change

83

## Concurrence et transactions

- Niveaux d'isolations des transactions
  - Niveaux d'isolations pessimistes:
    - READ UNCOMMITTED
    - READ COMMITTED
      - READ\_COMMITTED\_SNAPSHOT OFF
    - REPEATABLE READ
    - SERIALIZABLE
  - Niveaux d'isolations optimistes (versioning de ligne):
    - READ COMMITTED
      - READ\_COMMITTED\_SNAPSHOT ON
    - SNAPSHOT

84

## Concurrence et transactions

- Travailler avec les niveaux d'isolations utilisant le versioning de ligne
  - Bénéfices du versioning de ligne
    - Moins de verrous
    - Moins de blocages
  - Problèmes avec le versioning de ligne
    - Le versioning est couteux et ajoute de la charge à TempDB
    - Les applications doivent pouvoir gérer les conflits de mise à jour
  - Autres considerations
    - Il y a toujours des risques de verrous
    - Les écritures bloquent toujours les autres écritures
    - Il faut une seule connexion active pour configurer `READ_COMMITTED_SNAPSHOT ON`

85

## Concurrence et transactions

- Transactions
  - C'est une unité de travail, faite d'une ou plusieurs expression `TRANSACT-SQL`
    - Atomicity
    - Consistency
    - Isolation
    - Durability
  - Modes de transactions
    - Auto-commit
    - transactions explicites
    - Transactions implicites
    - Transactions Batch-scoped

86

## Concurrence et transactions

- Travailler avec les transactions
  - Nommer les transaction
    - Pas d'effets sur le code
  - Transactions imbriquées
    - Seule le statut de la transaction externe aura de l'effet
    - Utiliser @@TRANCOUNT pour tracer les transactions imbriquées
  - Les transactions se finissent dans les cas suivant
    - Erreur de ressource
    - XACT\_ABORT configure
    - Fermeture de connexion
  - Il est recommandé de garder les transactions les plus courtes possibles

87

## Verrous

- Architecture des verrous
  - Les verrous sont conçus comme un équilibre entre la consistance et la concurrence
    - Verrous – Niveau logique
    - Latches – Niveau physique
  - Le gestionnaire des verrous s'assure que:
    - Chaque verrou à un bloc de verrou
    - Chaque bloc de verrou a un ou plusieurs bloc de verrou propriétaire
    - La table des verrous est maintenue pour de meilleures performances

88

## Verrous

- Granularité des verrous et hiérarchie
  - RID, KEY
  - PAGE
  - EXTENT
  - HoBT
  - OBJECT
  - FILE
  - APPLICATION
  - METADATA
  - ALLOCATION\_UNIT
  - DATABASE

89

## Verrous

- Escalade de verrous
  - Réduit la mémoire prise par le gestionnaire de verrous en convertissant plusieurs verrous en un seul verrou agrégé
    - Les verrous de lignes et de pages sont escaladés en verrous de tables
  - Contrôlé au niveau de la table avec l'instruction `ALTER TABLE SET LOCK_ESCALATION`
  - Contrôlé au niveau de la session ou de l'instance avec les flags de trace 1224 et 1211

90

# Verrous

- Modes de verrous
  - Modes de verrous de données
    - Shared
    - Exclusive
    - Update
    - Intent
    - Key-range
  - Modes de verrous spéciaux
    - Schema
    - Conversion
    - Bulk update
  - Les noms des verrous sont abrégés dans les DMVs

91

# Verrous

- Compatibilité des modes de verrous

	Mode accordé					
Mode demandé	IS	S	U	IX	SIX	X
Intent shared (IS)	Yes	Yes	Yes	Yes	Yes	No
Shared (S)	Yes	Yes	Yes	No	No	No
Update (U)	Yes	Yes	No	No	No	No
Intent exclusive (IX)	Yes	No	No	Yes	No	No
Shared with intent exclusive (SIX)	Yes	No	No	No	No	No
Exclusive (X)	No	No	No	No	No	No

92

## Verrous

- Processus de modification des données
  - Pages de données pertinentes localisées dans le BUFFER POOL
  - Verrous avant modification des données
    - Verrou de mise à jour sur les lignes affectées
    - Verrou exclusif sur les pages de la table
    - Verrou exclusif sur la table
    - Verrou partagé dans la base de données
  - Verrous de modification de données
    - Verrou de mise à jour converti en verrou exclusif sur les lignes affectées
    - Verrou exclusif sur les pages de la table
    - Verrou exclusif sur la table
    - Verrou partagé dans la base de données

93

## Verrous

- Indicateurs de requêtes (Hint) pour les verrous
  - Indicateurs de requêtes affectant le mode de verrou
    - ROWLOCK
    - PAGLOCK
    - TABLOCK
    - TABLOCKX
    - UPDLOCK
    - XLOCK
  - Indicateurs de requêtes affectant le niveau d'isolation de table
  - READCOMMITTED
  - READCOMMITTEDLOCK
  - READUNCOMMITTED or NOLOCK
  - REPEATABLEREAD
  - SERIALIZABLE or HOLDLOCK
  - READPAST

94

## Verrous

- Verrous mortels (deadlocks)
  - Les deadlocks sont résolus par le gestionnaire de verrou
    - Par défaut il tourne toute les 5 secondes , la fréquence augmente si les deadlocks augmentent
    - Les victimes de deadlocks sont sélectionnés et annulés
  - Une priorité sur les deadlocks peut être utilisée pour influencer la prise de décision sur le choix d'une victime de deadlock

95

## Verrous

- Latches et Spinlocks
  - Latches
    - Protège les pages en mémoire
    - Latches I/O. PAGEIOLATCH\_ waits
    - Latches Buffer. PAGELATCH\_ waits
    - Latches Non-Buffer. LATCH\_ waits
  - Spinlocks
    - Verrous très légers
    - Cause rarement des problèmes de performance

96



Statistiques et index

## MODULE 6



97

### Aperçu du Module

- Statistiques et estimation de cardinalité
- Index
- Index Columnstore
- Index XML
- Index Spatial
- Index Textuel

98

## Statistiques et estimation de cardinalité

- Optimisation basée sur le coût
  - L'optimisation basée sur le coût sélectionne le plan de requête avec le coût estimé le plus faible pour l'exécution
  - Les statistiques sont une partie critique de l'optimisation basée sur le coût
    - Les statistiques donnent des informations sur la distribution des données dans une colonne ou un groupe de colonne
    - Les colonnes dans les tables et les index peuvent avoir des statistiques

99

## Statistiques et estimation de cardinalité

- Sélectivité des prédicats
  - Prédicats
    - Expression qui est vraie ou fausse
    - Présents dans les jointures, les clauses WHERE et HAVING
  - Sélectivité des prédicats
    - Les nombre de lignes d'une table qui correspond à un prédicat
    - Haute sélectivité = petit pourcentage de lignes retourné
    - Faible sélectivité = grand pourcentage de lignes retourné
  - Sélectivité des prédicats et optimisations des requêtes
    - La sélectivité est utilisé dans les opérations de jointures et pour choisir entre SCAN et SEEK

100

## Statistiques et estimation de cardinalité

- Inspection des statistiques
  - En-tête des statistiques
    - Métadonnées des statistiques
  - Vecteur de densité
    - Mesure d'unicité
  - Histogramme
    - La distribution des données , jusqu'à 200 étapes
  - DBCC SHOW\_STATISTICS

101

## Statistiques et estimation de cardinalité

- Estimation de la cardinalité
  - Dans SQL Server, l'estimation de la cardinalité essaie de prévoir le nombre de ligne retourné par une requête ou un opérateur de requête
  - SQL Server 2014 et 2016 ont subis un changement de logique pour l'estimation de cardinalité
  - Quelques facteurs peuvent amener à une mauvaise estimation de cardinalité:
    - Statistiques dépassées ou manquantes
    - Fonctions dans un prédicat
    - Tables variables

102

## Statistiques et estimation de cardinalité

- Création des statistiques
  - Création automatique
    - Quand AUTO\_CREATE\_STATS = ON
    - Statistiques sur une seule colonne
    - Noms commençant par \_WA..
  - Création manuelle
    - CREATE STATISTICS
    - Sp\_createstats

103

## Statistiques et estimation de cardinalité

- Mise à jour des statistiques
  - Mise à jour automatique
    - AUTO\_UPDATE\_STATISTICS = ON
    - L'option AUTO\_UPDATE\_STATS\_ASYNC
  - Mise à jour manuelle
    - Commande UPDATE STATISTICS
    - Sp\_updatestats
  - La mise à jour des statistiques engendre une recompilation des plans de requête

104

## Statistiques et estimation de cardinalité

- Statistiques filtrées
  - Commande CREATE STATISTICS suivie d'une clause WHERE
  - Certaines limitations
    - Bénéfices limités de AUTO\_UPDATE\_STATISTICS
    - Limité à une simple comparaison logique
    - Ne peut pas être créée sur tous les types de données/colonnes
    - Ne peut pas être créée sur une vue indexée
  - Histogramme à une meilleure résolution

105

## INDEX

- Heaps
  - Une table sans un index clustered est appelée Heap
  - Les pages IAM contiennent des pointeurs contenant les données Heap
    - IAM est le seul lien entre les pages dans une Heap, les lignes des données ne sont pas ordonnées
  - Les Lignes d'une table Heap sont identifiées par un identifiant de ligne (RID= Row identifier)
    - Les index non clustered sur une table Heap contiennent les pointeurs RID

106

# INDEX

- Structure des index
  - L'arbre de recherche balancé (B-Tree)
    - Structure de donnée qui s'auto-balance
    - Une node d'origine (root), plusieurs nodes « non -leaf», plusieurs nodes « leaf »
  - Clé d'index
    - Définit l'ordre des données dans l'index
  - Index Clustered
    - La node root et les nodes « non-leaf » contiennent les pages d'index, les nodes « leaf » contiennent les pages de données
    - Ordre de l'index = ordre des données dans la table
  - Index nonclustered
    - Toutes les nodes contiennent des pages d'index
  - Les Index spatial et XML ont des index spéciaux

107

# INDEX

- Choisir la bonne clé d'index
  - Critères pour les clés des index clustered
    - Unique
    - Non-nullable
    - Restreint
    - Statique
    - En constante augmentation
  - Critères pour les clés des index nonclustered
    - Prédicats utilisés fréquemment
    - Colonnes de jointures
    - Requêtes d'aggrégats
    - Éviter les index redondants ou en doublon, les clés non restreintes, les index utilisés par une seule requête dans les systèmes ayant des données qui changent souvent

108

## INDEX

- Index à une colonne et multi-colonnes
  - Index à une colonne
    - Chaque colonne de prédicat à son propre index
    - Moins de gains de performance mais beaucoup plus réutilisés
  - Index à multi-colonnes
    - Toutes colonnes des prédicats sont incluses dans la clé de l'index
    - Gain de performance plus élevé, mais réutilisation limitée

109

## INDEX

- Index filtrés
  - Index nonclustered avec un prédicat de filtrage
    - Les prédicats de filtrage sont définis avec la clause WHERE dans la définition de l'index
  - Meilleure performance qu'un index sur la totalité de la table
    - Statistique plus fine
    - Taille plus petite
  - Approprié quand les données d'une table sont requêtées par sous-ensembles
  - Certaines restrictions s'appliquent

110

## INDEX

- Les choix des index par l'optimiseur de requête
  - Méthodes d'accès aux données par les index
    - Balayage de table (scan)
    - Accès par index clustered
    - Recherche par un index Nonclustered sur une table Heap
    - Recherche par un index Nonclustered sur une table non-Heap
    - Index nonclustered couvrant
  - Prédicat sargable
    - WHERE <colonne> <opérateur> <valeur>
    - < colonne> doit apparaître seul
    - Un caractère de remplacement devant empêchera l'utilisation d'une recherche dans l'index

111

## INDEX

- Modification des données
  - DELETE
    - La référence à la clé est supprimée du niveau « leaf »
  - INSERT
    - Nouvelle clé de référence ajoutée à l'un ou l'autre:
      - Espace libre dans une page existante
      - Nouvelle page
  - UPDATE
    - DELETE suivi d'un INSERT

112



# INDEX

- Fragmentation des index

## Unfragmented Index

1	2	3	4
---	---	---	---

Physical order matches logical order

## Externally Fragmented Index

2	1	4	3
---	---	---	---

Physical order does not match logical order

## Internally Fragmented Index

1		2		3		4	
---	--	---	--	---	--	---	--

Pages are only partly filled with data (the shaded section of each page represents data)

113

# INDEX

- Ordre des colonnes dans un index
  - Performance du SELECT
    - Seule la première colonne à un histogramme
    - Utiliser la colonne la plus sélective comme première colonne, mais considérer aussi comment l'index sera utilisé
  - Performance de l'INSERT
    - Considérer comment l'ordre des colonnes va affecter les performances de l'INSERT pour un index clustered multicolonne

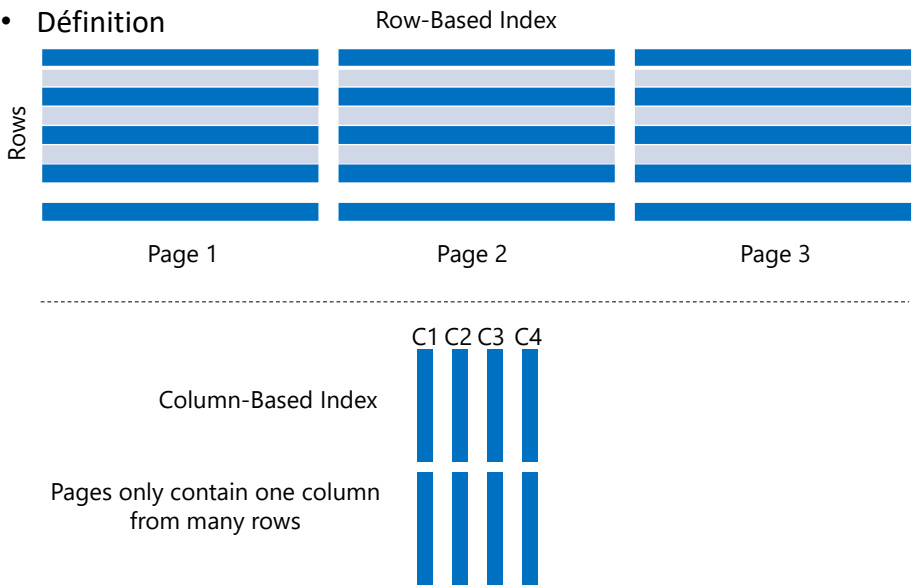
114

# INDEX

- Identifier et créer les index manquants
  - Plans de requêtes
  - Query Store
  - Database Engine Tuning Advisor
  - DMVs dédiées aux index manquants
- Outils à éviter
  - Index clustered suggérés
  - Modifications d’index existants suggérés
  - Analyser l’ordre des colonnes dans un index multicolonne
  - Index filtrés suggérés

115

# INDEX Columnstore



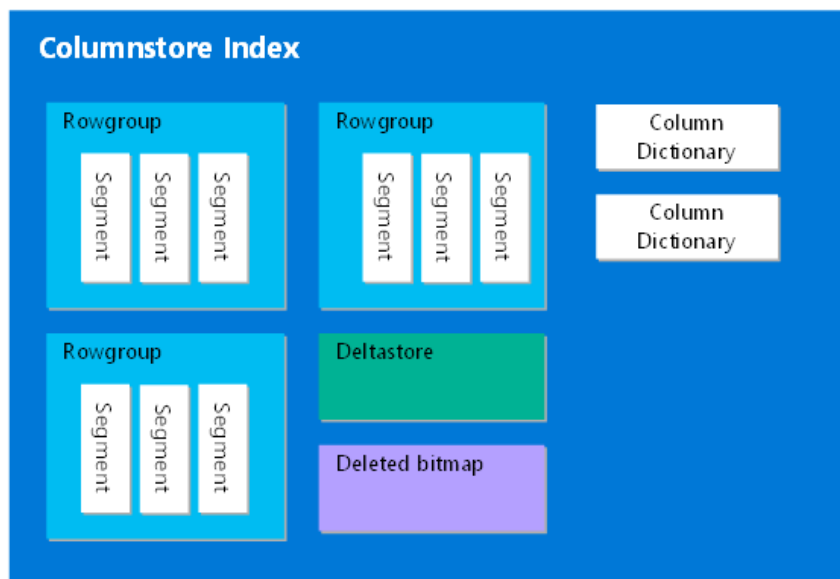
116

## INDEX Columnstore

- Fonctionnalités des index columnstore
  - Les index clustered et nonclustered peuvent être mis à jour en columnstore
  - Chaque table peut avoir seulement un index columnstore
  - Des index columnstore filtrants peuvent être créés
  - Limitations:
    - Types de données restreints
    - Seulement sur les tables ( pas sur les vues ni sur les vues indexées)
    - Pas sur les colonnes « sparse » ( qui sont optimisés pour du stockage de valeurs NULL)

117

## INDEX Columnstore



118

## INDEX XML

- Index XML primaire
  - Fourni un arbre d'objet persistant dans un format interne qui est utilisé accéder plus rapidement aux éléments et attributs dans le XML
  - A besoin d'une clé primaire clustered sur la table
- Index XML Secondaire
  - Peut seulement être construit après qu'un index XML primaire a été créé
  - 3 types d'index peuvent être créés pour améliorer la rapidité des requêtes XML
    - PATH
    - PROPERTY
    - VALUE

119

## INDEX Spatial

- Les index spatiaux dans SQL Server fonctionnent en deux phases
  - Filtre primaire
    - Trouve toutes les lignes candidates possible
    - Peut inclure des faux-positifs
  - Filtre secondaire
    - Supprime les faux-positifs
- Implémentation
  - Créés par l'instruction CREATE SPATIAL INDEX
    - Le type geometry à une configuration BOUDING\_BOX
    - La construction des index ONLINE n'est pas supportée
  - Une seule colonne peut avoir plus d'un index spatial
  - La table doit avoir une clé primaire clustered

120

# INDEX Textuel

- Prérequis
  - Créer un catalogue de texte intégral
  - Types de donnée:char, varchar, nchar, nvarchar, text, ntext, image, xml, varbinary, varbinary(max), incl FILESTREAM
  - Un index de texte intégral par table
  - Colonne non nullable et unique
  - Spécifier le langage pour chaque colonne de l'index
- Implémentation
  - Instruction CREATE FULLTEXT INDEX
- Requêtage
  - CONTAINS et FREETEXT
    - Retourne TRUE ou FALSE
    - A utiliser pour trouver des correspondances
  - CONTAINSTABLE et FREETEXTTABLE
    - Retourne une table
    - Contient une colonne clé et une colonne rang
  - NEAR
    - Recherche des lignes qui contiennent deux mots ou phrases proches l'une de l'autre

121

Exécution des requêtes et analyse du plan d'exécution

## MODULE 7



122

# Aperçu du Module

- Exécution de requêtes et optimiseur de requêtes
- Plans d'exécution de requêtes
- Analyser les plans d'execution de requêtes
- Traitement adaptatif des requêtes

123

## Exécution de requêtes et optimiseur de requêtes

- Phase logique du traitement des requêtes

Statement Clause	Processing Sequence
SELECT	8
DISTINCT <select_list>	9
FROM <left_table>	1
<join_type> JOIN <right_table>	3
ON <join_condition>	2
WHERE <where_condition>	4
GROUP BY <group_by_list>	5
WITH { CUBE   ROLLUP }	6
HAVING <having_condition>	7
ORDER BY <order_by_list>	10
<TOP specification>	11

124

## Exécution de requêtes et optimiseur de requêtes

- Phase physique du traitement des requêtes
  - Analyse syntaxique
    - Valide la syntaxe
    - Sortie: arbre de requête logique
  - Liaison
    - Prend l'arbre de requête logique et le lie aux objets de la base de données
    - Sortie: Arbre d'algèbre
  - Optimisation de la requête
    - Prend l'arbre d'algèbre et sélectionne le plan d'exécution de la requête
    - Sortie: plan d'exécution de la requête
  - Exécution de la requête
    - Exécute le plan d'exécution
    - Sortie: Résultats

125

## Exécution de requêtes et optimiseur de requêtes

- L'optimiseur de requête
  - Arbre logique de requête+ schema + statistiques + règles de transformation= plan d'execution potential
  - Optimisation basée sur le coût
    - Tout les plans ne sont pas considérés
    - De ceux qui sont considérés, le plan le moins couteux sera retourné
    - Plusieurs phases d'optimisations
      - Simplification
      - Plan Trivial
      - Optimisation complete
        - » Recherche 0
        - » Recherche 1
        - » Recherche 2

126

## Plans d'exécution de requêtes

- Plans estimé d'exécution et plan d'exécution réel
  - Chaque requête dans un batch ou une procédure stockée a son plan d'exécution propre
  - Plan d'exécution estimé
    - Le plan est compilé mais pas exécuté
  - Plan d'exécution réel
    - Inclus les informations du comportement estimé et actuel
    - Disponible seulement quand la requête est exécutée
  - Les plans d'exécution estimé et réel seront presque semblables pour la même requête, si ils sont générés en même temps

127

## Plans d'exécution de requêtes

- Formats de plan d'exécution
  - Le plan d'exécution est un arbre hiérarchique d'opérateurs
  - L'opérateur d'origine est l'opérateur final
  - Visualisable dans 3 formats
    - Graphique
    - XML
    - Texte

128



## Plans d'exécution de requêtes

- Capturer les plans d'exécutions
  - Plan graphique
    - A partir de SSMS
    - Peut être stocké et ouvert en tant qu'XML
    - Les "Live Query Statistics" montrent le plan en operation
  - XML
    - SHOWPLAN\_XML: plan estimé
    - STATISTICS XML: plan réel
  - Texte
    - SHOWPLAN\_TEXT: plan estimé
    - SHOWPLAN\_ALL: plan estimé avec les statistiques
    - STATISTICS PROFILE: plan réel

129

## Analyser les plans d'exécution de requêtes

- Opérateurs des plans d'exécutions
  - Les plans d'exécutions sont fait de un ou plusieurs opérateurs
  - Tout les opérateurs ont une sortie , certains supportent une ou deux sorties
  - La plupart des opérateurs ont leur propre icône dans le plan d'exécution graphique

130

## Analyser les plans d'exécution de requêtes

- Opérateurs de récupération de données: SCAN et SEEK
  - L'opérateur SCAN représente la lecture d'une table en entier
    - Peut être coûteux si la table est volumineuse
  - L'opérateur SEEK représente les lignes d'une table qui font référence à un index
    - Un scan peut être moins coûteux si beaucoup de lignes sont retournées
    - Peut seulement être utilisé quand un index approprié est disponible

131

## Analyser les plans d'exécution de requêtes

- Opérateurs de jointures
  - Nested Loops
    - La deuxième entrée est recherchée une fois pour chaque valeur dans la première entrée
    - La deuxième entrée est peu coûteuse à rechercher
  - Merge Join
    - Deux entrées ordonnées sont intercalées
    - Les deux entrées doivent être ordonnées
  - Hash Match
    - Une table Hash est construite à partir de la première entrée et comparée à une valeur Hash de la seconde entrée
    - Utilisée sur les grands volumes en entrée non-ordonnés

132

## Analyser les plans d'exécution de requêtes

- Plans d'exécution parallèles
  - Les plans pour les requêtes parallélisées ne montre pas l'activité de chaque travailleur individuellement
  - Les opérateurs des plans d'exécution utilisant le parallélisme sont indiqués dans le plan d'exécution
  - Les plans parallèles auront au moins une instance de l'opérateur Gather Stream, qui combine les résultats des opérateurs parallèles

133

## Analyser les plans d'exécution de requêtes

- Avertissements dans les plans d'exécution
  - Les opérateurs de plans d'exécution ont parfois des avertissements (warnings) associés
  - Les avertissements indiquent des problèmes qui peuvent avoir des impacts négatifs significatifs sur la performance de la requête
  - Les avertissements sont sérieux et doivent être investigués et résolus

134

## Traitement adaptatif des requêtes

- A propos du traitement adaptatif des requêtes
  - Le traitement adaptatif des requêtes est une nouveauté de SQL Server 2017
  - Il améliore automatiquement les performances des requêtes grâce à trois fonctionnalités:
    - Batch Mode memory grant feedback
    - Batch mode adaptive joins
    - Interleaved execution

135

## Traitement adaptatif des requêtes

- Batch Mode Memory Grant Feedback
  - Stocké dans le plan d'exécution de la requête
  - De mauvaises allocations cause des problèmes à chaque fois qu'un plan d'exécution en cache est lancé
  - Batch Mode Memory Grant Feedback revois le montant idéal de mémoire alloué quand:
    - Une mémoire insuffisante a été allouée
    - Une mémoire excessive a été allouée
  - Utilisez les événements étendus pour les tracer

136

## Traitement adaptatif des requêtes

- Batch Mode Adaptive Joins
  - Sélectionne dynamiquement une jointure Hash ou Nested Loop après que la première entrée soit scannée
  - L'opérateur de jointure est déterminé par le nombre actuel de ligne, et non pas par estimation de cardinalité
  - Les requêtes doivent être éligibles
    - Mode de compatibilité de la base de données à 140
    - Index Columnstore
    - Commande SELECT

137

## Traitement adaptatif des requêtes

- Interleaved Execution
  - Les requêtes utilisant des fonctions table estiment une cardinalité approximative
    - SQL Server 2014 et 2016 => 100 lignes
    - Versions antérieures => 1 ligne
  - Le traitement adaptative des requêtes utilise "Interleaved Execution" et estime ainsi la cardinalité réel afin de traiter le reste de la requête
  - La requête ne doit pas modifier les données ou être référencée dans une clause CROSS APPLY

138

Cache des plans d'exécution et recompilation

## MODULE 8



139

### Aperçu du Module

- Cache des plans de requêtes
- Résolution des problèmes du cache
- Réglage automatique
- Query Store

140

## Cache des plans de requêtes

- Mise en cache et récupération du plan de requête
  - Quatre magasins de cache, pour différents objets
    - Objets
    - SQL
    - Bound Trees
    - Procédures stockées étendues
  - Le plan compilé est retenu dans un compartiment dans le magasin de cache adéquat
  - Un plan peut être réutilisé si le compartiment et la clé du plan de cache concordant
  - Un plan exécutable est une instance spécifique à une session d'un plan compilé

141

## Cache des plans de requêtes

- Gestion du plan de cache
  - Le ménage du plan de cache commence quand le magasin du cache atteint les valeurs de seuil
  - Les plans les moins coûteux sont supprimés en premier du plan de cache
  - On peut vider le plan de cache avec une commande DBCC
  - Videz les plans de cache pour un seul objet avec **sp\_recompile**
  - D'autres opérations vident le plan de cache

142

## Cache des plans de requêtes

- Requêtes sans entrées dans le plan cache
  - Requêtes qui ne peuvent pas être mise en cache
    - Requêtes Ad-hoc et les requêtes T-SQL qui requièrent une résolution de nom d'objet
  - Requêtes marquées en recompilation
    - CREATE...WITH RECOMPILE
    - OPTION (RECOMPILE)
    - EXECUTE...WITH RECOMPILE
  - Les procédures nativement compilées pour les tables optimisées en mémoire

143

## Cache des plans de requêtes

- Maximiser l'efficacité du plan cache
  - Paramétrisation automatique
    - Par défaut, seulement utilisé pour les requêtes possédant une cardinalité distribuée de façon équivalente
  - Optimisé pour les charges de travaux ad-hoc
    - Les plans ad-hoc sont mis en cache seulement à la seconde exécution
  - Requêtes préparées
    - Utilisées par les API de base de données (ODBC, OLEDB)
  - Magasin de plan cache pour les objets
    - Attention au "parameter sniffing"

144



## Cache des plans de requêtes

- Examiner le plan cache
  - `sys.dm_exec_cached_plans`
    - Une ligne par plan mis en cache
  - `sys.dm_exec_query_plan`
    - Plan de requêtes au format XML
  - `sys.dm_exec_text_query_plan`
    - Plan de requêtes au format Texte
  - `sys.dm_exec_plan_attributes`
    - Attributs des plans
  - `sys.dm_exec_query_stats` et `sys.dm_exec_procedure_stats`
    - Statistiques des plans

145

## Résolution des problèmes du cache

- Recompilation et plan d'exécution de requête
  - Une recompilation arrive quand un plan en cache est invalidé et un nouveau plan est compilé
  - Un plan en cache peut être invalidé pour les raisons suivantes:
    - Il devient incorrect (par exemple après un changement sur le schema)
    - Il devient non-optimal (par exemple après un changement dans le statistiques)
  - Une recompilation peut être tracée en utilisant:
    - SQL Profiler
    - Les événements étendus
    - Windows Performance Monitor

146

## Résolution des problèmes du cache

- Problèmes de recompilation
  - « Parameter sniffing » - moins de recompilation qu'attendu. A adressé avec:
    - OPTION(RECOMPILE)
    - OPTION(OPTIMIZE FOR...)
  - Changement dans les statistiques
    - L'indicateur de requête KEEP PLAN réduit les recompilations causées par les changements dans les statistiques
    - L'indicateur de requête KEPPFIXEPLAN prévient les recompilations causées par les changements dans les statistiques
  - Essayer d'éviter les modèles de code qui génèrent des compilations en plus

147

## Résolution des problèmes du cache

- Problèmes du plan cache
  - Gonflement du Plan cache
    - Causé par de multiples copies de la même execution d'une requête qui entre dans le cache
    - Addressé par :
      - Un changement dans le code
      - Option d'optimisation pour les charges de travaux ad-hoc (au niveau du serveur)
  - Monitorer le plan cache avec
    - Statistiques d'attentes liées au plan cache
    - Compteurs de performance Windows

148

## Résolution des problèmes du cache

- Utiliser le plan cache pour guider une optimisation
  - Beaucoup de mesures de performances sont capturées pour les plan cache dans `sys.dm_exec_query_stats` et `sys.dm_exec_procedure_stats`
  - Utilisez ces DMVs pour identifier les requêtes candidates pour les optimisations

149

## Réglage automatique

- Définition
  - Les changements dans l'exécution des plans qui impacte les performances peut faire perdre du temps à trouver et à résoudre
  - Une régression du plan est un plan recompilé qui donne de plus mauvaises performances
  - Réglage Automatique (Automatic Tuning)
    - Collecte des données sur les performances des requêtes et les plans d'exécution
    - Trouve les liens entre les changements de plan et les performances réduites
    - Génère un script pour forcer à utiliser le dernier plan
    - Peut être configuré automatiquement pour appliquer le dernier script

150

## Réglage automatique

- `sys.dm_db_tuning_recommendations`
  - Cette DMV retourne:
    - Un score entre 0 et 100 pour indiquer l'impact sur les performances anticipé
    - Un JSON contenant des recommandations qui inclus:
      - Des métriques utilisés pour identifier les plans qui ont régressés
      - Les commandes utilisées pour appliquer les recommandations
  - Le contenu de cette DMV est retenu jusqu'à ce que l'instance redémarre

151

## Réglage automatique

- Correction des plans
  - Correction des plans automatique
    - Activé au niveau de la base de données
    - Les plans recommandés sont forcés quand:
      - Le gain en CPU > 10 secondes
      - Moins d'erreurs dans le plan recommandé
    - Les résultats sont monitorés
      - Si les performances ne sont pas meilleures, le plan est recompilé
  - Correction des plans manuelle
    - Applique les recommandations manuellement de la DMV
    - Monitore manuellement les performances

152

## Query Store

- Définition
  - Stocke les requêtes, les plans et les statistiques
  - Souligne les modèles d'utilisation de la base de données
  - Aide à la résolution des problèmes de performance
  - Compatible avec les bases de données on-premise et Azure SQL

153

## Query Store

- Activer le Query Store
  - Activation grâce à la commande T-SQL `ALTER DATABASE <nom de la BDD> SET QUERY STORE = ON;`
  - En utilisant l'interface dans les propriétés
  - Ne peut pas être activé pour les bases de données Système

154

## Query Store

- Configurer le Query Store
  - Voir les options du Query Store en utilisant **sys.database\_query\_store\_options**
  - Configurer les options du Query Store en utilisant **ALTER DATABASE SET QUERY STORE (OPTION = VALUE)**
  - Voir et configurer les options du Query Store en utilisant SSMS

155

## Query Store

- Accéder aux données du Query Store
  - On accède aux données du Query Store à travers les DMVs ou SSMS
  - SSMS :
    - Requêtes en régression
    - Consommation global des ressources
    - Les requêtes qui consomment le plus de ressource
    - Les requêtes traquées
  - DMVs
    - DMV du plan cache

156

## Query Store

- Forcer l'exécution de plans
  - SSMS
    - Dans le Query Store cliquez sur Force Plan quand on est sur la vue du plan de requête
  - Transact-SQL
    - Utiliser `sp_querystore_force_plan` pour forcer un plan
  - Vérifier régulièrement les performances des plans forcés

157

Evènements étendus (Extended Events)

## MODULE 9



158

## Aperçu du Module

- Principaux concepts des évènements étendus
- Travailler avec les évènements étendus

159

## Principaux concepts des évènements étendus

- Evènements étendus, SQL Trace, et SQL Server Profiler
  - Les traces SQL et SQL Profiler sont des outils pour collecter des informations sur l'activité sur une instance SQL Serve
  - Evènements étendus est le suceseur de SQL Trace et le remplacera complètement éventuellement
    - SQL Trace et SQL Server Profiler sont marqués comme obsoletes depuis SQL Server 2012
    - Les évènements étendus sont plus flexibles que SQL Trace
      - Supporte les nouvelles fonctionnalités mises en place depuis SQL Server 2012
      - Meilleure flexibilité veut aussi dire plus grande complexité

160



## Principaux concepts des évènements étendus

- Architecture des Evènements étendus
  - Le moteur des évènements étendus donne des moyens
  - L'utilisateur définit la session
    - La session collecte les évènements
      - Un évènement déclenche une action
      - Un évènement est filtré par prédicat
  - Un package définit les objets disponibles à une session

161

## Principaux concepts des évènements étendus

- Packages
  - Les exécutables et les modules exécutables révèlent des packages d'évènements étendus
  - Un package est un container pour d'autres types d'objets
    - Evènements
    - Prédicats
    - Actions
    - Cibles
    - Cartes
    - Types

162

## Principaux concepts des évènements étendus

- Evènements
  - Les évènements sont des points d'enregistrement dans le code exécutable
    - Quand un évènement est déclenché, cela indique que le code associé a été exécuté
    - Retourne les données dans un schéma donné
    - Les évènements sont compatibles avec Event Tracing de Windows

163

## Principaux concepts des évènements étendus

- Prédicats
  - Permet la construction de règles pour filtrer les évènements capturés
    - Constitués de deux sous-catégories
      - Comparaisons (opérateurs logiques =, >, etc...)
      - Sources. Valeurs qui peuvent être utilisées en entrées de comparaisons
    - Des prédicats complexes peuvent être construits ; tout les  $n$  évènements ou toutes les  $n$  secondes.

164

## Principaux concepts des évènements étendus

- Actions
  - Les actions fournissent des informations supplémentaires sur un évènement
    - Chaque évènement peut être lié à une ou plusieurs action
    - Les actions peuvent être déclenchées de façon synchrones après qu'un évènement associé a été déclenché.

165

## Principaux concepts des évènements étendus

- Cibles
  - Les cibles collectent les données des sessions d'évènements étendus
    - Une session peut écrire dans plusieurs cibles
    - Les cibles peuvent être synchrones ou asynchrones
      - Compteur d'évènement: synchrone
      - Fichier d'évènement: asynchrone
      - Association d'évènement: a synchrone
      - Evènement Tracing Window : synchrone
      - Histogramme: asynchrone
      - Buffer circulaire: asynchrone
  - Un évènement sera seulement écrit une seule fois dans une cible

166

## Principaux concepts des évènements étendus

- Sessions
  - Une session lie les évènements aux cibles
    - Les évènements peuvent inclure des actions
    - Les évènements peuvent être filtrés par des prédicats
    - Les sessions sont isolées les unes des autres
    - Une session à un état (démarrée ou stoppée)
    - Une session à un buffer qui détient la donnée de l'évènement

167

## Principaux concepts des évènements étendus

- Types et cartes
  - Types
    - Type de donnée pour les données des évènements étendus
  - Cartes
    - Table de conversion entier vers valeurs texte

168

## Travailler avec les évènements étendus

- Configurer la session
  - Les options de configuration de la session
    - MAX\_MEMORY
    - EVENT\_RETENTION\_MODE
    - MAX\_DISPATCH\_LATENCY
    - MAX\_EVENT\_SIZE
    - MEMORY\_PARTITION\_MODE
    - STARTUP\_STATE
    - TRACK\_CAUSALITY

169

## Travailler avec les évènements étendus

- Configurer les cibles
  - Compteur d'évènement
    - Pas de configuration
  - Fichier d'évènement
  - Association d'évènement
  - Evènement Tracing Window
  - Histogramme
  - Buffer circulaire

170

# Travailler avec les évènements étendus

- La session system\_health des évènements étendus
  - Créée par défaut sur toutes les instances SQL Server 2008 ou ultérieures
    - Démarre au démarrage de l’instance
    - Capture les évènement utiles pour la résolution de problème
    - Buffer circulaire et fichier de cibles

171

# Travailler avec les évènements étendus

- Scénarios d’utilisation

Usage Scenario	Event/Event column	Targets
Execution Time-outs	sqlserver.session_id, sqlserver.tsq_stack	Event pairing
Troubleshooting ASYNC_NETWORK_IO issues	sqlos.wait_info- wait_type of NETWORK_IO	Histogram
Tracking Error Handling in T-SQL code	sqlserver.error_reported- is_intercepted	Any
Tracking Recompilations	sqlserver.sql_statement_recompile sqlserver.tsq_stack- filter on database_id	Any
tempdb Latch contention	Latch_suspend_end	Histogram
Lock Escalation	sqlserver.lock_escalation	Any
Problematic Page Splits	sqlserver.page_split, sqlserver.transaction_log	Any
Troubleshooting Orphaned Transactions	database_transaction_begin and database_transaction_end	Event pairing
Tracking Session Waits	sqlos.wait_info	Any
Tracking Database and Object usage	sqlserver.lock_acquired	Histogram

172

## Travailler avec les évènements étendus

- « Best practices »
  - Lancer les sessions d'évènements étendus seulement qu'en vous en avez besoin
  - Utilisez l'interface SSMS pour parcourir les évènements disponibles
  - Comprenez les limitations des cibles en buffer circulaire
  - Considérez l'impact sur les performances de la collecte des plans d'exécution des requêtes
  - Comprenez le format du graphe des deadlocks

173

Monitoring et analyse de traces

## MODULE 10



174

## Aperçu du Module

- Monitoring et Trace
- Référence et Benchmarking

175

## Monitoring et Trace

- Objets de gestion dynamique (Dynamic Management Objects - DMO)
  - Utilisé pour monitorer la santé d'un serveur SQL
  - Utilisé au niveau du serveur: stocké dans la base master
  - Utilisé au niveau de la base de données: stocké dans le schéma sys de chaque base de données
  - Requêté en utilisant Transact\_SQL

176



## Monitoring et Trace

- **Moniteur de performance de Windows**
  - Affiche en temp-réel les données de performance
  - Monitor la santé du système et des applications
  - Sauvegarde les données de performance dans des fichiers textes ou en base de données
  - Possibilité de créer des ensembles de collecteurs de données personnalisés
  - Peut répondre aux alertes

177

## Monitoring et Trace

- **Moniteur d'activité**
  - Fourni des informations sur:
    - Les requêtes couteuses actives
    - Les I/O sur les fichiers des données
    - Les processus
    - Les requêtes couteuses récentes
    - Les attentes de ressources
    - Le statut du serveur

178

## Monitoring et Trace

- Évènements étendus
  - Packages
  - Cibles
  - Moteur
  - Session

179

## Monitoring et Trace

- Profileur SQL Server
  - Capture, gère et analyse les traces SQL Server
  - Utile pour
    - Trouver les requêtes utilisant beaucoup de ressources
    - Trouver les longues requêtes
    - Rejouer les transactions pour résoudre les problèmes
  - Impacte les performances de l'instance SQL Serve
  - Marqué comme obsolète et pourrait être supprimé d'une future version de SQL Server

180

## Monitoring et Trace

- Traces SQL
  - Fourni en temps-réel les informations sur l'activité du serveur
  - Est créé et configuré en utilisant des procédures stockées système
  - Fait subir moins de ralentissements que le Profileur SQL mais impact quand même les performances

181

## Monitoring et Trace

- Traces SQL par défaut
  - Est activée par défaut
  - Enregistre plusieurs événements différents comme:
    - Changements sur le schéma
    - Monitor la croissance automatique des fichiers
    - Analyse, ordonne et hash les avertissements
    - Monitor les événements liés à la sécurité

182

## Monitoring et Trace

- Analyse des données de trace
  - Profileur SQL
    - Utile pour une analyse préliminaire
    - Ne pas utiliser pour de nouveaux projets
  - Requêtes Transact-SQL
  - Database Engine Tuning Advisor
  - Outil RML Utilities Readtrace

183

## Référence et Benchmarking

- Méthodologie pour créer des références
  - Que capturer?
    - Compteurs Moniteur de performance
    - Sorties DMV
    - Données des traces

184

## Référence et Benchmarking

- Outil d'essai sous contrainte
  - Les essais sous contraintes et les tests de charge peuvent être faits en utilisant Microsoft SQL Server Distributed Replay

185

## Référence et Benchmarking

- Collection des données en utilisant les DMVs
  - Voici les DMVs à utiliser pour capturer des références:
    - `sys.dm_db_index_physical_stats`
    - `sys.dm_db_index_usage_stats`
    - `sys.dm_db_missing_index_details`
    - `sys.dm_os_wait_stats`
    - `sys.dm_exec_requests`
    - `sys.dm_exec_query_stats`
    - `sys.dm_db_file_space_usage`
    - `sys.dm_io_virtual_file_stats`
    - `sys.dm_os_sys_info`
    - `sys.dm_os_sys_memory`
    - `sys.configuration`
    - `SERVERPROPERTY`

186

## Référence et Benchmarking

- Collection des données en utilisant des compteurs du moniteur de performance
  - Utilitaire graphique pour voir en temp-réel la performance du système
  - La collection des données peut être déclenchée en réponse à un évènement
  - La collection des données peut être planifiée

187

## Référence et Benchmarking

- Analyser les données collectées
  - Excel
    - Utiliser l'outil Relog pour créer un csv avec les données
    - Appliquer des agrégations dans la feuille de travail
  - Base de données
    - Utiliser l'outil Relog pour importer les données dans une base de données
    - Analyser les données en utilisant Transact-SQL

188

## Référence et Benchmarking

- Database Engine Tuning Advisor
  - Permet un réglage sans connaissance des bases de données sous-jacentes
  - Accepte plusieurs types de charges
    - Plan cache
    - Trace SQL Profiler
    - Transact SQL
    - XML

189

Conception Objets SQL

## MODULE 11



190

## Aperçu du Module

- Procédures stockées: Best practices
- Fonctions: Best practices
- Alternatives aux déclencheurs
- Dénormalisation

191

## Procédures stockées: Best Practices

- Qualifier les noms dans les procédures stockées
- Garder l'option SET consistante
- SET NOCOUNT ON quand le nombre de lignes affectées n'est pas requis
- Appliquer une convention de nom consistante( et pas de procédures préfixées par sp\_)
- Utiliser @@nestlevel pour visualiser le niveau d'imbrication actuel (max 32)
- Utiliser les codes de retour pour identifier les différents résultats de fin d'exécution
- Garder une procédure pour chaque tâche

192



## Fonctions: Best practices

- Déterminer le type de fonction à utiliser (scalaire, table)
- Créer une fonction pour une tâche
- Qualifier les noms à l'intérieur des fonctions
- Considérer l'impact sur les performances de l'utilisation que vous allez faire de votre procédure
- Les fonctions ne peuvent pas contenir de gestion des erreurs structurées

193

## Alternatives aux déclencheurs

- Beaucoup de développeurs utilisent des déclencheurs ou des alternatives seraient préférables
  - Utiliser des contraintes pour vérifier les valeurs
  - Utiliser DEFAULT pour des valeurs non données pendant un INSERT
  - Utiliser des clés étrangères pour vérifier une intégrité référentiel
  - Utiliser des colonnes calculées
  - Utiliser des vues indexés pour précalculer des agrégats

194

## Dénormalisation

- **Colonnes calculées**
  - Les colonnes calculées sont dérivées d'autres colonnes ou de fonctions
  - Les colonnes calculées sont souvent utilisées pour donner accès aux données sans les dénormaliser
  - Les colonnes calculées persistées améliorent les performances des SELECT des colonnes calculées dans certaines circonstances
- **Vues indexées**
  - Une vue indexée est matérialisée et les données sont stockées sur le disque
  - Les vues non indexées stockent seulement la définition de la vue
  - Une vue indexée n'est pas une table
  - Elles sont utilisées de deux manières
    - Directement dans une requête, dans certaines situation la vue indexée sera plus rapide
    - Indirectement par l'optimiseur de requête