

*Notre vocation,  
votre réussite*

# FORMATIONS ORSYS

---

VOTRE SUPPORT DE COURS

Séminaires  
Cours de synthèse  
Stages pratiques  
Certifications  
Cycles certifiants  
e-Learning

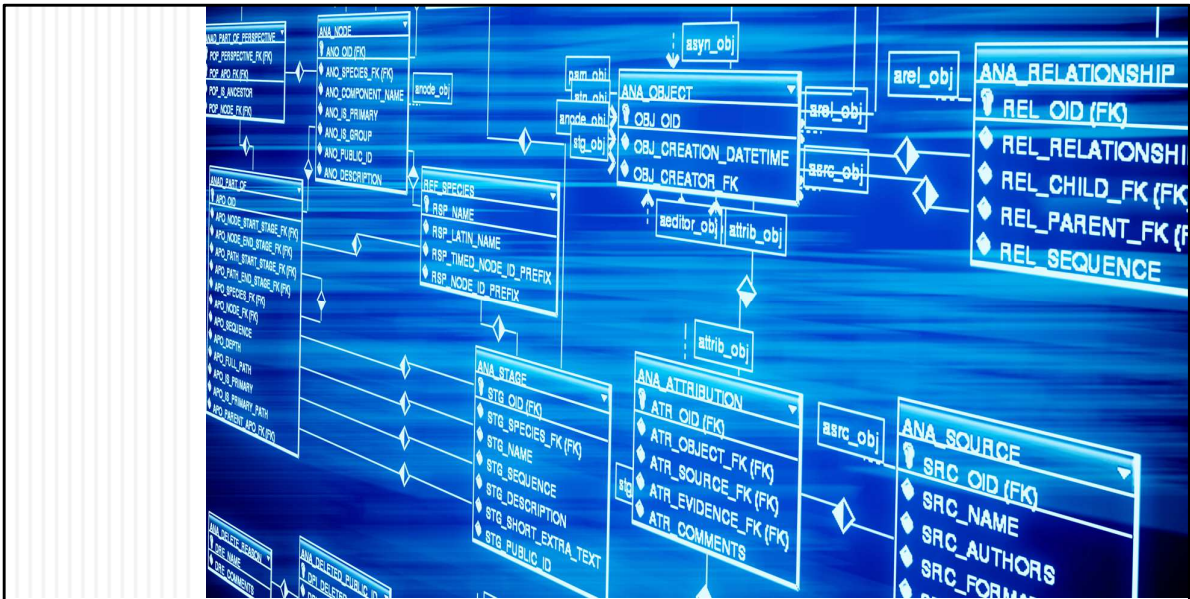
Ce support pédagogique vous est remis dans le cadre d'une formation organisée par ORSYS. Il est la propriété exclusive de son créateur et des personnes bénéficiant d'un droit d'usage. Sans autorisation explicite du propriétaire, il est interdit de diffuser ce support pédagogique, de le modifier, de l'utiliser dans un contexte professionnel ou à des fins commerciales. Il est strictement réservé à votre usage privé.



# SQL SERVER 2012/2017 ADMINISTRATION

Frédéric Brouard (SQLpro) [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
Richard Gaillard - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)

Version 3.001  
Date 10/07/2017



# 0

## Introduction

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

# 0 – Introduction

3



- Qui sommes-nous ?
- Vocabulaire
- Architecture logique
- Particularités
- Licences
- Nature des SGBD Relationnels
- Le relationnel
- L'algèbre relationnelle

# 0 / Qui sommes-nous ?

4

- Richard Gaillard :
  - Spécialiste Oracle
  - Consultant indépendant
  
- Frédéric Brouard
  - Spécialiste SQL Server
  - Sté SQL SPOT (Paris / Paca)

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Nos coordonnées :

Frédéric Brouard (alias SQLpro) : 06 11 86 40 66 – [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
<http://sqlpro.developpez.com/>  
<http://blog.developpez.com/sqlpro/>  
<http://www.sqlspot.com/>

Richard Gaillard (alias Pomalaix) : 06 17 60 00 05 - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)  
[http://www.db1.fr/DB1\\_freelances\\_files/CV%20richard%20gaillard.htm](http://www.db1.fr/DB1_freelances_files/CV%20richard%20gaillard.htm)

# 0 / Qui suis-je ?

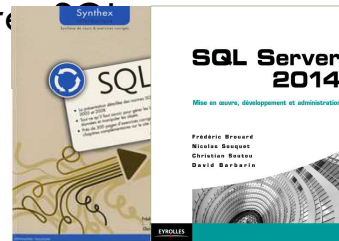


5

Frédéric BROUARD (SQLpro)

« Data Architect » - Sté. SQL Spot (Paris : PACA)

- Spécialiste SQL Server, expert SGBDR
- MVP depuis 14 ans, auteurs de livres
- Audit, optimisation, modélisation, conseil, formation...
- Enseignant école d'ingénieurs,



Université Paul Sabatier  
TOULOUSE III



Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Frédéric BROUARD : 06 11 86 40 66 – SQLpro@SQLspot.com – SQL SPOT SARL (Paris / PACA)

<https://sqlpro.developpez.com/>

<https://blog.developpez.com/sqlpro/>

<http://mssqlserver.fr/>

MVP (Most Valuable Professional) : label de qualité décerné annuellement par Microsoft à des personnalités reconnues pour leur expertise et leur implication dans le partage des connaissances.

<https://mvp.microsoft.com/fr-fr/overview>

<https://mvp.microsoft.com/fr-fr/PublicProfile/4039716?fullName=Arian%20%20Papillon>

# 0 / Qui êtes-vous ?

6

Merci de vous présenter !

- Nom, prénom, entreprise...
- Fonction, emploi...
- Attente de ce cours.



**N'hésitez pas à participer !**

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Vous pouvez poser toutes les questions que vous voulez au cours du déroulement de la formation.

Inutile de lever la main (ça fait perdre du temps...) posez votre question... Interrompez moi ! Sinon vous risquez de perdre le fil de votre pensée...

Les questions très spécifiques seront traitées en fin de journée.



# 0 / Vocabulaire

7

| SQL Server                  |   |
|-----------------------------|---|
| Instance                    | Une installation du moteur avec de multiples bases  |
| Page                        | Unité de stockage logique minimale (faisant toujours 8 Ko)  |
| Sauvegarde transactionnelle | Copie la partie du journal contenant les transactions achevées et purge le journal (moyen de générer des « redo logs ») |
| Base                        | Une ensemble de schémas SQL et d'objets dans les schémas  |
| Schéma SQL                  | Une unité de stockage logique (conteneur d'objets relationnels)   |
| Index clustered             | Table organisée sous forme d'index (à l'opposé, table « heap »)   |
| Extension                   | Bloc de 8 pages contiguës (faisant toujours 64 Ko : 8 x 8 Ko)   |
| Groupe de fichier           | Unité de stockage physique pour les tables et index   |

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Page = Bloc (dans Oracle)

Sauvegarde Transactionnelle = Redo Log / Archive Log (dans Oracle)

Groupe de fichier = table Space (dans oracle)

# 0 / Architecture logique

8

- Une instance :
  - un moteur relationnel (et de stockage)
  - plusieurs bases différentes autonomes (max 32 760)
- Jusqu'à 50 instances par machine
  - réglages de chaque instance séparément
  
- Une base :
  - Contient des schémas SQL

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Chaque instance nécessite une installation supplémentaire du produit dans un répertoire distinct, mais partage certains fichiers, certaines ressources

Une instance peut, par exemple, être configurée pour ne pas utiliser plus d'une certaine quantité de RAM et n'utiliser que certains processeurs (cœurs)...  
Ne pas confondre bases de données et schémas SQL !

Le mieux est de n'avoir qu'une seule instance par machine en production...

Le réglage conjoint de différentes instances est délicat et le multi instance ne doit pas être utilisé en production, sauf cas particulier. Il est néanmoins intéressant pour les machines hors production (développement, qualification, tests...)

# 0 / Particularité...

9

Conformité à la norme SQL :

- Implantation du schéma descriptif  
INFORMATION\_SCHEMA et ses vues
- Identifiants SQL de 128 caractères
- Gestion très étendue des collations
- Syntaxe et niveau de SQL le plus conforme à la norme
- Certaines syntaxes anormatives ont été abandonnées  
(exemple CUBE, ROLLUP...)

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Dans l'ordre du SGBDR le plus conforme au moins conforme, nous trouvons :

- IBM DB2
- SQL Server
- PostgreSQL
- Oracle
- Access
- MySQL

Il est à remarquer que PostgreSQL a régressé sur la norme SQL alors qu'auparavant il était l'un des plus conformes !

À l'inverse Oracle a progressé (ajout types et des fonctions normalisées INT, FLOAT, CAST, COALESCE, ...)

# 0 / Particularité ...

10

- **Multibase :**
  - une instance peut héberger jusqu'à 32 760 bases
  
- **Chaque base est logiquement autonome :**
  - sauvegardes séparées
  - pas d'intégrité référentielle déclarative interbase
  - réglages particuliers possibles au niveau de chacune des bases

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Sous Oracle, le mode multibase appelé multi tenant permet d'avoir plusieurs bases dans une même instance et n'est disponible que depuis la version 12 édition Entreprise. Sous Oracle, le requêtage interbase n'est pas natif et nécessite toujours un lien dblink.

Principales instructions : CREATE DATABASE..., ALTER DATABASE..., DROP DATABASE...

Le mieux est de n'avoir qu'une seule base de données par instance en production. Néanmoins, il existe de nombreuses installations comportant de nombreuses bases. Quelques exemples :

- Marmara : 22 bases, 2 To (2008)
- SPB (le Havre – assurance) : 30 bases, 8 To (2016)
- Transatel (téléphonie) : 60 bases, 14 To (2016)
- Géodis : 130 bases sur une seule instance, dont une de 7 To (2015)
- CER France (CEGID expert) : 10 000 bases, 2 To (2016 - exemple à ne pas suivre !)

# 0 / Particularité ...

11

- Multischéma :
  - Chaque base peut contenir un nombre indéterminé de schémas SQL (conteneurs)
- Propriétaires et schémas ne sont pas conjoints :
  - Un utilisateur SQL peut être propriétaire de plusieurs schémas SQL
  - Un utilisateur peut créer un objet dans un schéma sans pour autant être propriétaire du schéma
  - On peut transférer un objet d'un schéma à l'autre dynamiquement (ALTER SCHEMA ... TRANSFERT ...)

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Principales instructions : CREATE SCHEMA..., ALTER SCHEMA..., DROP SCHEMA...,

Il est intéressant de créer plusieurs schémas pour une même base afin :

- de structurer les objets de manière verticale (ERP par exemple) ou horizontale (données communes, administratives ou « système »...)
- de gérer la sécurité au niveau schéma (possibilité de donner des privilèges à travers les schémas) => sécurité dynamique !

## 0 / Particularité ...

12

- Logique naturellement multibase et optimisée
- Pas de « dblink »
- Syntaxe par notation pointée (database.schéma.objet)

```
SELECT *
```

```
FROM master.sys.databases AS db
```

```
JOIN msdb.dbo.suspect_pages AS sp
```

```
ON db.database_id = sp.database_id;
```

- Requêtes interbases optimisées
- Chaque base dispose de son journal de transactions

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

La requête présente une jointure entre une vue système (databases) de la base *master* contenue dans le schéma *sys* et une table système (suspect\_pages) de la base *msdb* contenue dans le schéma *dbo*.

Il est possible de faire des requêtes avec des serveurs externes via le concept de « serveurs liés » (linked servers), par exemple avec Oracle (8, 9, 10, 11), DB2, Sybase, SQL Server et toute autre base ou « table » attaquant via ODBC (MySQL, PostgreSQL, Excel, fichiers texte...). La syntaxe est la même avec en sus le nom de serveur en préfixe.

# 0 / Particularité ...

13

## Optimisation :

- Deux niveaux :
  - sémantique (contraintes, dont CHECK et FK)
  - statistiques (d'index et de colonne)
- Parallélisme automatique
- Coût des requêtes indépendant du matériel (étalonnage)

Le coût de requête n'a pas d'unité.

Il est souhaitable de limiter le parallélisme, sauf en contexte OLAP, car SQL Server a tendance à trop en faire.

# 0 / Particularité ...

14

- Transactions :
- Porte sur l'intégralité des commandes SQL :
  - DML (INSERT, UPDATE, DELETE...)
  - DDL (CREATE, ALTER, DROP ...)
  - DCL (GRANT, REVOKE...)
- Niveau d'isolation :
  - 5 (4 normatifs + SNAPSHOT)
  - Dynamique
- SELECT FOR UPDATE inutile !

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Les niveaux d'isolation normatifs sont :

READ UNCOMMITTED (lectures sales)

READ COMMITTED (lecture propre, valeur par défaut)

REPEATABLE READ (lecture répétable)

SERIALIZABLE (mise en série – pas de concurrence)

Le niveau SNAPSHOT est un versionnement de lignes. Il peut être utilisé dans deux modes :

- SNAPSHOT
- READ COMMITTED SNAPSHOT.

Dans ce dernier cas, le READ COMMITTED est remplacé par un SNAPSHOT

Dynamique : on peut changer de niveau d'isolation à tout moment, y compris en plein milieu d'une transaction et une même requête peut avoir des tables ayant différents niveaux d'isolation...



# 0 / Particularité ...

15

- Sécurité à deux niveaux :
  - Compte de connexion : règle les privilèges pour les opérations d'instance (création des bases, restauration des bases...)
  - Utilisateur SQL : règle les privilèges de manière indépendante dans chacune des bases (sauvegarde de la base, lecture des tables...)
  - Un compte de connexion est rattaché à au plus un utilisateur SQL dans chacune des bases

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Les comptes de connexion sont stockés dans la base système *master* qui est la base centralisatrice du serveur. Ils servent à accéder à l'instance (authentification par compte système Windows ou compte SQL)

Les utilisateurs SQL sont stockés dans les différentes bases du serveur (bases système ou de production). Ils règlent les problèmes de sécurité au niveau de la base et sont liés à un compte de connexion

CREATE LOGIN est la commande pour créer des compte de connexion pour régler la sécurité au niveau serveur. Autres commande de sécurité au niveau serveur :

CREATE SERVER ROLE... : pour créer un rôle personnalisé de serveur (il existe des rôles de serveur prédéfinis, par exemple sysadmin)

ALTER SERVER ROLE... ADD MEMBER : pour donner à un compte de connexion un rôle de serveur

CREATE USER est la commande pour créer des utilisateurs de bases de données afin de régler la sécurité au niveau de la base. Autres commandes de niveau base :

CREATE ROLE ... : pour créer un rôle personnalisé de niveau base (il existe des rôles de bases de données prédéfinis, par exemple db\_owner)

ALTER ROLE... ADD MEMBER : pour donner à un utilisateur SQL d'une base un rôle de base de données

Il est néanmoins possible de créer des utilisateurs SQL qui se connectent directement à une base sans passer par le compte de connexion, par le biais des « CONTAINED DATABASES » disponibles depuis la version 2012.

# 0 / Particularité ...

16

- Index couvrant :
  - Ajoute une clause INCLUDE pour éviter une
- Index verticaux (ColumStore)
- Vues indexées toujours synchrones
- « In Memory » :
  - Apparue avec la version 2014
  - Étendue avec la version 2016
  - Incluse dans la version Enterprise et 2016 SP1
- Procédures compilées natives
- « Stretch » tables : tables en cloud (2016)

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Les « stretch » tables sont des tables mises en cloud dans une base locale, par exemple pour des besoins d'archivage. Cette fonctionnalité est apparue avec la version 2016

# 0 / Avantages SQL Server...

17

- Offre de versions très large :
  - LocalDB (gratuite, OLTP seulement : 1 base)
  - Compact Edition (gratuite, OLTP seulement)
  - Express (gratuite, 3 déclinaisons, OLTP seulement)
  - Web (en mode SPLA, peu onéreuse, OLTP seulement)
  - Standard (BI incluse)
  - Enterprise (BI incluse, 24h24 / 7j7) et *Developer*
  - Azure Database (cloud)

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

L'édition BI correspond à l'édition Enterprise pour la partie BI et l'édition Standard pour la partie OLTP.

L'édition **Standard** est conçue pour des bases de moyenne à grosse taille (1 To au plus) qui n'ont pas de besoin de production 24h sur 24 et 7j sur 7, c'est-à-dire disposant d'heures creuses pour la maintenance.

L'édition **Enterprise** est conçue pour les grosses bases (plusieurs To) fonctionnant 24h / 24 et 7j / 7 sans heures creuses.

Ce qui fait la différence est essentiellement :

- L'indexation ONLINE (CREATE / ALTER INDEX ... WITH (ONLINE = ON))
- La compression et le partitionnement des données
- Des méthodes d'audits complémentaires pointues (Database Audit, CDC...)
- Des méthodes de sécurité complémentaires (TDE, HSM...)
- La gestion de la haute disponibilité via AlwaysOn (cluster de basculement multi nœuds)
- Le traitement de données transactionnelles « in memory »
- L'ajout à chaud de RAM et de CPU (suivant hardware)
- Une optimisation avancée (gouverneur de ressources, guide de plans, automatisation des vues indexées...)
- Des opérations de bases de données particulières (cliché, restauration « on line »...)

L'inconvénient est que cela nécessite plus de ressources

Comparaison des éditions Localdb, CE et express :

<http://blogs.msdn.com/b/jerrynixon/archive/2012/02/26/sql-express-v-localdb-v-sql-compact-edition.aspx>

Il existe en sus l'édition **Developer** qui est strictement équivalente à l'édition Enterprise et s'installe sur les machines desktop. Elle ne peut pas être utilisée en production.

Pour les tests et le développement, vous pouvez utiliser l'édition Developer sur un serveur. Mais si vous êtes abonné au MSDN, alors vous pouvez utiliser toutes les éditions de SQL Server à titre de développement, tests ou essais.

# 0 / Avantages SQL Server...

18

## SQL Server et le NoSQL...

- En sus du relationnel, possibilité de faire :
  - Du « document db » avec FTS et FILESTREAM/FILETABLE
  - Du « big table » avec partitionnement, indexation ColumStore
  - De la paire « clef/valeur » en cache avec « in memory »
  - Du graphe depuis la version 2017...
- À surveiller aussi : CosmosDB dans le cloud Azure !

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

FTS : Full Text Search, inclus la recherche dans les fichiers électroniques, y compris pour les métadonnées et la recherches sémantique (v. Enterprise pour cette dernière)

Dans la version 2017, vous pouvez créer des tables de nœuds et d'arêtes pour gérer des graphes. La syntaxe SQL a été étendue pour naviguer dans ces graphes. Le modèle est calqué sur Neo4J.

# 0 / Avantages SQL Server...

19

- Pas de « module » à rajouter, à activer, à payer...
- Services Windows séparés pour les différentes briques fonctionnelles :
  - SQL Server (bases OLTP)
  - Analysis Services (SSAS : bases OLAP et Data mining)
  - Integration Services (SSIS : ETL)
  - Reporting Services (SSRS : solution d'édition d'états)

Mais certains modules ne sont disponible qu'en version Enterprise

# 0 / Avantages SQL Server...

20

Facilités de compréhension :

- Interfaces des outils en français
- Documentation officielle en français
- Communautés officielles en français
- Hotline MS compétente
- Éditeur à l'écoute

## SQL Server 2014

Mise en œuvre, développement et administration

Frédéric Brouard  
Nicolas Souquet  
Christian Soutou  
David Barbarin

EYROLLES

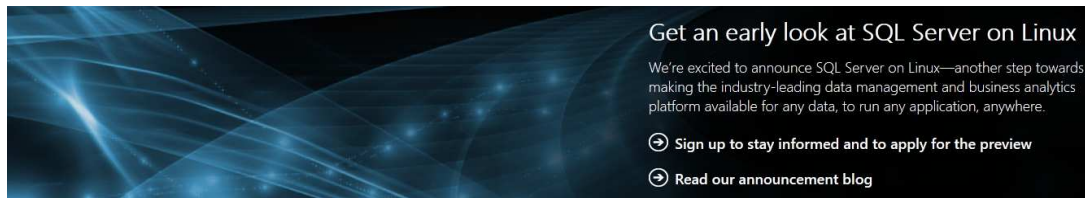
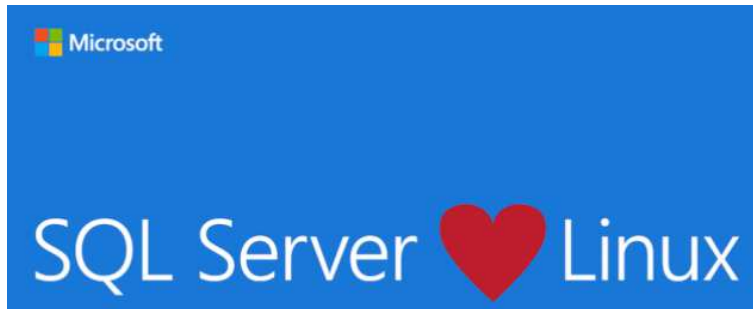
Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

# 0 / Inconvénient SQL Server...

21

## Plateforme Windows ...



Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Depuis la version 2017, Microsoft SQL Server est disponible sous Linux

<https://www.microsoft.com/fr-fr/sql-server/sql-server-downloads>

- Red Hat Enterprise Linux 7,3
- Ubuntu Linux 16,04
- SUSE Linux Enterprise Server v12 SP2

Et sous Docker

# 0 / Fonctionnalités SQL Server...

22

## □ Matrice fonctionnelle :

| EDITION                          | OLTP    | SSAS    | SSIS | SSRS    |
|----------------------------------|---------|---------|------|---------|
| <b>Express</b>                   | OUI (1) | NON     | NON  | NON     |
| <b>Exp. w. Tools</b>             | OUI (1) | NON     | NON  | NON     |
| <b>Exp. W. Advanced Services</b> | OUI (1) | NON     | NON  | OUI (2) |
| <b>Web</b>                       | OUI (3) | NON     | NON  | OUI (3) |
| <b>Standard</b>                  | OUI (4) | OUI (4) | OUI  | OUI (4) |
| <b>Enterprise</b>                | OUI (6) | OUI (6) | OUI  | OUI (6) |

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Limites pour la version 2014 :

- 1) Dans les limites de l'édition Express, c'est-à-dire bases limitées à 10 Go de données relationnelles, 1 Go de cache, 4 cœurs
- 2) Limites spécifiques de RAM à 4 Go
- 3) Limité à 16 cœurs, 64 Go de cache pour les bases OLTP et 64 Go de cache pour SSRS
- 4) Limité à 16 cœurs, 128 Go de cache pour les bases OLTP et 64 Go de cache pour les bases OLAP et 64 Go de cache pour SSRS
- 5) Mêmes limites que la standard pour les bases OLTP, pas de limites pour les bases OLAP et SSRS
- 6) Aucune limite

Taille maximale d'une base relationnelle : 524 Po



# 0 / Fonctionnalités SQL Server...

23

- Fonctionnalités incluses dans le moteur relationnel :
  - Indexation textuelle (conforme à la norme SQL)
  - Type XML natif (avec support de XQuery / XPath)
  - SIG avec objets géométriques et géographiques (conforme à l'OGC)
  - Routines CLR (.net en sus de Transact SQL)
  - Données Filestream et FileTable (fichiers)
  - Tables de graphes (nœuds et arrêtes)

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Ces fonctionnalités sont incluses dans toutes les éditions à partir de l'édition EXPRESS...

# 0 / Licences

24

- 2 modes :
  - Par client
    - les accès doivent être clairement identifiés
    - Impossible pour site web !
  - Par serveur :
    - Impératif pour site web
    - Intéressant si plus de
      - 50 clients (Standard)
      - 100 clients (Enterprise)
- Les serveurs froids (haute dispo) sont gratuits

Administration SQL Server - CH 00 - introduction

V3.0 10/07/2017

Les serveurs utilisés exclusivement pour la haute disponibilité ne font pas objet d'un paiement de licence et peuvent être testés au cours de 30 journées au maximum

La haute disponibilité :

- via AlwaysOn : disponible qu'en édition Enterprise pour les version 2012 à 2016 et nécessite le paiement de la « Software Insurance »
  - À partir de la version 2016 SP1 édition Standard, AlwaysOn peut fonctionner base par base avec une limite de 2 replicas
- via Mirroring (*deprecated* à partir de 2012) : disponible à partir de la standard
- via Log Shipping : possible dans toutes les éditions, mais automatisée à partir de la Web

# 0 / Nature des SGBDR

25

SGBD Relationnel... inventé par Codd en 1970 !

- Forme particulière des bases de données dans lesquelles les informations sont stockées dans des « relations »

Dictionnaires de français  
**LAROUSSE**

 **relation**  
(latin *relatio*)

- Action de rapporter en détail ce dont on a été le témoin ou dont on a eu connaissance ; récit qu'on en fait : *Faire la relation des événements auxquels on a participé.*

Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

Les « relations » relatent l'information...

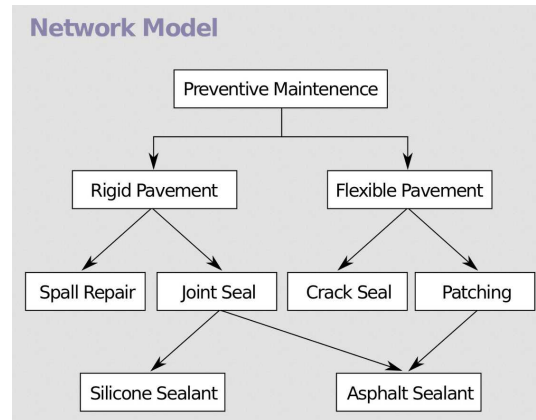
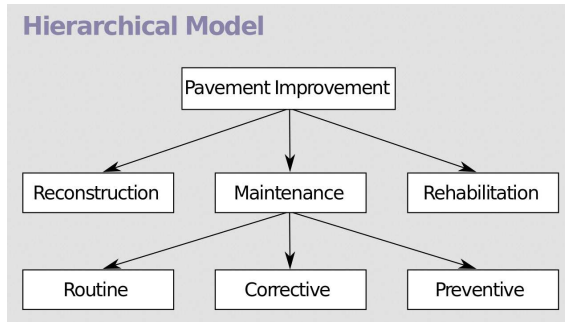
Auparavant, il existait des bases de données organisées sous forme de hiérarchies ou de « réseaux » (en fait on devrait parler de « graphe »).

# 0 / Nature des SGBDR

26

La situation avant Codd :

- Bases hiérarchique
- Bases « réseau »
- Langage « fichiers »



Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

Image des modèles hiérarchique et réseau : wikipedia

# 0 / Nature des SGBDR

27

Prédominance du « hardware » :

- fichiers, enregistrements, pointeurs...

Le papier de Codd (juin 1970)

- « A Relational Model of Data for Large Shared Data Banks »

*Information Retrieval*

## A Relational Model of Data for Large Shared Data Banks

E. F. Codd  
*IBM Research Laboratory, San Jose, California*

Optimisation SQL Server - CH 01 - introduction

```
INITIALIZATION.  
READY PRESIDENTIAL-AREA.  
OPEN non-database COBOL files.  
MOVE "FALSE" TO NO-MORE-STATES.  
FIND FIRST STATE IN ALL-STATES-SS.  
PERFORM PROCESS-STATE THRU FINISH-STATE  
UNTIL NO-MORE-STATES = "TRUE".  
GO TO FINISH-UP.  
PROCESS-STATE.  
MOVE 0 TO PRESIDENT-COUNT.  
IF NATIVE-SON IS EMPTY  
MOVE "TRUE" TO NO-MORE-SONS,  
ELSE MOVE "FALSE" TO NO-MORE-SONS.  
PERFORM COUNT-NATIVE-SONS  
UNTIL NO-MORE-SONS = "TRUE".  
GO TO FINISH-STATE.  
COUNT-NATIVE-SONS.  
FIND NEXT PRESIDENT IN NATIVE-SON.  
IF DATABASE-STATUS = DONE  
MOVE "TRUE" TO NO-MORE-SONS  
ELSE ADD 1 TO PRESIDENT-COUNT.  
FINISH-STATE.  
IF PRESIDENT-COUNT IS GREATER THAN 1  
FIND STATE CURRENT,  
GET STATE,  
write out state name and president count.  
FIND NEXT STATE IN ALL-STATES-SS  
IF DATABASE-STATUS = DONE  
MOVE "TRUE" TO NO-MORE-STATES.  
FINISH-UP.  
FINISH PRESIDENTIAL-AREA.  
CLOSE non-database COBOL files.  
STOP RUN.
```

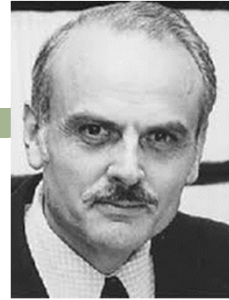
V3.0 10/07/2017

Un exemple de LMD pour les bases de l'époque. Notez la notion de fichier sous-jacente (OPEN, CLOSE...) et les boucles.

Extrait papier fondateur des bases de données relationnelles de Franck Edgar Codd « A Relational Model of Data for Large Shared Data Banks » ACM/org. Disponible en copie pdf à : <https://cs.uwaterloo.ca/~david/cs848s14/codd-relational.pdf>

# 0 / Nature des SGBDR

28

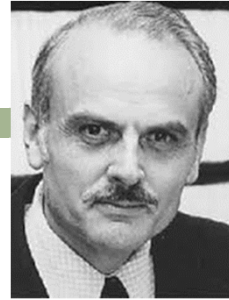


Conception de Codd :

- **Séparation entre hardware et données...**
  - Pas de dépendance des applicatifs au niveau :
    - Ordre des données
    - Indexation des données
    - Situation des données (chemin d'accès aux fichiers)
- Impose une **solution logique** via le concept de « relation »
- Se base sur une approche purement **sémantique** des données.
  - Pas de numéro de ligne (car pas d'ordre), pas de pointeurs, pas d'enregistrement...
  - **Rapprochement des données par valeurs !**

# 0 / Nature des SGBDR

29



Les modèles de bases de données précédents mélangeaient les aspects physique et logique...

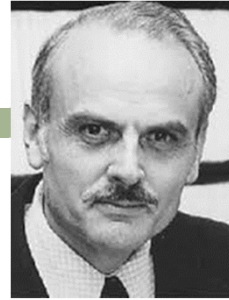
La notion de séparation entre les aspects :

- purement physiques (stockage, indexation...)
- strictement logique (écriture des requêtes)

est **FONDAMENTALE !**

# 0 / Nature des SGBDR

30



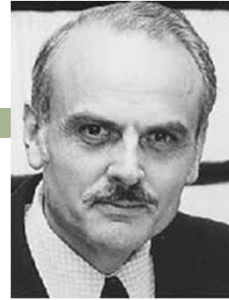
## Physique et logique...

- Le **développeur** ne doit pas penser aux conséquences de son code (requêtes et routines SQL), mais appliquer les bonnes pratiques.
- **L'administrateur** ne doit pas influencer sur la structure des objets ni sur l'écriture du code. Il doit s'occuper de gérer les index, le stockage, les statistiques... et remonter aux développeurs le code anormalement long s'il ne peut en diminuer la durée d'exécution.
- Le **modélisateur** doit se préoccuper de générer un modèle normalisé et efficace



# 0 / Nature des SGBDR

31



## Vision de Codd :

- Purement mathématique...
  - La relation est un objet mathématique basé sur la théorie des ensembles
- Aux relations sont associés des opérateurs...
  - Algèbre relationnelle
- Pas de « liens » prédéfinis !
  - Rapprochement des données par valeurs et non par lien physique (pointeurs)

# 0 / Nature des SGBDR

32

- Achèvement de l'œuvre de Codd...
- Problématique transactionnelle.
  - Que faire en cas de panne si :
    - un ensemble de données doit être modifié
    - des informations de deux relations doivent être modifiées conjointement
- Solution de Gray et Bernstein par « journalisation »



Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

James Gray (12/1/1944 - 28/1/2007 perdu en mer) Chercheur pour Digital, Tandem, IBM et AT&T, puis Microsoft – Prix Turing en 1998 pour ses travaux sur les bases de données et en particulier la gestion des transactions informatiques.  
Philip Bernstein est encore actif en tant que chercheur pour Microsoft.

# 0 / Le relationnel

33

- SGBD Relationnels :
- Approche mathématique basé sur la **théorie des ensembles**
  - Pas d'ordre « naturels » ou « par défaut »
  - Les ensembles sont des « sacs »
  
- Quel est la dernière bille mise Dans le sac ?



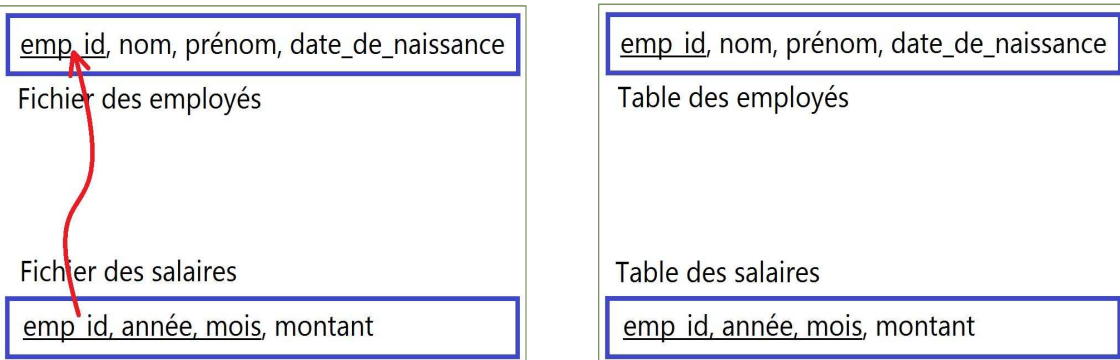
# 0 / Le relationnel

34

SGBD Relationnels :

- Rapprochement des données par des valeurs

*Une anecdote en passant...*



Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

Présentation effectuée par Chris Date lors du congrès CODASYL de 1973...

# 0 / Le relationnel

35

SGBD Relationnels :

- Opérations « ensemblistes »

*RÈGLE 7 - Insertion, suppression et modification ensemblistes :*

*Le SGBDR retourne un ensemble d'éléments en réponse aux requêtes qui lui sont soumises. Il doit pouvoir mettre à jour un ensemble d'éléments en exécutant une seule requête.*

Autrement dit : les opérations doivent porter aussi bien sur des lots de tuples issues de plusieurs tables que juste pour un tuple seul issu d'une unique table.



**ATTENTION : SQL ne veut pas dire « relationnel » !**

Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

Succès du relationnel : années 1985 / 1990

Problème, le terme « relationnel » est appliqué à tout et n'importe quoi !

Codd se fâche et propose de vérifier cela à l'aide des « douze règles de Codd » permettant de définir ce qui est exigé d'un système de gestion de base de données (SGBD) afin qu'il puisse être considéré comme relationnel...

Ces règles sont basées sur ses travaux originaux effectués à partir de 1970, et ont été publiées dans deux articles de vulgarisation du magazine Computerworld (octobre 1985) :

- *Is your DBMS really relational ?*
- *Does your DBMS run by the rules ?*

À lire : <http://sqlpro.developpez.com/SGBDR/ReglesCodd/>

# 0 / Le relationnel

36

Séparation entre

- aspects logiques :
  - Comment manipuler les données
- Aspects physique :
  - Stockage, indexation, distribution, répartition, tri...

Il existe donc 2 moteurs et 2 métiers...

**Moteur relationnel :**

- modélisateur
- développeur

**Moteur de stockage :**

- DB administrateur

Il faut impérativement conserver cette scission. Le mélange des genres conduit souvent à la catastrophe !

# 0 / Le relationnel

37

Structure d'une relation :

- possède un nom (unique au sein de la base de données)
- possède une collection d'attributs (au moins un)
- possède une clef primaire
- peut posséder d'autres clefs appelées indifféremment :
  - clefs alternatives
  - clefs subrogées



**ATTENTION : pas de relation sans clef primaire !**

Une clef alternative ou subrogée se traduit en SQL par une contrainte UNIQUE

# 0 / Le relationnel

38

À quoi sert la **clef** ?

- À accéder, au plus, à un tuple de la relation !

Corolaire :

- Tous les tuples d'une relation sont différents

C'est la notion d'unicité !



Différents ou distinct ou unique !



# 0 / Le relationnel

39

Définition de l'attribut :

- Nom unique au sein de la relation
- Possède un domaine de valeur
- Possède toujours une valeur atomique



**ATTENTION : un attribut est toujours valué (pas de NULL) !**

# 0 / Le relationnel

40

Domaine :

- Ensemble des valeurs possibles pour un attribut

Exemples :

- Pourcentage de réduction ? [0 .. 100]
- Prix ? [0 .. 100000000000000] en €
- Vitesse ? [0 .. 300000] en km/s
- Angle ? [-∞ .. + ∞]
- Date de naissance ? [1/1/1 .. {aujourd'hui}]
- Patronyme (français) ?  
Lettres + certains caractères  
( ' - blanc ) sauf aux extrémités

Différents ou distinct ou unique !

# 0 / Le relationnel

41

Atomicité ?

Atome : *plus petite partie insécable de la matière (grecs)*

Donnée atomique => impossible de la découper,  
fragmenter sans perte de sens

L'atomisation des données est le point de départ de la  
modélisation des données



**ATTENTION : l'atomicité est souvent violée => performances catastrophiques**

Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

L'atomicité est l'une des choses les plus difficile à appliquer et est souvent mal comprise.

Son non respect conduit à des bases mal modélisée et très peu performants.

# 0 / Le relationnel

42

## Différences entre « tables » et « relation »

|             | Table                  | Relation                    |
|-------------|------------------------|-----------------------------|
| Clef        | Pas obligatoire        | Obligatoire                 |
| Valeurs     | NULLs possible         | Toujours renseignée         |
| Atomisation | Pas toujours respectée | Obligatoire et fondamentale |

Une table est relationnelle si et seulement si :

- Elle possède une clef
- Toutes ses colonnes sont valuées
- Les valeurs sont toutes atomiques



**ATTENTION : les SGBDR sont conçus pour manipuler des relations (pas des tables !)**

Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

En principe une modélisation relationnelles parfaite répond aux 4 règles suivantes :

- 1 - données atomiques
- 2 – pas de NULL
- 3 – pas de redondance
- 4 – la modification d'une information, ne doit pas conduire à mettre à jour plus d'une ligne

Si ces 4 règles sont respectées, la base est en 6<sup>e</sup> forme normale.

# 0 / Algèbre relationnelle

43

Ensemble des opérations portant sur les relations

Les opérations génèrent en sortie une nouvelle relation  
=> Fermeture

- Opérations unaires (ou monadique) : une seule relation présente dans l'opération
- Opérations binaires (ou dyadiques) : deux relations présentent dans l'opération

Dans certains cas, les opérations peuvent porter sur plus de 2 relations...

# 0 / Algèbre relationnelle

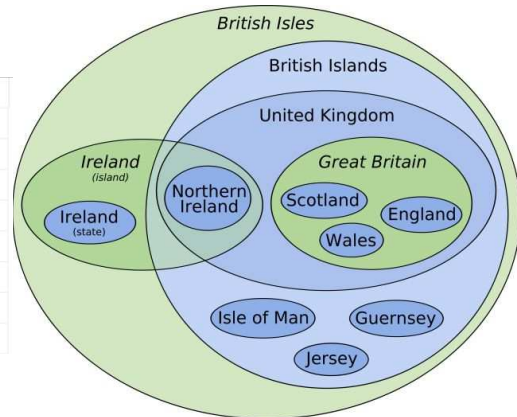
44

## Représentation des relations...

Sous forme tabulaire

| name          | create_date             | collation_name |
|---------------|-------------------------|----------------|
| DB_VUE_INDEX  | 2017-03-28 14:19:49.137 | French_BIN     |
| msdb          | 2016-04-30 00:46:38.773 | French_BIN2    |
| master        | 2003-04-08 09:13:36.390 | French_BIN2    |
| DB_ODS_SPACEX | 2017-04-27 15:28:17.153 | French_BIN2    |
| DB_TEST_ITM   | 2017-04-27 10:54:40.177 | French_BIN2    |
| DB_HOTEL_SIG  | 2017-03-30 14:33:15.863 | French_CI_AS   |
| DB_GEO        | 2017-03-19 19:43:56.940 | French_BIN2    |
| DB_MEDICTACT  | 2017-03-27 15:19:21.593 | French_BIN2    |

Sous forme ensembliste



# 0 / Algèbre relationnelle

45

Monadiques :  
Restriction

| name          | create_date             | collation_name |
|---------------|-------------------------|----------------|
| DB_VUE_INDEX  | 2017-03-28 14:19:49.137 | French_BIN     |
| msdb          | 2016-04-30 00:46:38.773 | French_BIN2    |
| master        | 2003-04-08 09:13:36.390 | French_BIN2    |
| DB_ODS_SPACEX | 2017-04-27 15:28:17.153 | French_BIN2    |
| DB_TEST_ITM   | 2017-04-27 10:54:40.177 | French_BIN2    |
| DB_HOTEL_SIG  | 2017-03-30 14:33:15.863 | French_CI_AS   |
| DB_GEO        | 2017-03-19 19:43:56.940 | French_BIN2    |
| DB_MEDICACT   | 2017-03-27 15:19:21.593 | French_BIN2    |

**Restriction** : consiste à ne retourner que certains tuples

Projection

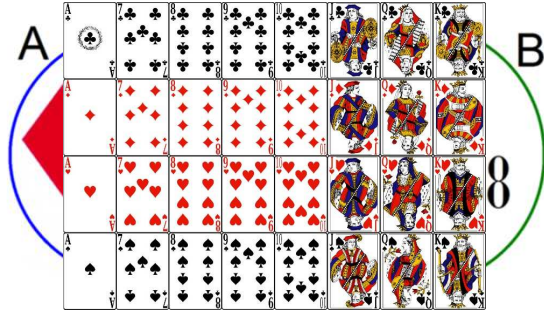
| name          | create_date             |
|---------------|-------------------------|
| DB_VUE_INDEX  | 2017-03-28 14:19:49.137 |
| msdb          | 2016-04-30 00:46:38.773 |
| master        | 2003-04-08 09:13:36.390 |
| DB_ODS_SPACEX | 2017-04-27 15:28:17.153 |
| DB_TEST_ITM   | 2017-04-27 10:54:40.177 |
| DB_HOTEL_SIG  | 2017-03-30 14:33:15.863 |
| DB_GEO        | 2017-03-19 19:43:56.940 |
| DB_MEDICACT   | 2017-03-27 15:19:21.593 |

**Projection** : consiste à ne retourner que certains attributs

# 0 / Algèbre relationnelle

46

Dyadique :  
Produit cartésien



**Produit cartésien** : consiste à associer chaque élément d'un ensemble à chacun des éléments de l'autre

Optimisation SQL Server - CH 01 - introduction

Division relationnelle



**Division** : consiste à retrouver les éléments d'un ensemble associés à tous ceux d'un autre ensemble

V3.0 10/07/2017

Exemple de division relationnelle : comment faire pour savoir quelles sont les couleurs complètes dans un jeu de carte auquel il manque certaines cartes...

Réponse : répartir par couleur et vérifier s'il y a bien le bon nombre de cartes.



# 0 / Algèbre relationnelle

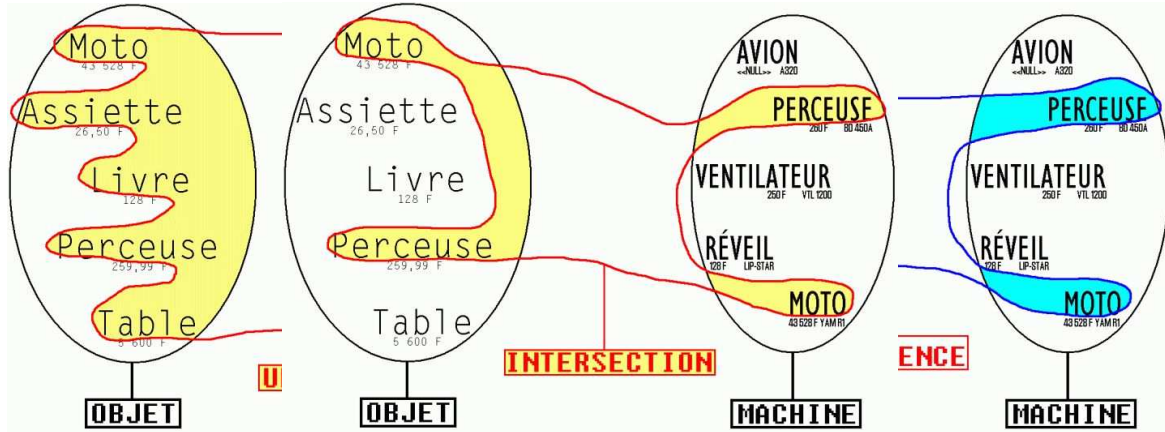
47

Dyadique (ensemblistes) :

Union

Intersection

Différence



Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

# 0 / Algèbre relationnelle

48

## Dyadique (ensemblistes) : Jointure

|        | PRS_ID               | TIT_ID | SEX_ID | PSP_NOM   | PSP_PRENOM |            |
|--------|----------------------|--------|--------|-----------|------------|------------|
|        | 91                   | 91     | 1      | 1         | COUASSE    | françois   |
|        | 92                   | 92     | 1      | 1         | JOLY       | CHRISTOPHE |
|        | 93                   | 93     | 1      | 1         | BENATT...  | PIERRE     |
|        | 94                   | 94     | 2      | 1         | PARIS      | MICHÈLE    |
|        | ...                  | ...    | ...    | ...       | LEAL       | JANY       |
| ADR_CP | ADR_VILLE            |        |        |           |            |            |
| 78000  | VERSAILLES           |        |        | BENATT... | BERNARD    |            |
| 84524  | VERGNOLLES CEDEX 452 |        |        | AIACH     | ALEXANDRE  |            |
| 11254  | MONTMAIZIN           |        |        | GAL       | FABRICE    |            |
| 75015  | PARIS                |        |        | CHAMB...  | EDITH      |            |
| 13002  | MARSEILLE            |        |        | DUQUE...  | JACQUES    |            |
| 94152  | BONNEUIL CEDEX       |        |        | CHEVAL... | yanis      |            |
| 75012  | PARIS                |        |        | JEAN      | HENRI      |            |
| 75012  | PARIS                |        |        | DUFOUR    | ALBAN      |            |
| 75014  | PARIS                |        |        | MARTIN    | MARC       |            |

Consiste à rapprocher différentes relation par association de valeurs

Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

**ATTENTION** : une jointure n'est pas :

- Forcément naturelle (entre clef primaire et clef étrangère)
- Obligatoirement par égalité (équi-jointure)

# 0 / Algèbre relationnelle

49

Langage SQL :

N'est pas purement relationnel, notamment :

- NULL possible,
- Clef pas obligatoire

Autres problématique :

- Atomicité difficile à contrôler « naturellement »
- Utilise parfois un « ordre » par défaut dans la syntaxe des commandes
- Étendu aux objets (norme SQL:1999)

Normalisé en 1986 (ANSI) puis par l'ISO...

Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

La langage SQL a été inventé par Don Chamberlain et Raymond Boyce au début des années 70 sous le nom de SEQUEL (Structured English QUERY Language) et rebaptisé SQL (Structurel Query Language).

# 0 / Algèbre relationnelle

50

Langage SQL - Ordre logique d'exécution...

| Ordre | Clause   |
|-------|----------|
| 5     | SELECT   |
| 1     | FROM     |
| 2     | WHERE    |
| 3     | GROUP BY |
| 4     | HAVING   |
| 6     | ORDER BY |

## ATTENTION :

le moteur relationnel n'a aucune obligation de traiter dans ce même ordre et, pour des raisons d'optimisation, réordonnera et regroupera certaines opérations pour simplifier et aller plus vite !



Par exemple filtrer avant de joindre ou réduire certains prédicat.

# 0 / Algèbre relationnelle

51

## Langage SQL – composantes

| Partie     | Description   | Exemple                          |
|------------|---|----------------------------------|
| Commande   | Instruction complète et autonome effectuant une tâche particulière dans le SGBDR  | SELECT, DECLARE, SET...          |
| Requête    | Catégorie d'instruction de nature ensembliste portant sur le traitement des données, la structuration des objets ou l'accès aux données | SELECT, UPDATE, CREATE, GRANT... |
| Clause     | Partie d'une requête effectuant une tâche spécifique  | ORDER BY, WHERE...               |
| Expression | À l'intérieur d'une clause, articulation de différents éléments calculés entre eux afin d'obtenir un résultat                           | Colonne1 + 3                     |
| Prédicat   | Expression renvoyant un valeur booléenne (TRUE, FALSE, UNKNOWN)   | Colonne1 = 3                     |

Optimisation SQL Server - CH 01 - introduction

V3.0 10/07/2017

Ne pas oublier que la logique SQL est ternaire !



# SQL SERVER 2012/2017 ADMINISTRATION

Frédéric Brouard (SQLpro) [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
Richard Gaillard - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)

Version 3.001  
Date 10/07/2017



1

## Installation & configuration

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

# 1 – Installation & configuration

3



- Historique
- Versions & éditions
- Installation
- Compte de service
- Architecture
- Configuration
- Bases système
- Migration
- Outils



# 1 / historique

4

- 1974 : projet Ingres (*Stonebraker et Wong*)
- 1984 : projet Sybase héritant du moteur Ingres
- 1986 : premier SGBDR C/S commercialisé
- 1987 : trigger (*premier code côté SGBDR*)
- 1988 : accord Sybase Microsoft pour portage Win.
- 1989 : sortie de SQL Server Unix et OS/2
- 1993 : rupture Sybase MS. Sortie v. 4.21 Windows
- 1993 : première version "full Windows" : 6 puis 6.5

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Sybase SQL Server a été le premier SGBDR à fonctionner en mode client/serveur. Les produits concurrents fonctionnaient tous sur gros système à l'époque (IBM System R et Oracle).

Sybase SQL Server a été le premier SGBDR à intégrer du code côté serveur en inventant la notion de déclencheur (trigger).

Les procédures stockées sont arrivées en version 6.

# 1 / historique

5

- 1998 : Sybase débaptise SQL Server pour ASE
- 1999 : version 7. Réécriture moteur relationnel, rajoute la BI
- 2000 : version 2000
- 2003 : version 2000 64 bits
- 2005 : version 2005. Refonte du moteur relationnel
- 2008, 2010 : version 2008 et 2008 R2
- 2010 : SQL Azure (version en « cloud » MS)
- 2012 : version 2012
- 2014 : version 2014, refonte de l'optimiseur « In Memory »
- 2016 : version 2016, « Stretch table », table temporelles
- 2017 : version Linux, tables de graphes

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

ASE : Adaptive Server Enterprise.

- La version 7 incluait les UDF scalaires.
- La version 2000 incluait les UDF tables et la BI.
- La version 2005 y ajoute le partitionnement, un "MVCC", le "MARS", les Web Services, le support de XML, les routines en .net (SQL CLR), les CTE, le varDecimal et une sécurité à très haut niveau (incluant le chiffrement des données) en sus du mirroring (haute disponibilité).
- La version 2008 y ajoute le spatial (SIG), le FILESTREAM, intègre au moteur le "full-text" (déjà présent dans les versions précédentes), le chiffrement TDE et l'indexation OnLine.
- La version 2008 R2 y ajoute les référentiels de données (Master Data Management, Master Data Services), la gestion centralisée des serveurs et le DDL OnLine ; PowerPivot et StreamInsight font leur apparition.
- La version 2012 y ajoute AlwaysOn (SQL Server Failover Cluster Instances et Availability Groups), de nouveaux objets et fonctions spatiales, la commande THROW et les index "columnstore".
- La version 2014 y ajoute les tables "in memory", la journalisation asynchrone, l'extension du cache de données (Buffer Pool Extension). En sus Microsoft introduit SQL Azure et la possibilité de sauvegarde vers le cloud MS.
- La version 2016 rend le « in memory » pleinement opérationnel et y ajoute les « stretch tables » (tables en cloud) et les tables temporelles
- La version 2017 est portée sous Linux et y rajoute les tables de graphes

# 1 / perception

6

SQL Server encore perçu comme « petit SGBDR »...

Pourtant...

- Plus fiable en pratique qu'Oracle ! (voir les CVE du NIST)
- Très grosses bases de données : CDiscount, Ventes Privées, fnac.com, Sarenza, Rue du Commerce, ... de 10 à 50 To
- « Mission-critical » : système d'alerte de la BSPP (Pompiers de Paris), système d'alerte du Service de Prévision des Crues du Grand Delta du Rhône (SPCGD)...
- Moins cher (entre 4 et 25 fois moins cher que oracle)
- Plus performant (TPC-E)

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Sur l'image, extrait de « Gartner Magic Quadrant for Operational Database 2015 » :  
*Market image: Although SQL Server is an enterprise-class DBMS, Microsoft continues to struggle to dispel a perception of weakness in this area. Inquiries from Gartner clients demonstrate a continuing perception that SQL Server is not used for mission-critical enterprise wide applications — a view that inhibits wider use of SQL Server as a primary enterprise-class DBMS.*

Pour la fiabilité de SQL Server, voir différentes études, notamment :  
ITIC 2013 - 2014 Database Reliability & Deployment Trends Survey

Pour les VLDB voir aussi :

Pan STARRS : 1,1 Po en 2008 ! (aujourd'hui près de 4 Po sur 8 serveurs)

Hotmail : 500 To en 2010

Banco Itau > 50 To (2010)

Webtrends : 50 To en 2008

Progressive : (une table de plus de 21 milliard de lignes et de 10 To de données.. En 2010)

Vimpelcom (Russie) 15 To

Centipede (plus de 100 bases de plus de 1 To en 2008)

First American Tile : 8 To en 2008

Bwin 8 : To en 2008

Autres bases de plusieurs To en France : Conforama (BI), Mediterranean Shipping Company (MSC), Geodis, Bouygues, Bolloré, Humanis, Essilor, Criteo...

# 1 / versions, éditions

7

- Depuis 2000, version = année de sortie, sauf R2 (2010)
- En interne :

| Commercial | N°  | Interne  | Build                     |
|------------|-----|----------|---------------------------|
| 2000       | 80  | 539      | 8.0.194 à 2039            |
| 2005       | 90  | 611, 612 | 9.0.1399.06 à 9.0.5000.0  |
| 2008       | 100 | 655      | 10.0.1600.22 à 6000.29    |
| 2008 R2    | 100 | 660      | 10.50.1600.1 à 6000.34    |
| 2012       | 110 | 706      | 11.0.2100.60 à 6020.0     |
| 2014       | 120 | 782      | 12.0.2000.8 à 5000.0      |
| 2016       | 130 | 852      | 13.0.1601.5 à 13.0.5026,0 |
| 2017       | 140 | 869      | 14,0,1000,169             |

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Internal : 612 = 2005 avec VARDECIMAL activé. Le n° interne est utilisé pour passer les scripts d'upgrade des objets systèmes lorsqu'une base est restaurée sur un serveur de version supérieur.

Le n° de build, dépend des "service packs" passés. La plage est donnée depuis la RTM, jusqu'au dernier Service Pack connu (en date du 1/8/2015)

## Service Pack

Pour les versions 7 à 2008 : 4 SP

Pour la version 2008 R2 et 2012 : 3 SP

Pour la version 2014 : 2 SP

Pour la version 2016 : 1 SP

Source :

<http://sqlserverbuilds.blogspot.fr/>

# 1 / versions, éditions

8

Compatibilité ascendante des bases :

- ❑ Depuis la version 7 (1999) compatibilité binaire des fichiers.
- ❑ Migration directe :
  - ❑ de 7 à 2000
  - ❑ de 2000 à 2005, 2008, 2008 R2
  - ❑ depuis 2005 jusqu'à n'importe quelle version (de 2008 à 2017)
- ❑ Même chose pour les sauvegardes

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Il n'y a pas de compatibilité descendante des bases, mais vous pouvez utiliser un principe d'export/import ou de réplication (y compris service broker) pour ce faire.

# 1 / versions, éditions

9

- **Édition : limite les fonctionnalités, influe sur le tarif**

| Édition    | Description                                  |
|------------|--|
| CE         | Compact Edition (pour mobile)                |
| Local db   | Embarquée (une seule base)                   |
| Express    | Gratuite (différentes déclinaisons)          |
| Web        | Mode locatif seulement                       |
| Standard   | Pour bases avec heures creuses et non VLDB   |
| Enterprise | Pour VLDB ou sans heures creuses             |
| Azure      | Sur le "cloud"                               |
| Developer  | Equi. À Standard, pour test et développement |

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Ce tableau est valable pour la version 2016

L'édition CE évolue de manière indépendante (smartphone)

Local db : base unique, moteur mono utilisateur, dédié à l'embarqué.

L'édition Express existe en différentes déclinaisons :

- Express
- Express with tools
- Express with advanced services

CE, Local db, Express et Developer sont gratuites.

Bases limitées à 10 Go (hors FILESTREAM/FILETABLE)

L'édition WEB ne peut être que louée auprès d'un hébergeur agréé. Peu coûteuse, elle rentre en concurrence avec MySQL et PostgreSQL.

VLDB : Very Large DataBases (grosses bases, en général de plus de 2 To)

# 1 / versions, éditions

10

- Version, éditions, comment le savoir ?

```
SELECT SERVERPROPERTY('Edition') AS Edition,  
        SERVERPROPERTY('EditionID') AS EditionID,  
        SERVERPROPERTY('EngineEdition') AS EngineEd,  
        SERVERPROPERTY('ProductVersion') AS ProdVer,  
        SERVERPROPERTY('ProductLevel') AS ProductLevel;
```

```
SELECT @@VERSION;
```

```
EXEC xp_msver;
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Mais aussi de l'extérieur, via le registre Windows

Voir aussi :

```
EXEC sp_server_info
```

# 1 / versions, éditions

11

- SQL Server est :
  - Multi base (à partir de l'édition Express)
  - Multi schéma
  - Multi instance
  - 32 et 64 bits (64 bits pour les versions serveur)
- Nom d'objet :
  - [ <serveur> . ][ <base> . ][ <schema> . ] nom\_objet
- Nombre de bases de données :
  - CE et local : 1
  - Express à Enterprise : 32 760

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Le schéma est une unité logique de stockage des objets de la base.

La base est l'unité physique de stockage des données (autonome) assurant le respect de l'intégrité des données.

Une instance de SQL Server est une installation des services SQL Server ayant ses propres réglages et hébergeant ses propres bases (au plus 50 instances sur un même serveur)

Redémarrer une instance consiste à relancer le service considéré.



# 1 / versions, éditions

12

Une édition incorpore plusieurs services... Principalement :

- SQL Server (moteur OLTP)
  - ▣ Agent SQL (du moteur OLTP)
- SSAS (SQL Server Analysis Services) moteur OLAP (BI)
- SSIS (SQL Server Integration Services) ETL intégré
- SSRS (SQL Server Reporting Services) outil de reporting
- SQL Server Full-text Filter Daemon Launcher
  - ▣ Utilisé pour les "i-filter" de l'indexation textuelle.
- SQL Server Browser
  - ▣ Utilisé pour "publier" (diffuser) les services SQL Server
- SQL Server launchpad
  - ▣ Utilisée pour l'OLAP avancé avec le langage R.

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Chaque Edition possède plus ou moins de services :

- Standard, BI et Entreprise les ont tous
- WEB : SQL Server avec Agent + SSRS
- Express : SQL Server sans Agent mais avec SSRS si "Advanced Services"

# 1 / versions, éditions

13

## Éditions Standard et Enterprise (version 2017) :

### □ limites techniques :

| Fonction          | Enterprise | Standard                 |
|-------------------|------------|--------------------------|
| CPU OLTP          | Illimité   | 4 CPU / 24 cœurs         |
| CPU OLAP          | Illimité   | 4 CPU / 24 cœurs         |
| Cache OLTP        | Illimité   | 128 Go                   |
| Cache OLAP        | Illimité   | MOLAP : 64 Go / Tab : 16 |
| Cache SSRS        | Illimité   | 64 Go                    |
| Taille d'une base | 524 Po     | 524 Po                   |
| Buffer pool       | Illimité   | 128 Go                   |
| Columnstore cache | Illimité   | 32 Go (SP 1)             |
| In Memory         | Illimité   | 32 Go (SP 1)             |

7/2017

Ce tableau est valable pour l'édition 2017

524 Po par base :

- soit au plus  $524 \times 32\,760$  (nombre maximum de bases) = 17 166 240 Po par instance
- soit au plus  $17\,166\,240 \times 50$  (limite du nombre d'instances) = 858 312 000 Po par serveur

Autres limites :

- Octets pour une clef primaire /étrangère: 900
- Octets pour index « in memory » : 2500
- Octets pour index non clustered : 1700
- Longueur de ligne : 8060 (sauf si row\_overflow) sans compter les LOBs
- Colonne par index : 32 (y compris pour PK/FK – 16 si clustered et 900 octets)
- Colonnes par table : 1024 (30 000 pour table « wide » - « big table »)
- Index par table : 10 000 (dont 1 clustered, 1 000 relationnels)
- Index « in memory » par table : 8
- Imbrication (récursivité des routines) : 32
- Référence de FK, par table : 253
- Nombre de ligne par table : pas de limite
- Partitions par table/index : 15 000
- Statistiques par table : 30 000
- Table dans un SELECT : pas de limite
- Trigger par table : pas de limite
- Utilisateur connectés : 32 717

# 1 / versions, éditions

14

Ce que l'édition Enterprise permet en plus de la standard :

- Haute disponibilité (depuis 2016 SP1 pour OA en version Standard) :

| Fonction               | Enterprise            | Standard           |
|------------------------|-----------------------|--------------------|
| Cluster                | max                   | 2 nœuds            |
| AlwaysOn               | 8 dont 2 synchrones   | 2 nœuds synchrones |
| Mirroring <sup>1</sup> | synchrone, asynchrone | synchrone          |

- Hardware à chaud : permet le rajout de RAM et CPU à chaud si le serveur physique le permet.

1 : le mirroring est considéré comme obsolète (deprecated) depuis la version 2012 et est remplacé par AlwaysOn

# 1 / versions, éditions

15

Principales fonctionnalités en sus dans l'édition Enterprise :

- Directeur de connexion
- Restauration en ligne
- Indexation en ligne
- DDL en ligne
- **Partitionnement**
- **Compression des données**
- Gouverneur de ressources
- Filestream Multi conteneurs
- Cache optimisé NUMA
- **Tables en mémoire**
- **Journalisation asynchrone**
- Chiffrement TDE
- Gestion de clefs externes
- Réplication avec Oracle
- Réplication de pair à pair
- **Extension du cache**

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

En rouge, ajouté à la version 2016 SP1

DDL (Data Definition Language) : partie du SQL qui permet de créer, modifier ou supprimer les objets (CREATE, ALTER, DROP...).

NUMA (Non Uniform Memory Access) : accès non uniforme à la mémoire pour les systèmes dotés de plusieurs CPU.

TDE (Transparent Data Encryption) : chiffrement du stockage (fichiers) de la base de manière "transparente".

La gestion de clefs externes nécessite un HSM (Hardware Security Module), boîtier électronique sur le réseau qui crée et conserve les clefs de chiffrement.

# 1 / versions, éditions

16

Principales fonctionnalités en sus dans l'édition Enterprise :

- DPV avec mise à jour
- Opérations d'index //
- Vérification d'intégrité //
- Vues indexées automatisées
- Captures Oracle Attunity
- SSIS vers Oracle rapide
- SSIS vers Teradata rapide
- SSIS correspondance floue
- Master Data Services
- Captures CDC
- Jointure en étoile optimisée
- **Indexation ColumnStore**
- PowerPivot (SharePoint)
- Algo. Data Mining avancés
- SSRS : ferme de serveurs
- ...

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

DPV (Distributed Partitioned View) : vues partitionnées distribuées (une vue partitionnée sur différentes instances à travers différents serveurs physiques)

// => en parallèle

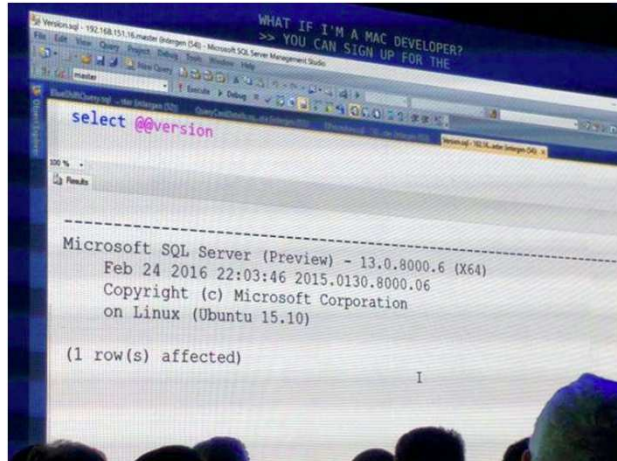
SSIS (SQL Server Integration Services) : ETL intégré

SSRS (SQL Server Reporting Services) : service de reporting intégré

# 1 / OS ?

17

- Les versions Standard, BI et Enterprise s'installent sur un OS Windows Server
  - **Formatage : cluster de 64 Ko**
- Les versions Express et Developer s'installent sur un OS Windows Desktop
- La version Linux s'installe sur différentes éditions et sous docker (2017)



La version 2016 nécessite l'OS Windows 10 pour une installation Desktop.

# 1 / installation

18

Deux modes :

- Par assistant
- En ligne de commande
  - Indispensable pour la version « core »

Pour la ligne de commande, deux modes

- Paramètres à préciser en ligne
- Fichier de configuration (configurationFile.ini)

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Le fichier configurationFile.ini est propre à chaque instance et est construit automatiquement lors de chaque installation.

Il est disponible dans l'arborescence de votre instance à ...\\Setup Bootstrap\\Log\\AAAAMMJJ\_HHMMSS, où AAAAMMJJ\_HHMMSS est la date et heure de début d'installation.

Ce répertoire sert de référence pour la mise en place des patches, CU (Cumulative Update) et SP (Service Pack).

# 1 / installation

19

Sur Windows server core :

- Permet d'installer SQL Server sans la couche graphique
- Nécessite a minima :
  - Windows Server 2008 R2 (core)
  - Windows Server 2012 (standard ou datacenter)

À lire :

<https://msdn.microsoft.com/fr-fr/library/hh231669%28v=sql.120%29.aspx>



# 1 / installation

20

## Mise à jour de version

Il est possible de changer :

- de version pour une version supérieure
  - exemple : 2012 => 2016
- d'édition pour une édition supérieure
  - exemple : Standard => Enterprise

Cela peut se faire par « écrasement de version » :

On installe la nouvelle pour remplacer l'ancienne

- Sinon, par installation à côté et déplacement des bases

Voir la matrice des compatibilité pour la mise à jour vers 2016 :

<https://msdn.microsoft.com/en-us/library/ms143393.aspx>

La migration d'une base dans le cadre d'une installation à côté, prends moins d'une seconde quelque soit la taille de la base (il faut procéder par détachement puis attachement)

Il n'est pas possible de revenir en arrière sur une installation qui à fait une mise à jour supérieure (« upgrade »).

Dans un tel cas vous pouvez scripter tous les objets de la base et réimporter les données ou en amont définir une réplication de données « inverse »

# 1 / installation

21

Points importants...

Choix immuables :

- répertoire d'installation (exe)
- répertoire d'installation des bases système (sauf tempdb)
- collation du serveur (OLTP et OLAP)

La base système tempdb peut être déplacée, mais ce déplacement nécessite un redémarrage du serveur.

Le choix de la collation du serveur entraîne le paramétrage de la collation des bases système (pas de modification ultérieure possible) y compris tempdb.

La collation du serveur est utilisée par défaut pour toute nouvelle base créée (template « model »)

# 1 / installation

22

Collation (norme SQL), attribut/opérateur COLLATE

Permet de définir le comportement des chaînes de caractères au regard :

- De la casse (majuscule, minuscule) : CI ou CS
- Des symboles diacritiques (accents, ligatures...) : AI ou AS
- Des kana types du japonais (katakana et hiragana) : KS
- De la "largeur" de caractère (par exemple 2 et ²) : WS
- Des caractères supplémentaires (certaines langues) : SC
- Du tri relatif à la langue

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Oracle était le seul SGBDR à ne pas offrir le support normatif des collations. Il se reposait sur le concept de NLS (National Language Support) ce qui le rendait inapte à la plupart des applications multilingues.

Oracle a introduit le concept de collation à partir de la version 12c, mais offre un support limité (moins de 100 collations)

CS/CI : Case Sensitive/Insensitive

AS/AI : Accent Sensitive/Insensitive

KS : Kanatype Sensitive (différentiation du katakana et de l'hiragana – à défaut confondu)

WS : Wide Sensitive (différentiation de la "largeur" du caractère en cas de CI)

SC : Supplementary Characters (caractères supplémentaires) pour ajouter certains caractères particuliers pour certains jeux de caractères (UTF8/16). Exemples : emoticons



# 1 / installation

23

Définir la collation :

- À l'installation de l'instance (pour serveur OLTP et OLAP)
- À la création de la base (CREATE DATABASE)
- À la création de toute colonne littérale (table ou vue)
- Dans chaque prédicat manipulant des littéraux

```
CREATE DATABASE MaBase COLLATE French_CI_AS;  
CREATE TABLE T (C CHAR(16) COLLATE French_CS_AS_WS);  
SELECT * FROM T WHERE C = 'toto' COLLATE French_BIN;
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

## Notes sur le classement des caractères en français :

Le classement opère selon l'ordre alphabétique (de « as » à « z ») sans tenir compte de la casse. Pour les lettres diacritiques, la règle est la suivante :

- Pour une même racine de lettre, La première lettre non accentuée prime, mais la seconde lettre accentuée prime !

```
CREATE TABLE #T (mot VARCHAR(32) COLLATE French_CI_AI);
```

```
INSERT INTO #T  
SELECT mot  
FROM (VALUES ('cote'), ('côte'), ('Côte'), ('coté'), ('Coté'), ('côté'), ('Côté'), ('coter'),  
( 'élève'), ('élevé')) AS T (mot);
```

```
SELECT * FROM #T  
ORDER BY mot COLLATE French_CI_AI;
```

**Ainsi, le mot « élève » précède le mot « élevé ».**

**TRES IMPORTANT** : les tables « in-memory » version 2014 ne supportent que les collations binaires (...BIN ou ...BIN2) pour des raisons d'efficacité liées à la méthode d'indexation (hachage). En 2016, vous pouvez choisir d'autres collations.

# 1 / installation

24

Définir la collation :

Pour la base, le choix est immuable :

- Concerne les tables système
- Entraîne à défaut celui de toutes les colonnes

Pour la colonne :

- Modification possible si migration des données

# 1 / installation

25

2 types de collations :

- SQL : relatives aux jeux de caractères (plus performant - 77)
- Windows : indépendantes des jeux de caractères, plus « fonctionnels » (3 879)

Collations Windows "binaires" performantes et sensibles à tout (BIN et BIN2)

Liste des collations (3 955, plus de 100 langues) :

```
SELECT * FROM sys.fn_helpcollations()
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

- Le jeu de caractères est :
- ISO\_1 pour les types "ASCII" (CHAR, VARCHAR) => 1 caractère = 1 octet (avec la page de code relative à la collation)
- UNICODE pour les types "UNICODE" (NCHAR, NVARCHAR) => 1 caractère = 2 octets (UCS-2, Little-Endian)

Sauf exception, les collations SQL "mappent" les jeux de caractères Windows correspondants.

Les collations Windows sont totalement indépendantes des jeux de caractères.

SQL Server n'encode pas les chaînes de caractères stockées. Par exemple il n'est pas possible de stocker de l'UTF8 (certains SGBDR le font croire, mais cela n'est techniquement pas possible, car pas indexable...). Cependant, lors de la restitution des données, l'encodage est assuré à la volée.

BIN = BINaire : l'ordre des caractères est celui de l'UCS2 en "Little-Endian"

BIN2 = BINaire point de code : ordre des caractères modifié pour respect sémantique (à préférer systématiquement)

# 1 / installation

26

Collations, effets secondaires :

- Erreur : *"Impossible de résoudre le conflit de classement entre "X" et "Y" dans l'opération equal to."*

Notamment si la base de production est en collation X et la base tempdb en collation Y

Plusieurs solutions :

- Préciser la collation dans la création des tables temporaires
- Utiliser le SELECT ... INTO pour créer la table temporaire
- Utiliser une base "autonome" (*contained database*)

Vous pouvez préciser la collation à la création d'une table temporaire en utilisant le mot clef `database_default`. Ainsi sera choisit la collation par défaut de la base contextuelle.

Exemple :

```
CREATE TABLE #MaTable (MaColonne VARCHAR(32) COLLATE database_default);
```

# 1 / installation

27

## Codification du nom des collations :

### □ SQL :

SQL\_<nom>\_<cp>\_<reglages>

### □ Windows :

<langue>[\_<version>]\_<reglages>

## Exemples :

SQL\_AltDiction\_CP850\_CI\_AI

SQL\_EBCDIC280\_CP1\_CS\_AS

SQL\_Latin1\_General\_CP437\_BIN

Breton\_100\_CS\_AI\_WS\_SC

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

SQL\_AltDiction\_CP850\_CI\_AI : alphabet Latin1-General, insensible à la casse, insensible aux accents, insensible aux kana types, insensible à la largeur de caractère. Tri SQL Server 57 pour les caractères UNICODE et sur code page 850 pour les caractères ASCII  
Remarquez les 8 collations EBCDIC afin de pouvoir reprendre sans perte des données encodées avec l'horrible jeu de caractère qu'IBM inventa dans les années 70 pour rendre les données hébergées sur ses systèmes (IBM 370 et suivants) difficilement interopérables avec la concurrence ! Dans un tel cas, créez une table « tampon » en collation EBCDIC pour intégrer les données externe venant d'un système IBM puis intégrez les dans vos tables de production. Aucune perte de caractère notamment diacritiques.



# 1 / installation

28

## Instances :

- par défaut (une seule)
  - nom du serveur physique Windows
  - Port TCP 1433 par défaut
- nommée : nom\_serveur\nom\_instance
  - Port dynamique attribué par SQL Browser (UDP 1434)
  - Valable pour tous les composants (SQL, SSAS, SSRS, ...)
- Chaque instance est totalement indépendante
- Concurrence pour les ressources !

Pour le cas du multi instance, il est important de partager les ressources, notamment pour la RAM.

# 1 / installation

29

Politique des mises à jour SQL Server :

- Patch : ne jamais passer sans avis de la hot line MS, sauf niveau "security"
- CU (Cumulative Update) :
  - (< version 2016) ne pas passer sauf si invite MS dans un article
  - (>= version 2016) passer au plus vite
- SP (Service Pack) : passer au plus vite

La plupart ne nécessite pas le redémarrage.

Moindre nécessité pour Windows, sauf niveau "security".

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Pensez aussi à passer les SP de Windows, même si SQL Server utilisant très peu Windows.

La politique des « patch » à changé avec la version 2016; les CU sont désormais certifiés.

# 1 / installation

30

Redémarrage de l'instance SQL Server :

- Vidage du cache => lectures disques, établissement systématique de nouveau plan...
- Suppression des statistiques de fonctionnement => post diagnostic impossible, perte des métriques des DMV et XE...

**Ne jamais redémarrer une instance SQL Server sauf absolue nécessité.**

Toutes les opérations peuvent se faire à chaud !

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Une pratique hélas autrefois courante était de redémarrer les serveurs Windows. Ceci est une aberration, même si parfois cela donne l'impression de résoudre certains problèmes !

DMV (Dynamic Management View) : vue de gestion dynamique  
XE (eXtended Event) : évènement étendu

# 1 / installation

31

Renommer un "serveur" SQL Server :

- Préalablement, renommer le serveur au niveau Windows
- Dans le serveur SQL :
  - Identifier le serveur courant via @@SERVER
  - Voir si c'est différent de SERVERPROPERTY('InstanceName')
  - Supprimer le serveur local de la table des serveurs (sys.servers, server\_id = 0) via la procédure sp\_dropserver
  - Rajouter le serveur local via la procédure sp\_addserver avec l'option @local = 'local'
- Redémarrer l'instance

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

**ATTENTION** : le nom du serveur peut être inscrit en clair dans certaines routines ou objets (vues par exemple) que les développeurs ont créés !

# 1 / comptes de service

32

## Sécurité...

Chaque service tourne sous un compte Windows de service

- Droits minimaux
- Créé par défaut lors de l'installation en compte local
- Peut utiliser un compte de domaine (notamment si AD)

## **Respectez le principe des droits minimaux**

Élargir si nécessité (par exemple destination des sauvegardes)

Administration SQL Server - CH 01 - installation & configuration

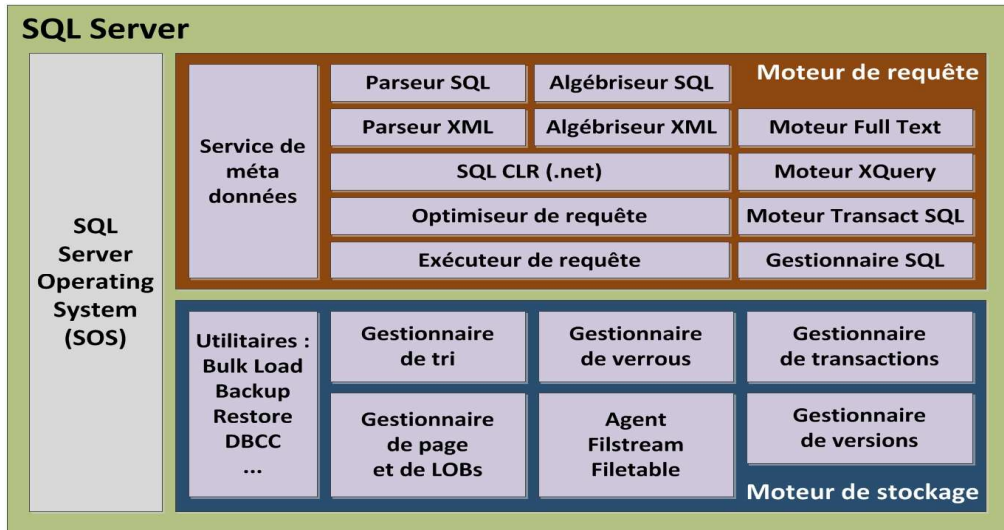
V3.0 10/07/2017

Pour la création des comptes de domaine (à effectuer préalablement à l'installation) lire  
:  
<https://msdn.microsoft.com/en-us/library/ms143504%28v=sql.120%29.aspx>

# 1 / architecture

33

## Structure interne du moteur relationnel :



Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Notez la complète dissociation du moteur de requête (logique) et du moteur de stockage (physique).

Notez que le tri est une opération physique !

Notez la présence de SQL OS qui assure la gestion des threads (CPU/core), du cache (RAM) et des opérations physiques de lecture/écriture (disque).

# 1 / configuration

34

Niveau instance :

□ Lire, configurer :

```
EXEC sp_configure;
```

□ Voir :

```
SELECT * FROM sys.configurations;
```

Niveau base :

```
SELECT * FROM sys.databases;
```

```
ALTER DATABASE ...
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

sp\_configure permet de lire les paramètres de configuration dans la limite de "show advanced options" qui, avec la "run value" à 1, permet de tout voir.

Appliquer une configuration se fait en 2 temps :

- Modifier le ou les paramètres (config\_value) par sp\_configure (un par un)
  - Le nouveau paramétrage n'est pas encore appliqué mais apparaît dans la colonne « config\_value »
- Appliquer les paramètres modifiés en exécution (commande RECONFIGURE)
  - Le nouveau paramétrage est appliqué et apparaît dans la colonne « run\_value »

# 1 / configuration

35

Configuration minimale de l'instance :

- Limitation de la RAM :
  - ▣ *max server memory* : RAM - delta (en Mo)
- Limitation du parallélisme
  - ▣ *cost threshold for parallelism* : entre 12 et 50
  - ▣ *max degree of parallelism* : entre 1 et 8, sauf DW
- Activation de l'optimisation des requêtes "ad hoc"
  - ▣ *optimize for ad hoc workloads* : 1
- Activation des sauvegardes compressées
  - ▣ *Backup compression default* : 1

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Toujours limiter le cache utilisé par votre instance (**max server memory**) en fonction de la RAM disponible :

- 8 à 16 Go : RAM serveur – 2 Go
- 16 à 32 Go : RAM serveur – 3 Go
- 32 à 128 Go : RAM serveur – 4 Go
- 128 à 512 Go : RAM serveur – 6 Go
- 512 et + : RAM serveur - 8 Go

Descendre encore plus bas si d'autres services SQL sont actifs (ou que d'autres services non SQL ou applications sont actifs, ce qui n'est pas recommandé)

Parallélisme :

- **cost threshold for parallelism** (seuil de coût du parallélisme) : seuil de coût de requête au delà duquel SQL Server ré étudie le plan de requête sous l'angle du parallélisme.
- **max degree of parallelism** : limite par défaut du nombre de threads en parallèle pour une même requête (contournable par l'option MAXDOP au niveau de chaque requête).
- **optimize for ad hoc workloads** : minimise la mise en cache des requêtes "ad hoc" lors du premier appel.
- **backup compression default** : compresses toutes les sauvegardes de tous type, afin de gagner en place et temps (même lors de la restauration).

Requête "ad hoc" : requête directe non contenue dans une routine (UDF, déclencheur, procédure stockée).

Certains paramètres de configuration du serveur sont aussi disponibles, base par base.



# 1 / configuration

36

## Paramétrage des bases :

- ❑ DATE\_CORRELATION\_OPTIMIZATION { ON | OFF }
  - ❑ Rajoute des statistiques de corrélation sur les paires de tables jointes par une contrainte d'IR sur type DATETIME
- ❑ PARAMETERIZATION { SIMPLE | FORCED }
  - ❑ Systématise la paramétrisation des valeurs passées dans les requêtes
- ❑ TORN\_PAGE\_DETECTION { ON | OFF }
  - ❑ Active la détection des pages abimées (*obsolète*), utilisez :
- ❑ PAGE\_VERIFY { CHECKSUM | TORN\_PAGE\_DETECTION | NONE }

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

DATE\_CORRELATION\_OPTIMIZATION, à lire :

<https://technet.microsoft.com/fr-fr/library/ms177416%28v=SQL.105%29.aspx>

TORN\_PAGE\_DETECTION est obsolète. Préférez PAGE\_VERIFY

Pour les bases OLTP, PAGE\_VERIFY CHECKSUM est préférable.

Pour les bases OLAP, PAGE\_VERIFY TORN\_PAGE\_DETECTION ou NONE peut être utilisé sauf alimentation DW en différentiel.

Certains paramètre de configuration du serveur sont aussi disponible, base par base.

# 1 / configuration

37

Paramétrage de la base :

- AUTO\_CLOSE : fermeture automatique de la base
- AUTO\_SHRINK : réduction automatique des fichiers

**NE JAMAIS METTRE À ON !<sup>1</sup>**

Statistiques ( à mettre à ON ) :

- AUTO\_CREATE\_STATISTICS { OFF | ON [ ( INCREMENTAL = { ON | OFF } ) ] }
- AUTO\_UPDATE\_STATISTICS { ON | OFF }
  - AUTO\_UPDATE\_STATISTICS\_ASYNC { ON | OFF }

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

1 : ces options ne doivent concerner que les bases de type CE, Express ou "localDB"

AUTO\_CLOSE : ferme la base dès qu'il n'y a plus de connexions : vidage du cache, suppression des statistiques d'exécution... contrairement à ce qu'affirme Microsoft !

AUTO\_SHRINK : réduit automatiquement la taille des fichiers de la base si 25 % de place libre.

Ces deux options posent de problèmes majeurs de performance.

AUTO\_...STATISTICS : (concerne les statistiques de l'optimiseur destinées à établir un bon plan de requête)

CREATE : permet de créer des statistiques sur colonnes en dehors des statistiques d'index

UPDATE : met à jour les statistiques jugées obsolètes

ASYNC : mode asynchrone de mise à jour des statistiques, valable en édition Enterprise (nécessite l'activation préalable de AUTO\_UPDATE\_STATISTICS)

# 1 / configuration

38

Pas assez de cache ?

- Optez pour l'extension du « buffer pool »...
  - Ce qui rajoute un cache de second niveau.
- Optez pour
  - des disques SSD
  - des flashdisks sur carte PCI (fusion IO par exemple)
- Ratio RAM : 4 à 8 (32 maximum)

Syntaxe :

```
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION
    ON ( FILENAME = 'nom_emplacement_fichier' ,
        SIZE = <taille> )
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

La taille du buffer pool se donne en KB, MB ou GB au choix

DMV spécifique : sys.dm\_os\_buffer\_pool\_extension\_configuration (Transact-SQL)

Voir aussi : sys.dm\_os\_buffer\_descriptors (Transact-SQL)

# 1 / bases système

39

4 visibles :

- master : liste des bases, connexions, messages d'erreur
- model : "template" pour création des bases
- msdb : gestion des travaux SQL Server
- tempdb : objets temporaires

1 invisible :

- mssqlsystemresource : routines système

Autres bases possibles, installables à la demande :

- distribution (réplication), semanticdb (rech. sémantiques)

La base *distribution* (dont le nom peut différer) est utilisée pour les travaux de réplication des données entre les bases.

La base *semanticdb* est utilisée pour la recherche sémantique.

# 1 / bases système

40

Base master, contient :

- La sécurité au niveau serveur (comptes de connexion...)
- La liste des bases de données (vue sys.databases)
- Les messages d'erreur (vue sys.messages)
- Les réglages du serveur (vue sys.configurations)

À sauvegarder !

- Vous pouvez y ajouter vos propres tables, vues ou routines
- Vous pouvez promouvoir une proc. au niveau "système"
  - elle sera visible et exécutable dans toutes les bases

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Pour voir les comptes de connexion et plus généralement la sécurité au niveau serveur :

- sys.server\_principals (objets de sécurité de niveau serveur)
- sys.server\_permissions (privilèges attribués)
- sys.server\_role\_members (rôles accordés)
- sys.system\_components\_surface\_area\_configuration (éléments de la surface d'exposition du serveur)

Pour promouvoir une procédure au niveau système :

- Elle doit être créée dans le schéma dbo
- Son nom doit commencer par "sp\_"
- Vous devez la promouvoir à l'aide de la procédure sp\_MS\_marksystemobject

# 1 / bases système

41

Base model :

- Sert de modèle pour la création de toute nouvelle base

**ATTENTION** : aucune connexion dans cette base lors d'un  
**CREATE DATABASE**

- Vous pouvez y ajouter vos objets (par exemple une  
procédure générique de maintenance)

**Ne pas sauvegarder sauf si ajout(s)**

# 1 / bases système

42

Base msdb :

- Contient les travaux de l'Agent SQL
- Historique des sauvegardes
- Historique d'exécution des travaux

À sauvegarder

# 1 / bases système

43

## Base tempdb :

- Contient les objets temporaires explicites et implicites.
- Utilisé implicitement par :
  - Tris (ORDER BY) et groupages (GROUP BY) importants
  - Sous-requêtes volumineuses
  - Variables table
  - Niveau d'isolation SNAPSHOT
  - Pseudo tables INSERTED et DELETED (trigger et clause OUTPUT)
  - Curseurs si décorrés
  - ...

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Pas de sauvegarde possible. Recrée à chaque démarrage de l'instance.  
Peut être très volumineuse et doit être gérée au niveau du stockage comme une base de production ne contenant qu'un seul "storage" (FileGroup PRIMARY).



# 1 / bases système

44

Base mssqlsystemresource (communément appelée "ressource") :

- Invisible, inatteignable
- Contient les objets du schéma sys
- Plus de 2000 objets
- is\_ms\_shipped = 1 (en principe)
- Visibles dans sys.system\_objects

Version, avec :

```
SELECT SERVERPROPERTY('ResourceVersion'),  
        SERVERPROPERTY('ResourceLastUpdateDateTime')
```

Les objets de cette base sont communs à toutes les bases.

Elle est liée à la base master dont elle constitue en quelque sorte une « sous base ». La sauvegarde de master entraîne la sauvegarde de mssqlsystemresource

# 1 / bases système

45

Base mssqlsystemresource, voir le contenu :

```
SELECT name FROM sys.all_objects  
WHERE is_ms_shipped = 1;
```

```
SELECT name FROM sys.system_objects;
```

Code de la routine obtenu par

- sys.system\_sql\_modules
- ou avec OBJECT\_DEFINITION

La vue sys.all\_objects présente les objets d'une base associés aux objets communs de master et mssqlsystemresource

# 1 / migration

46

SQL Server peut migrer directement dans la limite de 2 versions majeures antérieures.

Différents scénarios de migration de version/édition :

- Avec arrêt de service :
  - Détachement et rattachement
  - Sauvegarde et restauration
  - Sauvegarde et réplication
  - Script, recréation et import (complexe, lourd et coûteux)

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Pour les bases anciennes (version 2000, voire 2005), il est préférable de passer par le conseiller de mise à niveau (SQL Server Upgrade Advisor) :

<http://www.microsoft.com/fr-fr/download/details.aspx?id=11455>

Celui-ci fournit un diagnostic de la migration :

- Actions préventives à la migration (par exemple modifier les anciennes jointures externe propriétaires \*=...)
- Actions à prévoir au moment de la migration (par exemple recalculer les statistiques)
- Actions à prévoir après migration et avant production (par exemple modifier les appels aux anciennes tables système)

<https://technet.microsoft.com/fr-fr/library/ms144256%28v=sql.100%29.aspx>

# 1 / migration

47

## □ Avec arrêt de service :

| Méthode         | Temps indispo               | Temps mise en œuvre | Facilité         |
|-----------------|-----------------------------|---------------------|------------------|
| Détach./attach. | Minime à Moyen <sup>1</sup> | Très rapide         | Très facile      |
| Sauve./rest.    | Moyen à long <sup>2</sup>   | Très rapide         | Facile           |
| Sauve./replic.  | Minime à moyen <sup>3</sup> | Moyen               | Moyen            |
| Script...       | Important                   | Très important      | Moyen à complexe |

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

1 : minime si la migration se fait d'une instance à l'autre sur la même machine (au plus quelques secondes)

2 : pour diminuer le temps d'indisponibilité, faire des sauvegardes transactionnelles les plus rapprochées possible

3 : pour diminuer le temps d'indisponibilité, diminuer la latence de la réplication. Ce mode permet aussi le retour arrière

# 1 / migration

48

- Par détachement et rattachement, exemple :

```
EXEC master.dbo.sp_detach_db N'MaBase'  
  
CREATE DATABASE MaBase ON  
(NAME='MaBase_data',  
  FILENAME=N'C:\SQL\DATA\MaBase.mdf'),  
(NAME='MaBase-Tran',  
  FILENAME=N'C:\SQL\LOG\MaBase_log.ldf')  
FOR ATTACH;
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Les anciennes procédures stockées `sp_attach_db` et `sp_attach_single_file_db` sont considérées comme obsolètes.

Il est possible de rattacher une base sans son journal. Cela permet d'aller encore plus vite et de reconstruire un journal de transaction « vierge » ayant les dimensions de celui d'origine. Exemple :

```
CREATE DATABASE MaBase ON  
(NAME='MaBase_data',  
  FILENAME=N'C:\SQL\DATA\MaBase.mdf')  
FOR ATTACH_REBUILD_LOG;
```

# 1 / migration

49

Sans arrêt de service :

- Par mirroring
- Via AlwaysOn

| Méthode   | Temps mise en œuvre | Facilité     |
|-----------|---------------------|--------------|
| Mirroring | Moyen               | Assez facile |
| AlwaysOn  | Moyen               | Moyen        |

# 1 / outils

50

3 outils essentiels :

- Gestionnaire de configuration (réglage des services SQL)
- SSMS (boîte à outils du développeur BD et du DBA)
- SQLcmd.exe (SQL Server en ligne de commande)

De nombreux autres :

- Pré-installés (Profiler SQL, Database Tuning Advisor...)
- Téléchargeables depuis site MS (exemple SQLio)
- Téléchargeables depuis site communautaire  
CodePlex/GitHub (exemple Dbdiff)

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Pour SQL IO :

<http://www.microsoft.com/en-us/download/details.aspx?id=20163>

Pour Dbdiff :

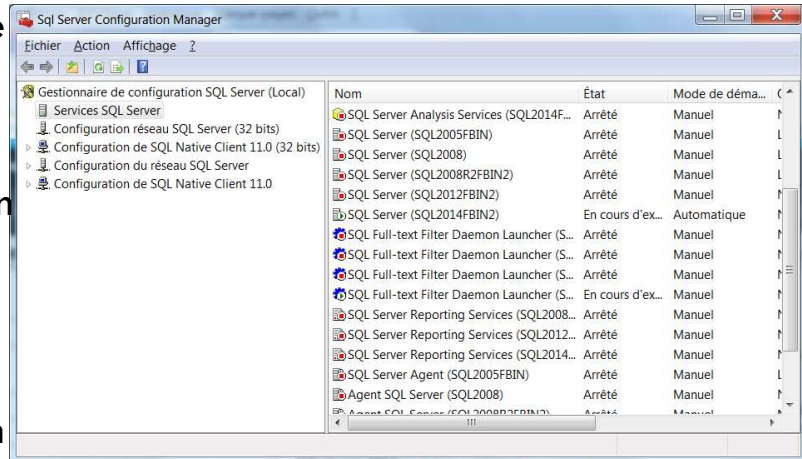
<https://dbdiff.codeplex.com/>

# 1 / outils

51

## Gestionnaire de configuration :

- Administration des services SQL Server
- Configuration des protocoles d'accès



Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

A l'exception des commandes *net start* et *net stop*, il est important de passer par cette interface pour toute autre manipulation, car certaines options activent ou désactivent des ACLs (Access Control List).

Par exemple, pour changer le port TCP d'un service, allez dans l'onglet "Configuration du réseau SQL Server", développez-le et choisissez "Protocoles pour <nom de votre instance>". Dans la liste, cliquez double sur "TCP-IP". Une boîte de dialogue apparaît. Cliquez sur l'onglet "Adresses IP". Modifiez les valeurs de port des adresses IP concernées.

Par exemple, vous pouvez définir un alias de serveur ("surnom" donné à une instance) : pour ce faire, allez dans "Configuration de SQL Native Client" puis cliquez avec le bouton droit sur "Alias" et sélectionnez "Nouvel alias". Dans la zone "Nom de l'alias", tapez le nom de l'alias. Il s'agit du nom que les applications clientes utiliseront pour se connecter. Dans la zone "Serveur", indiquez le nom ou l'adresse IP d'un serveur. Dans la zone "Protocole", sélectionnez le protocole utilisé pour cet alias.



# 1 / outils

52

## SQL Server Management Studio (SSMS)

C'est l'outil à tout faire pour le :

- Concepteur (création des objets)
- Développeur (vues, routines...)
- Administrateur (gestion, maintenance...)

Permet d'appeler des outils complémentaires :

- Profiler SQL
- Assistant de paramétrage du moteur de BD
- ...

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Vous pouvez rajouter des plug-ins dans SSMS, pour ajouter des fonctionnalités.

Par exemple :

- Poor Man's T-SQL Formatter

<http://architectshack.com/PoorMansTSqlFormatter.ashx>

- SSMS Tools

<http://blog.netapsys.fr/plugin-ssms-tools-pour-sql-server-management-studio/>

- SCC Subversion

[http://zeusedit.com/agent/ssms/ms\\_ssms.html](http://zeusedit.com/agent/ssms/ms_ssms.html)

- SSMS Tools Pack

<http://www.ssmstoolspack.com/>

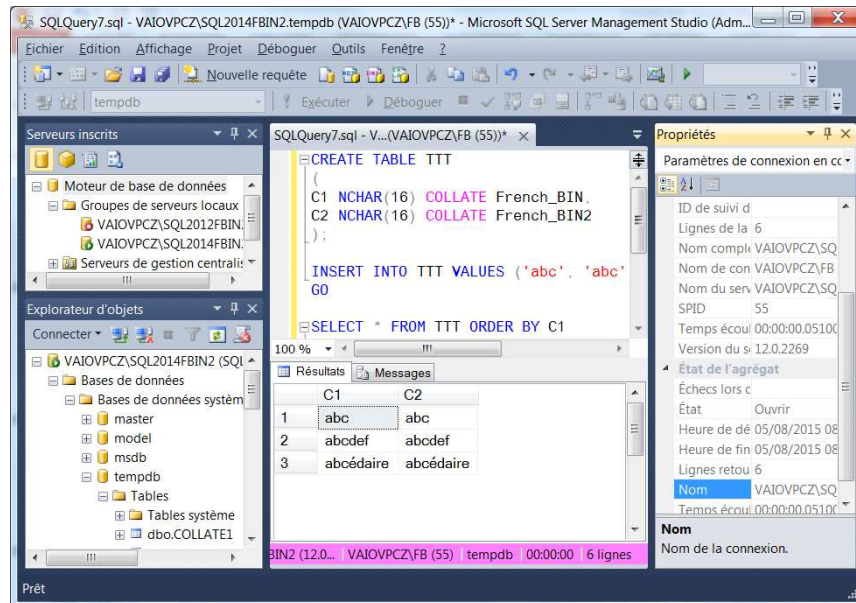
- SSMSBoost

<http://www.ssmsboost.com/>

...

# 1 / outils (SSMS)

53



Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

# 1 / outils

54

## SQL Server Management Studio (SSMS)

Fenêtres "dockables" :

- Serveur inscrits (ajoutez-les tous !)
- Explorateur d'objets (arborescence des instances)
- Détails de l'explorateur d'objets (liste relative)
- Propriétés (détails contextuels)
- Fenêtre de requêtes (éditeur texte avancé)
- ...

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Serveurs Inscrits : ajoutez autant d'instances que vous voulez dans cette fenêtre. Regroupez-les. Vous pouvez donner une couleur différente pour chacune de vos instances. Vous pouvez créer un serveur centralisé pour lancer des commandes sur plusieurs serveurs simultanément...

Explorateur d'objets (raccourci F8) : permet de scruter l'ensemble de la hiérarchie des objets de chaque instance à laquelle vous vous connectez.

Détails de l'explorateur d'objets (raccourci F7) : liste les éléments en fonction de la position du curseur dans l'arborescence de l'explorateur d'objets.

Propriétés (raccourci F4) : donne des détails fins de manière contextuelle. Vous pouvez vous en passer dans les premiers temps. Sert très occasionnellement.

Fenêtre de requêtes : sert à écrire vos requêtes SQL.

# 1 / outils

55

## SQL Server Management Studio (SSMS)

- Barres de menu :
  - Évolue en fonction du contexte
- Barre d'état des fenêtres d'éditeur, indique :
  - L'état de la connexion
  - Le serveur visé
  - Le client qui l'utilise
  - La base contextuelle
  - La durée d'exécution
  - Le nombre de lignes manipulées

# 1 / outils

56

## SQL Server Management Studio (SSMS)

- Onglet des fenêtres d'éditeur :
  - ▣ Nom du fichier/requête + info de connexion (bulle)
  - ▣ Clic droit pour gérer (par exemple "fermer tout sauf cet élément")
- Espace de saisie des fenêtres d'éditeur :
  - ▣ Numérotation des lignes possible
  - ▣ Indication des lignes "changées"
  - ▣ Réduction ordre SQL (+)

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Vous pouvez utiliser des "plugins" pour rajouter des fonctionnalités à l'éditeur.

Exemple :

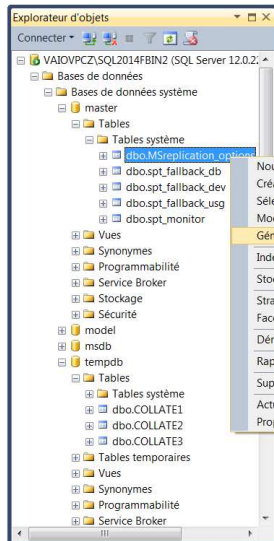
<https://ssmsaddin.codeplex.com/>

Propose un "formateur" de requêtes SQL et bien d'autres gadgets !

# 1 / outils

57

## SQL Server Management Studio (SSMS) – Explorateur :



Clic droit très utile :

- Ouvre un menu contextuel permettant de :
  - Obtenir des détails
  - Scripter les objets
  - Visionner des rapports

Vue arborescente complète :

- Des objets serveur
- Des bases utilisateur
- Des bases système
- De l'Agent SQL

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

# 1 / outils

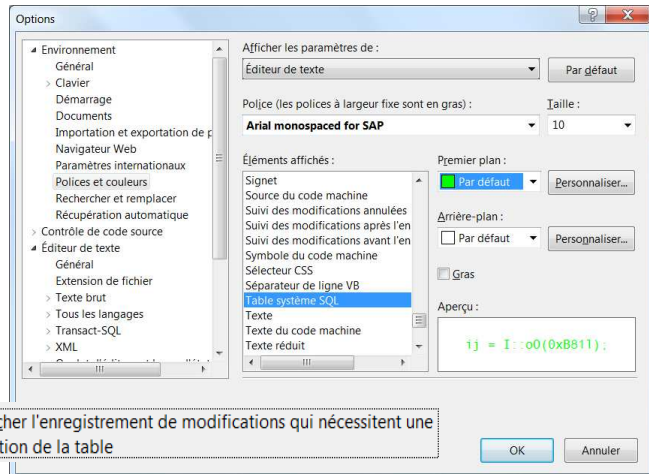
58

## SQL Server Management Studio (SSMS)

### □ Hautement paramétrable (menu : outil/options)

Pensez à :

- Modifier la couleur de police (illisible) des "tables système"
- Autoriser l'enregistrement de modification qui nécessite une recréation de la table (utilisateurs avertis !)



Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

**Attention** : "Autoriser l'enregistrement de modifications qui nécessitent une recréation de la table" signifie qu'en cas de modification importante, un script SQL sera exécuté qui détruit la table et la recrée. Bien entendu les données sont stockées temporairement dans une table annexe.

Ceci peut avoir deux inconvénients :

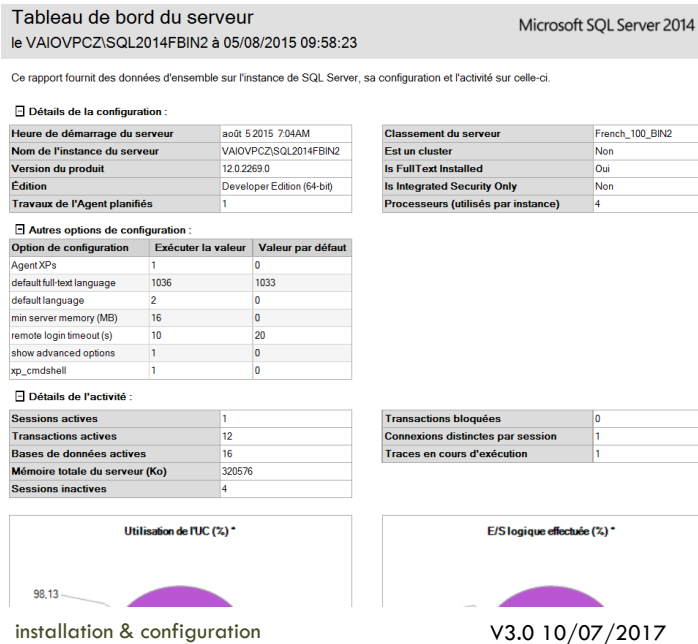
La durée du traitement : pour de gros volumes, la durée peut être très longue

En cas de problème, le ROLLBACK peut prendre du temps, voire saturer le journal de transactions

# 1 / outils (SSMS / Rapport)

59

## Exemple de rapport au niveau serveur : tableau de bord du serveur



Il existe des rapports prédéfinis (standard) aux niveaux suivants :

- Serveur
- Base (sur chaque nom de base)
- Sécurité/connexion
- Gestion
- Gestion / collecte de données
- Agent SQL Server

Vous pouvez rajouter vos propres rapports en les créant préalablement via SSRS (SQL Server Reporting Services)

<https://msdn.microsoft.com/en-us/library/bb153684.aspx>

Un collection de rapports complémentaires est disponible sur le site codeplex :

<https://ssmsinforeports.codeplex.com/>



# 1 / outils

60

## SQLcmd.exe

- Exécutable en ligne de commande pour lancer des requêtes
- Nécessite une authentification au serveur :
  - -E : authentification Windows intégrée par le biais de votre compte de connexion système
  - -U<nom Utilisateur> -P<mot de Passe> : authentification SQL
  - -S<nom Serveur> : nom de l'instance SQL
- Éventuellement :
  - -d<nom Database>

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Attention, les "commutateurs" sont sensibles à la casse !

# 1 / outils

61

## SQLcmd.exe

- Lancement d'une requête :
  - -q "requête SQL" : tamponne une requête (GO pour la lancer)
  - -Q "requête SQL" : lance une requête
- Pour sortir :
  - exit ou quit
- -A : mode DAC (Dedicated Administrative Connector)
- Permet d'utiliser des variables

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Pour de plus amples détails :

<https://msdn.microsoft.com/fr-fr/library/ms162773.aspx>

Le DAC est une connexion d'urgence libérant des ressources pour son exécution, destinée à des opérations en cas de "blocage" du serveur. Cette possibilité n'est active que localement.

# 1 / métadonnées

62

Métadonnées (uniquement des vues) :

- Logiques, norme SQL :
    - dans le schéma SQL **INFORMATION\_SCHEMA**
  - Physiques, propres à SQL Server :
    - Dans le schéma SQL **sys**
- Deux niveaux :
- Statique (objets, fichiers, index...)
  - Dynamique (DMV - nom commençant par dm\_)

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Bien que ces vues reposent sur des tables système internes, l'accès aux tables système est interdit.

DMV : Data Management View. Données dynamiques d'exécution de SQL Server, collectées et présentées sous forme de vues.

Équivalent des vues V\$ d'Oracle (GV\$ => sys.dm\_db\_...)

**Les vues d'information de schéma (INFORMATION\_SCHEMA. ...) doivent être utilisées par les développeurs.**

**Ces derniers ne doivent jamais utiliser les vues système MS SQL Server :**

- Leur accès (privilèges SQL) n'est pas garanti, alors que c'est le cas des vues INFORMATION\_SCHEMA de visibilité publique et filtrées en fonction des privilèges
- Microsoft se réserve la possibilité de modifier la structure et le contenu de ces objets à tout moment, pouvant les rendre obsolètes ou présentant des données incorrectes ou incomplètes.

# 1 / métadonnées

63

Métadonnées normalisées (INFORMATION\_SCHEMA) :

| Table_name               | Contenu                                     |
|--------------------------|---|
| .SCHEMATA                | Schémas SQL                                 |
| .TABLES                  | Tables et vues                              |
| .VIEWS                   | Vues, avec leur définition                  |
| .COLUMNS                 | Colonnes des tables et vues                 |
| .ROUTINES                | Fonctions et procédures, définition incluse |
| .TABLE_CONSTRAINTS       | Contraintes de table                        |
| .KEY_COLUMN_USAGE        | Colonnes utilisées par clefs (PK, UQ, FK)   |
| .REFERENTIAL_CONSTRAINTS | Contraintes d'intégrité référentielle       |
| .CHECK_CONSTRAINTS       | Contraintes de validation                   |
| .CONSTRAINT_COLUMN_USAGE | Colonnes utilisées par des contraintes      |

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Un extrait... 21 en tout

# 1 / métadonnées

64

Métadonnées normalisées (INFORMATION\_SCHEMA), exemple :

```
SELECT T.TABLE_SCHEMA, T.TABLE_NAME, T.TABLE_TYPE,
       C.COLUMN_NAME, C.DATA_TYPE, C.ORDINAL_POSITION
FROM   INFORMATION_SCHEMA.TABLES AS T
       INNER JOIN INFORMATION_SCHEMA.COLUMNS AS C
         ON T.TABLE_SCHEMA = C.TABLE_SCHEMA
         AND T.TABLE_NAME = C.TABLE_NAME;
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Liste les tables, avec le type de table ("BASE TABLE" ou "VIEW"), les colonnes avec leur type et leur position ordinale.

Notez la jointure sur le couple de colonnes TABLE\_SCHEMA + TABLE\_NAME

Dans le résultat, la colonne TABLE\_CATALOG fait référence au nom de la base contextuelle (historiquement, on parlait de "banques" de données, dans lesquelles on trouvait des "catalogues" de données).

# 1 / métadonnées

65

Métadonnées SQL Server (sys) principales au niveau base :

| Table_name       | Objet   |
|------------------|---|
| .all_objects     | Objets de la base et objets communs                         |
| .objects         | Objets de la base   |
| .all_columns     | Colonnes des objets de base et communs                      |
| .columns         | Colonnes des objets de la base                              |
| .all_sql_modules | Routines de la base et routines communes                    |
| .sql_modules     | Routines de la base   |
| .all_views       | Vues de la base et vues communes                            |
| .views           | Vues de la base   |
| .all_parameters  | Paramètres des routines de la base et des routines communes |
| .parameters      | Paramètres des routines de la base                          |

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Un premier extrait... Nombreuses : 404 vues + 34 fonctions table

Le contenu de ces vues diffère d'une base à l'autre.

.all\_... => les éléments de la base courante et ceux communs situés dans la base « ressource »

Vous pouvez obtenir le poster gratuit de l'ensemble des DMV (Data Management View) de la version 2012 de SQL Server à cette URL :

<http://www.microsoft.com/en-us/download/details.aspx?id=39083>

# 1 / métadonnées

66

Métadonnées SQL Server (sys) principales au niveau base, exemple :

```
SELECT o.type_desc, o.name, c.name
FROM   sys.all_objects AS o
       JOIN sys.all_columns AS c
           ON c.object_id = o.object_id
WHERE  is_ms_shipped = 1;
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

La jointure se fait sur l'object\_id, référence interne des objets.

is\_ms\_shipped (est-ce livré par Microsoft ?) indique si l'objet a été fourni avec SQL Server (donc système)

# 1 / métadonnées

67

Métadonnées SQL Server (sys) niveau serveur :

| Table_name      | Objet                        |
|-----------------|------------------------------|
| .servers        | Liste des serveurs liés      |
| .databases      | Liste des bases              |
| .master_files   | Fichiers de toutes les bases |
| .configurations | Configuration du serveur     |

Vues de niveau serveur.

Le contenu de ces vues est identique quelle que soit la base contextuelle.

Il y en a bien sûr beaucoup d'autres !



# 1 / métadonnées

68

Métadonnées SQL Server (sys) : niveau serveur, exemple :

```
SELECT db.name,  
       CASE WHEN source_database_id IS NOT NULL  
            THEN 'SNAPSHOT'  
            WHEN db."state" <> 0 THEN db.state_desc  
            WHEN db.is_in_standby = 1 THEN 'STANDBY'  
            WHEN db.is_read_only = 1 THEN 'READ ONLY'  
            ELSE db.state_desc  
       END AS etat, mf.*  
FROM   sys.databases AS db  
       INNER JOIN sys.master_files AS mf  
            ON db.database_id = mf.database_id
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Une base est identifiée par son database\_id  
Cette requête montre toutes les bases et les fichiers utilisés par les bases.

# 1 / métadonnées

69

Métadonnées SQL Server (sys) : quelques autres vues bien utiles, au niveau base :

| Table_name           | Objet                                |
|----------------------|--------------------------------------|
| .tables              | Tables de la base                    |
| .views               | Vues de la base                      |
| .types               | Types personnalisés de la base       |
| .sequences           | Séquences de la base                 |
| .triggers            | Déclencheurs de la base              |
| .procedures          | Procédures de la base                |
| .check_constraints   | Contraintes de validation de la base |
| .default_constraints | Contraintes de défaut de la base     |

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Et encore :

sys.identity\_columns : colonnes auto-incrémentées par IDENTITY

sys.computed\_columns : colonnes calculées

# 1 / métadonnées

70

Métadonnées SQL Server (sys), exemple :

```
SELECT OBJECT_SCHEMA_NAME(t.object_id),
       t.name AS TABLE_NAME,
       COALESCE((SELECT 'OUI'
                 FROM sys.identity_columns AS ic
                 WHERE ic.object_id = t.object_id), 'NON')
       AS CONTIENT_IDENTITY,
       COALESCE((SELECT DISTINCT 'OUI'
                 FROM sys.computed_columns AS ic
                 WHERE ic.object_id = t.object_id), 'NON')
       AS CONTIENT_COLONNE_CALCULEE
FROM   sys.tables AS t
```

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

Notez l'identifiant de schéma SQL : schema\_id.

OBJECT\_SCHEMA\_NAME est une fonction SQL Server permettant de retrouver le nom du schéma SQL par le biais d'un identifiant de l'objet.

Mieux vaut utiliser une jointure avec la vue sys.schemas.

# 1 / métadonnées

71

Métadonnées SQL Server (sys), quelques autres vues bien utiles, au niveau base :

| Table_name                   | Objet  |
|------------------------------|--|
| .schemas                     | Liste des schémas SQL de la base                 |
| .database_files              | Fichiers de la base                              |
| .indexes                     | Liste des index                                  |
| .index_columns               | Liste des colonnes des index                     |
| .stats                       | Liste des statistiques d'optimiseur              |
| .stat_columns                | Liste des colonnes des statistiques d'optimiseur |
| .sql_expression_dependencies | Liens entre les objets                           |
| .synonyms                    | Liste des synonymes                              |

Administration SQL Server - CH 01 - installation & configuration

V3.0 10/07/2017

En sus de `sys.sql_expression_dependencies` qui retourne les dépendances entre objets de la base par id, vous pouvez utiliser la fonction `sys.dm_sql_referencing_entities()` qui fournit les dépendances entre objets de la base par nom

NOTA : pour les dépendances entre objets, les liens non référencés à la première compilation (au moment de la création de l'objet) seront référencés lors de la première exécution de l'objet.

# 1 / métadonnées

72

Métadonnées SQL Server (sys), exemple :

```
SELECT OBJECT_SCHEMA_NAME(s.object_id) AS TABLE_SCHEMA,
       OBJECT_NAME(s.object_id) AS TABLE_NAME,
       COALESCE(i.name, s.name) AS NOM,
       CASE WHEN i.name IS NULL THEN 'column'
            ELSE 'index' END AS NATURE
FROM   sys.indexes AS i
       LEFT OUTER JOIN sys.index_columns AS ic
         ON ic.object_id = i.object_id
           AND ic.index_id = i.index_id
       RIGHT OUTER JOIN sys.stats AS s
         ON i.object_id = s.object_id
           AND i.name = s.name;
```

# 1 / outils

73

## Aide en ligne

- Désormais sur Internet
- Possibilité de télécharger une aide succincte (surtout la syntaxe des commandes)
  - Installation et configuration via SSMS (dans le menu "?")
- **Prévoyez cette installation pour les serveurs n'ayant pas accès à Internet !**

L'aide téléchargeable n'est plus disponible qu'en langue anglaise depuis la version 2014, afin d'éviter toute ambiguïté et toute erreur de traduction.

<https://www.microsoft.com/en-us/download/details.aspx?id=42557>



# SQL SERVER 2012/2017 ADMINISTRATION

Frédéric Brouard (SQLpro) [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
Richard Gaillard - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)

Version 3.001  
Date 10/07/2017



## 2

## Gestion des espaces de stockage

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

Sans mention particulières, figures et photos sont de Frédéric BROUARD



## 2 – Gestion des espaces de stockage

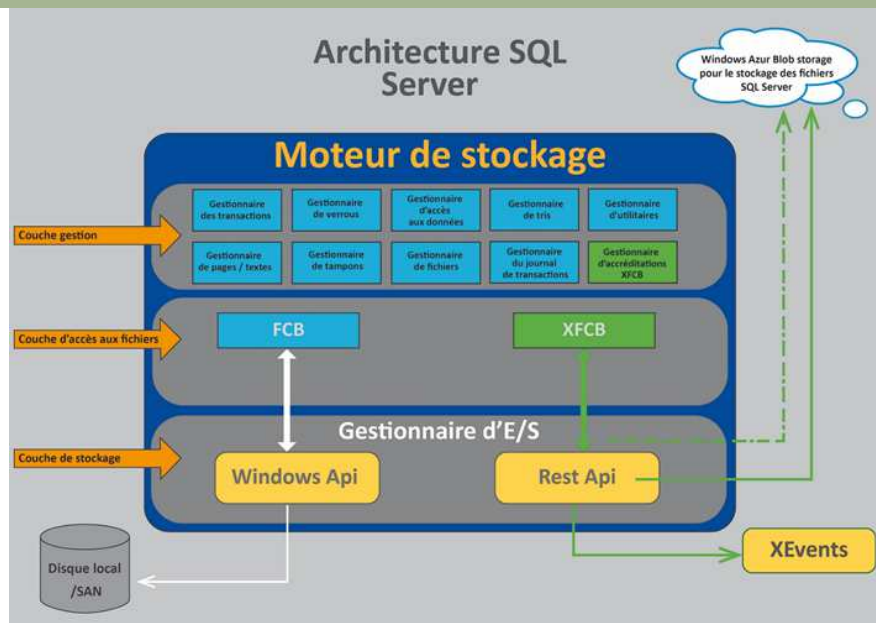
3



- Moteur de stockage
- Types de stockage
- Groupes de fichier et fichiers
- Gestion des fichiers
- Gestion des espaces morts
- Structure des tables et index
- Pages et extensions
- Tables, données et colonnes calculées
- Clichés de bases de données
- Import/export des données

## 2 / moteur de stockage

4



Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Architecture en 3 couches :

Couche gestion : transaction, verrouillage, tri, méthodes d'accès, utilitaires (sauvegarde, vérification, chargement...), page et LOBs, tampons, fichiers de données et journal de transaction

Couche de contrôle : [eXternal] File Control Bloc (*eXternal pour stockage des fichiers vers le cloud Azure*)

Couche de stockage : Windows API, REST API (Representational State Transfer : pour le stockage distribué)

Au sujet de REST, lire :

[http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)

La possibilité de stocker ses bases dans Azure est apparue avec la version 2012.

Figure (Igor Pagliai) tirée de :

<http://www.datanami.com/wp-content/uploads/2014/03/microsoft-data-platform.jpg>

## 2 / types de stockage

5

- **Tables et index :**
  - fichier(s) via groupe(s) de fichier(s)
- **Transactions :**
  - Fichier WAL (Write Ahead Log)
- **LOBs hors table, constitués par des fichiers (stockées sous le contrôle de SQL Server) :**
  - Filestream
  - Filetable
- **In Memory :**
  - Tables non persistantes, journal propre

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Plus génériquement on utilise le terme de « storage »

LOBs : Large Objects (objets peu relationnel stockés sous le contrôle de la base) déclinés en :

- BLOBs : Binary LOBs (stocke du binaire, par exemple son, image, vidéo) via le type **VARBINARY(max)**
- CLOBs : Character LOBs (encodage « ASCII ») via le type **VARCHAR(max)**
- NCLOBs : National Character LOBs (encodage « UNICODE ») via le type **NVARCHAR(max)**

**ATTENTION** : Les types de données IMAGE, TEXT et NTEXT sont obsolètes depuis la version 2005, mais on les rencontre encore hélas fréquemment. Les transformer !

FILESTREAM correspond au « datalink » de la norme SQL et est apparu avec la version 2008 de SQL Server. Il permet de stocker des fichiers sous le contrôle du SGBDR. FILETABLE permet de stocker des fichiers dans une arborescence du système de fichier sous le contrôle du serveur SQL. Apparu avec la version 2012 de SQL Server.

Les tables « in memory » nécessite un groupe de fichier spécifique et les options :

- MEMORY\_OPTIMIZED = ON
- DURABILITY = SCHEMA\_ONLY ou SCHEMA\_AND\_DATA

L'intégralité de la table figure en mémoire.

## 2 / types de stockage

6

### REMARQUE IMPORTANTE

- Vous devez considérer les fichiers de SQL Server :
  - données
  - transactions
  - sauvegardes
- **...comme si c'étaient des disques d'un OS**

SQL Server utilise des routines de lectures et d'écriture des fichiers qui « squeeze » le comportement habituel de l'OS à ce sujet (utilisation directes d'API)

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Pour une utilisation optimale et performante, il est nécessaire de correctement gérer les fichiers de SQL Server (capacity planning – voir dans ce même chapitre).

Taille, placement, organisation, caractéristique des disques physiques et type de RAID peuvent influencer de manière très importante sur les performances !

N'oubliez pas avant tout qu'un SGBDR quel qu'il soit est un système de stockage des données...

En particulier le couple SQL Server / Windows permet des gains fantastique au niveau des accès disque lorsqu'il est bien géré.

Au contraire, l'utilisation d'un stockage virtuel ou plus généralement d'une VM mal maîtrisé, donne des résultats exécrables.

Pour les grosses installations, (base de plusieurs centaines de Go, plusieurs centaines d'utilisateurs...) la virtualisation est déconseillée.

Pour information, un rôle particulier de niveau serveur s'appelle « diskadmin ». Il existe depuis la version 6 de SQL Server

**IMPORTANT** : le serveur hôte doit avoir des disques formatés avec des cluster de 64 Ko  
<https://technet.microsoft.com/en-us/library/dd758814%28v=sql.100%29.aspx>

## 2 / groupe de fichiers

7

CREATE DATABASE...

Par défaut, il est créé un groupe de fichiers (storage) nommé :

### **PRIMARY**

Ce groupe de fichiers contient un fichier (.mdf)

*mdf = master data file*

Dans le même temps un fichier est créé pour gérer les transactions (.ldf)

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Par défaut :

Données de la base : <nombase>.mdf => (mdf : master data file) 5 Mo, croissance 1 Mo

Transaction de la base : <nom\_base>\_log.ldf => (ldf : log data file) 2 Mo, croissance 10%

Les extensions .mdf et .ldf sont conventionnelles.

L'extension .ndf (secondary data file) est utilisée par défaut pour tout autre fichier de données.

## 2 / groupe de fichiers

8

### Rôle du groupe de fichier **PRIMARY** :

- Sert de point d'entrée pour accéder au fichier comportant la définition de l'organisation du stockage de la base
- DATA\_SPACE\_ID = 1

### Rôle du fichier « **master data file** »

- Contient la définition de l'organisation du stockage de la base
- Est toujours situé dans le groupe de fichier PRIMARY
- Non supprimable
- FILE\_ID = 1

Le nom logique du fichier primaire (master data file) comme son nom sur disque et son extension n'a aucune importance.

Seul compte son identifiant dans les tables système (file\_id = 1).

Par défaut, SQL Server le crée en empruntant le nom de la base suivi de l'extension .mdf.

## 2 / groupe de fichiers

9

Objet logique de la base, contenant 0 à n fichier(s)

Destinataire des objets porteurs d'informations :

- Tables
- Index

Caractéristique :

- Nom (identifiant SQL)
- Mode : READ\_ONLY ou READ\_WRITE

Pour chaque BD : 1 groupe de fichiers par défaut

Un groupe de fichiers vide ne peut pas servir de destination à un objet de la base et ne peut être le groupe de fichier par défaut.

Il doit toujours exister un groupe de fichiers de nom PRIMARY qui contient au moins un fichier (appelé « master data file » - fichier *primaire*), souvent identifié par l'extension .mdf.

PRIMARY est le groupe de fichiers par défaut, sans autre précision.

Vous pouvez changer le groupe de fichier par défaut, mais vous ne pouvez pas supprimer le groupe de fichier PRIMARY, ni le premier fichier créé (master data file – fichier primaire)

## 2 / groupe de fichiers

10

Créer un groupe de fichier :

```
ALTER DATABASE <nom_base>  
    ADD FILEGROUP <nom_filegroup>
```

Supprimer un groupe de fichier (si aucun fichier) :

```
ALTER DATABASE <nom_base>  
    REMOVE FILEGROUP <nom_filegroup>
```

Exemples :

```
ALTER DATABASE MaBase  
    ADD FILEGROUP FG_DATA;
```

```
ALTER DATABASE MaBase  
    ADD FILEGROUP FG;
```



## 2 / groupe de fichiers

11

Modifier un groupe de fichier :

```
ALTER DATABASE <nom_base>
MODIFY FILEGROUP <nom_filegroup>
    { { READ_ONLY | READ_WRITE }
    | DEFAULT
    | NAME = <nouveau_nom> }
```

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Exemples :

```
ALTER DATABASE MaBase
MODIFY FILEGROUP FG_DATA
DEFAULT;
```

```
ALTER DATABASE MaBase
MODIFY FILEGROUP FG
NAME = FG_INDX;
```

## 2 / groupe de fichiers

12

Pour spécifier la destination des tables et index :

- Utiliser la clause ON dans le CREATE de l'objet

```
CREATE TABLE <nom_table> (<description>)  
ON <filegroup>
```

```
CREATE INDEX <nom_index>  
ON <nom_table> (<liste_cols>)  
ON <filegroup>
```

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Exemple :

```
CREATE TABLE T_EMAIL  
(ID INT IDENTITY PRIMARY KEY,  
MAIL VARCHAR(256))  
ON FG_DATA;
```

```
CREATE INDEX X ON T_EMAIL (MAIL)  
ON FG_INDX;
```

## 2 / groupe de fichiers

13

Pour les LOBs (BLOB, CLOB, NCLOB) possibilité de dédier un groupe de fichier spécifique.

Dans ce cas, utilisez l'option TEXTIMAGE\_ON :

```
CREATE TABLE <nom_table> (<description>)  
ON <filegroup>  
TEXTIMAGE_ON <filegroup_LOBs>
```

Le groupe de fichier destinataire des LOBs (TEXTIMAGE\_ON) recevra les données des types :

- text, ntext, image (obsolète)
- xml,
- varchar(max), nvarchar(max), varbinary(max)
- geometry, geography (SIG)

Et tous les types CLR définis par l'utilisateur

Si forte volumétrie de LOBs (par exemple GED) utilisez RBS (Remote Blob Storage).

À lire : [http://download.microsoft.com/download/9/5/2/9521D8DA-5D3C-4817-BB9D-B5B1BD293365/SQL\\_Server\\_2008\\_R2\\_Remote\\_Blob\\_Storage.docx](http://download.microsoft.com/download/9/5/2/9521D8DA-5D3C-4817-BB9D-B5B1BD293365/SQL_Server_2008_R2_Remote_Blob_Storage.docx)

## 2 / fichier

14

Objet logique de la base, pointant sur un fichier physique :

- doit être inclus dans un groupe de fichier pour les données
- pour les transactions, spécifiez par « LOG »

Caractéristiques :

- Nom logique (unique au sein de la base)
- Emplacement physique (unique dans le SI)
- Taille (limite 1 6 To (data) / 2 To (transactions))
- Taille d'incrément
- limite de taille

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

NAME : nom logique

FILENAME : emplacement physique

SIZE : taille de départ en octets (Ko, Mo, Go, ou To)

FILEGROWTH : pas de croissance en octets ou %

MAXSIZE : limite de taille en octets

Le nombre maximal de fichiers par base est de 32 767 soit à 16 To le fichier une base d'une capacité maximale de 524 272 To (soit 524 Po).

Le nombre maximal de base sur une même instance est de 32 760, soit au plus 17 175 150 720 To (16 772 608 Po)

Le nombre maximal d'instance par machine étant de 50, la volumétrie maximale de stockage via SQL Server pour une même machine est théoriquement limitée à :

858 757 536 000 To (838 630 406 Po).

## 2 / fichier

15

Peuvent être définis lors de la création de la base :

CREATE DATABASE ...

Peuvent être ajoutés à tout moment :

ALTER DATABASE ... ADD FILE (...) TO FILEGROUP;

ALTER DATABASE ... ADD LOG FILE (...)

Peuvent être modifiés, supprimés à tout moment :

ALTER DATABASE ... { REMOVE | MODIFY } FILE (...)

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Par défaut, un CREATE DATABASE MA\_BASE se résume, au niveau fichiers/groupe de fichiers à :

```
CREATE DATABASE MA_BASE
ON PRIMARY
(NAME = N'MA_BASE',
 FILENAME = N'C:\...\MA_BASE.mdf',
 SIZE = 4288KB,
 MAXSIZE = UNLIMITED,
 FILEGROWTH = 1024KB )
LOG ON
( NAME = N'MA_BASE_log',
 FILENAME = N'C:\...\MA_BASE_log.ldf',
 SIZE = 1344KB,
 MAXSIZE = 2048GB,
 FILEGROWTH = 10%)
```

L'emplacement par défaut des fichiers est défini lors de l'installation.

On peut le changer dans l'IHM (propriété du serveur, onglet « Paramètre de base de données », rubrique « Emplacement de la base de données »)

Cela conduit à lancer la procédure stockée système suivante :

```
EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE',
    N'Software\Microsoft\MSSQLServer\MSSQLServer',
    N'DefaultData', --> sinon DefaultLog, BackupDirectory
    REG_SZ,
    N'C:\...' --> nouvel emplacement
```

## 2 / fichier

16

Définition de fichier, exemple :

```
( NAME          = N'F_DATA ' ,  
  FILENAME      = N'C:\SQL\DATA1.ndf' ,  
  SIZE          = 1GB ,  
  MAXSIZE       = UNLIMITED ,  
  FILEGROWTH    = 25MB )
```

- Les tailles peuvent être exprimées en KB, MB, GB, TB
- Pour la croissance, on peut spécifier un %

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Sans spécification,

SIZE = 1 MB,

Pour les données : FILEGROWTH = 1 MB, MAXSIZE = UNLIMITED.

Pour les transactions : FILEGROWTH = 10%, MAXSIZE = 2 TB

## 2 / fichier

17

Ajout d'un fichier de donnée, exemple :

```
ALTER DATABASE MA_BASE
ADD FILE
(NAME          = N'F_DATA2' ,
FILENAME      = N'C:\SQL\DATA2.ndf' ,
SIZE          = 50 GB ,
FILEGROWTH   = 25 MB )
TO FILEGROUP [PRIMARY] ;
```

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Ceci ajoute un fichier de données au groupe de fichier PRIMARY de nom logique F\_DATA2, emplacement physique C:\SQL\DATA2.ndf, taille 50 Go, pas de croissance de 25 Mo, taille maximale non limitée.

Notez l'utilisation impérative des crochets pour entourer le nom du groupe de fichier PRIMARY, ceci afin de ne pas le confondre avec le mot clef de définition d'une clef primaire de table. Les crochets n'étant pas ordinairement obligatoire.

Vous pouvez ajouter plusieurs fichiers simultanément dans un même groupe de fichier. Séparez la définition de chaque fichiers par une virgule :

```
ADD FILE
( ... ),
( ... ),
( ... )
TO FILEGROUP ...
```

## 2 / fichier

18

Ajout d'un fichier de transaction, exemple :

```
ALTER DATABASE MA_BASE
ADD LOG FILE
(NAME          = N'F_TRAN' ,
FILENAME      = N'C:\SQL\TRAN2.1df' ,
SIZE          = 20 GB ,
FILEGROWTH    = 10 MB ) ;
```

Les fichiers du journal de transactions, ne sont rattachés à aucun groupe de fichiers.

Un seul fichier est utilisé de manière simultanée. Lorsque le premier est plein, le second prend le relais et ainsi de suite.



## 2 / fichier

19

Modification d'un fichier de données, exemple :

```
ALTER DATABASE MA_BASE
MODIFY FILE
(NAME          = N'F_DATA2' ,
SIZE          = 200 GB ,
MAXSIZE       = 1 TB ,
FILEGROWTH    = 25 MB );
```

Il n'est pas possible de modifier l'emplacement d'un fichier par ce biais. Il faudra détacher la base, déplacer le fichier et la rattacher.

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Le fichier concerné par la modification, doit être indiqué par son nom logique (NAME)

Pour modifier le nom logique d'une fichier, utilisez la caractéristique NEWNAME.

Exemple :

```
ALTER DATABASE MA_BASE
MODIFY FILE
(NAME      = N'F_DATA2',
NEWNAME   = N'F_DATA_RO');
```

Pour « déplacer » un fichier à chaud, il faut en fait en créer un nouveau, déplacer les objets via ALTER INDEX ... DROP\_EXISTING ... puis supprimer l'ancien fichier vide. Voir plus loin.

## 2 / fichier

20

### Modifier les emplacements physiques du stockage...

Il convient de :

- Détacher la base :  
`EXEC sp_detach_db 'MA_BASE'`
- Déplacer les fichiers au niveau système :  
`MOVE ...`
- Recréer la base avec rattachement :  
`CREATE DATABASE MA_BASE`  
`ON ( ... ),`  
`( ... )`  
`FOR ATTACH;`

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Dans la clause ON, spécifier tous les fichiers de la base.

On peut aussi en profiter pour reconstruire un journal de transactions comme à l'origine :

- 1) Ne pas spécifier le ou les fichier(s) du journal de transactions
- 2) Modifier l'option finale FOR ATTACH en FOR ATTACH\_REBUILD\_LOG

## 2 / fichier

21

Migrer les données d'un objet...

Peut se faire d'un groupe de fichier à l'autre.

Doit utiliser la commande :

```
CREATE ... INDEX <nom_index_existant>  
    WITH (DROP_EXISTING = ON, ...)  
    ON <nouveau_filegroup>
```

Les spécifications de l'index doivent être les mêmes.

Pour une table en HEAP, créez un index CLUSTERED et supprimez le.

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

L'index doit avoir une structure strictement identique à l'original pour ses paramètres logiques (UNIQUE, [NON]CLUSTERED...)

Pour savoir si une table est en HEAP :

```
SELECT type_desc  
FROM sys.indexes  
WHERE object_id = OBJECT_ID('schema.table')  
AND index_id <= 1;
```

## 2 / fichier

22

Supprimer un fichier, exemple :

```
ALTER DATABASE MA_BASE  
    REMOVE FILE F_DATA2 ;
```

Les fichiers doivent être vide :

- ❑ Migrez les données dans d'autres groupes de fichiers ou dans un autre fichier du même groupe;
- ❑ Vérifiez la migration par une requête de métadonnées dans les vues systèmes

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Pour vider un fichier de données en déplaçant ses données dans d'autre(s) fichier(s) du même groupe ::

```
DBCC SHRINKFILE (FILENAME, EMPTYFILE);
```

Pour s'assurer du vidage :

```
DBCC UPDATEUSAGE (0);
```

```
SELECT name AS FileName,
```

```
    size * 8 AS SIZE_KB,
```

```
    size * 8 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT) * 8 AS FREE_KB
```

```
FROM sys.database_files;
```

Il y a toujours un espace résiduel même lorsqu'un fichier est vide de données, du fait des pages « techniques »

Le fichier primaire comme le premier fichier du journal de transaction, ne peut être supprimé.

## 2 / planification de capacité

23

Il est important que :

- Les fichiers soient correctement dimensionnés pour absorber la charge pour toute la durée du serveur;
- La croissance soit préventive.
- Éviter toute opération de réduction des fichiers

*Pour les données, mais encore plus pour les transactions !*

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

N'oubliez pas que SQL Server considère les fichiers comme étant des disques... et formate les fichiers créés (pages et extensions) pour y recevoir les données

Lorsque vous créez un fichier, SQL Server recherche le meilleur emplacement sur le disque dont il connaît intimement la structure (nombre de plateaux, nombre de pistes et secteur des plateaux).

À moins qu'il ne s'agisse d'un disque virtuel (VM) ou bien que le niveau de RAID empêche de voir la structure (RAID 5, 6,, 50, 60, DP...)

Les opérations de croissance et décroissance pénalisent fortement les performances. Pour le fichier du journal de transactions, elles ont toute les chances d'intervenir en pleine transaction !

Vous pouvez accélérer la création des fichiers de données des bases en autorisant le service SQL Server à faire de l'« Instant File Initialization ».

<http://blog.developpez.com/mikedavem/p11775/sql-server-2005/instant-file-initialization-et-setfilevaliddata>

Néanmoins, ceci n'est pas toujours conseillé. Mais c'est relativement très intéressant pour la restauration de grosses bases !

## 2 / planification de capacité

24

Pour les fichiers de données :

- Prévoyez de dimensionner pour 3 à 5 années d'exploitation
- Estimation :
  - par calcul (via Power AMC par exemple)
  - En appliquant une règle de trois sur un test de volumétrie

Pour le fichier du JT : 20% des données

Les bases de données de type DATEWAREHOUSE peuvent avoir un journal plus petit si elle sont placées en mode de journalisation simple.

Certaines bases de données ont des fonctionnement particulier et peuvent avoir un JT plus ou moins important selon le cas...

## 2 / planification de capacité

25

### Conseils :

- ❑ Ne placez pas vos tables dans le groupe de fichier PRIMARY !
  - ❑ Isolez-le et donnez lui une petite taille (250 Mo par exemple)
- ❑ Créez au moins un groupe de fichiers avec plusieurs fichiers d'égale longueur
  - ❑ Le nombre de fichier dépend des disques physique à disposition (faites au moins du 1 pour 1)
- ❑ Créez des groupes de fichiers pour des besoins spécifiques

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Besoins spécifiques comme des stockages « read only » ou le partitionnement.

Respectez ces règles, même pour une VM ou un stockage sur SAN mutualisé

Pour la base de données tempdb, n'utilisez que le groupe de fichiers PRIMARY et multipliez le nombre de fichiers d'égale longueur en fonction des CPU/cœurs

## 2 / planification de capacité

26

Auditez régulièrement les espaces de stockage :

- Taux de remplissage des fichiers;
- Taux de remplissage des disques de l'OS sur lesquels des fichiers SQL sont stockés;
- Activité en IO des fichiers.

Faite du préventif !

1) Remplissage des fichiers des bases :

Voir article SQLpro « Métriques du stockage de vos bases »

<http://blog.developpez.com/sqlpro/p12910/ms-sql-server/metriques-du-stockage-de-vos-bases>

2) Remplissage des disques :

```
SELECT DISTINCT volume_mount_point,
    total_bytes / 1048576 AS DISK_SIZE_MO,
    available_bytes / 1048576 AS DISK_FREE_MO
FROM sys.master_files AS mf
    CROSS APPLY sys.dm_os_volume_stats(mf.database_id, file_id) AS osf
```

3) Activité en IO des fichiers :

```
SELECT db.name, mf.name, type_desc, physical_name, GETDATE(), vfs.*
FROM sys.master_files AS mf
    INNER JOIN sys.databases AS db
        ON mf.database_id = db.database_id
    CROSS APPLY sys.fn_virtualfilestats(mf.database_id, file_id) AS vfs
```

À relever régulièrement, par exemple toutes les heures et alimenter une table.



## 2 / disques physique

27

Les meilleures **performances** seront obtenues avec :

- Un stockage dédié (pas de SAN mutualisé)
- Des agrégats RAID :
  - De niveau 0 ou 1
- Des LUNs alignées aux disques physiques

***Dans tous les autres cas : dégradation sévère des performances !***

SQL Server « audite » la structure des disques qu'il doit utiliser afin de placer ses fichiers au meilleurs endroits pour les opérations qu'il doit effectuer.

Un agrégat RAID avec de l'entrelacement (5, 6, DP...) cause des problèmes de performances aux écritures de SQL Server qui ne peut optimiser ses accès.

Un SAN mutualisé cause de graves problèmes de performance à SQL Server qui doit parfois attendre que d'autres opérations concurrentes se terminent afin de poursuivre ses opérations d'accès.

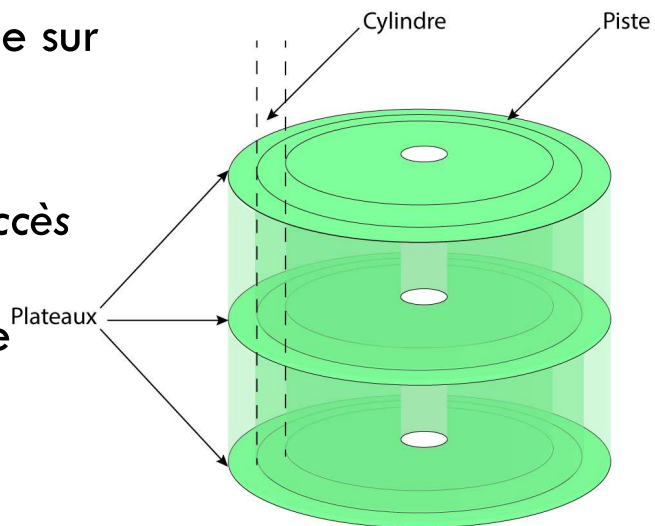
## 2 / disques physique

28

En présence d'un disque, SQL Server crée ses fichiers par cylindre sur tous les plateaux

*Cela optimise les accès*

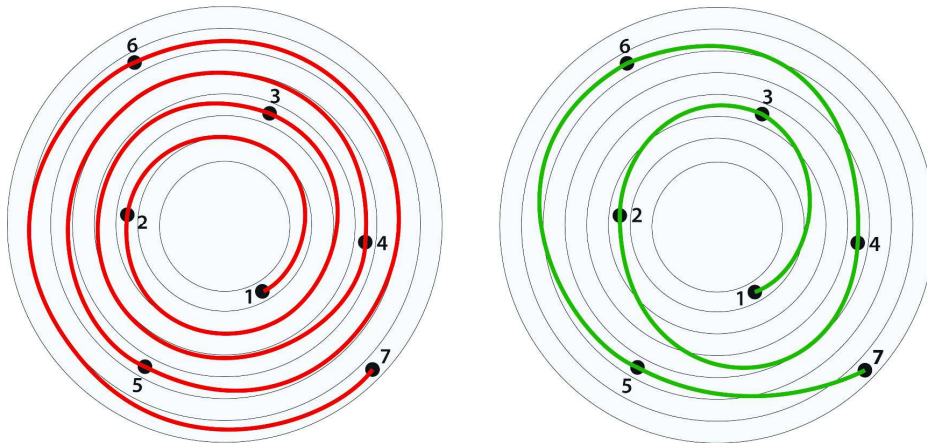
Favorisez la vitesse de rotation et non le temps d'accès



## 2 / disques physique

29

Cette technique permet, de prévoir le trajet optimale des accès disque



Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

En rouge, accès séquentiel (chronologique) des données : pas optimal !  
En vert, accès par regroupement de contiguïté des données : optimal !

Les données sont écrites sur le disque de manière asynchrone par page aléatoire dans les différents fichiers (parallélisme possible), par défaut environ toutes les minutes, tandis que les transactions sont écrites de manière synchrone (sauf dans le cas de la journalisation asynchrone via « delayed durability ») et de manière séquentielle dans un seul fichier.

## 2 / disques physique

30

### Machine virtuelle et performance

- ❑ Ne font généralement pas bon ménage
- ❑ Pertes de performance de 8 à 15 %
- ❑ Dégradation sévère des performances avec stockage virtuel

*Avec une VM, mieux vaut utiliser un stockage non virtuel*

**Agissez comme si c'était pas une VM !**

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Les études les plus récentes montrent que Hyper V de Microsoft est plus performant et beaucoup plus stable que VSphere (VMWare) !

<http://www.infoworld.com/article/2614229/virtualization/virtualization-showdown--microsoft-hyper-v-2012-vs--vmware-vsphere-5-1.html>

<https://www.petri.com/choose-windows-server-2012-r2-hyper-v-over-vmware-vsphere>

<http://blogs.technet.com/b/keithmayer/archive/2013/09/24/vmware-or-microsoft-comparing-vsphere-5-5-and-windows-server-2012-r2-at-a-glance.aspx>

<https://redmondmag.com/Articles/2015/05/01/vSphere-vs-HyperV.aspx?Page=2>

<http://www.computerweekly.com/feature/VMware-vs-Hyper-V-storage-comparison>

**En particulier les sauvegardes via VEEAM sont strictement incompatibles avec la haute disponibilité AlwaysOn !**

## 2 / transactions et performances

31

Le journal de transaction est généralement le premier point de contention.

Utilisez

- du RAID 10
- du SSD
- Les deux !

Sinon, possibilité de journalisation asynchrone via « delayed durability » :

Paramétrage de la base :

```
ALTER DATABASE <nom_base>  
    SET DELAYED_DURABILITY =  
    { DISABLED | ALLOWED | FORCED };
```

Lors de la validation de la transaction :

```
COMMIT TRANSACTION  
    WITH (DELAYED_DURABILITY = ON);
```

Le second point de contention est le stockage dans la base tempdb

Pour les tables In Memory, afin de journaliser de manière asynchrone, vous devez commencer la transaction par :

```
BEGIN ATOMIC WITH (DELAYED_DURABILITY = ON, ...)
```

Remarque : la tempdb est une base à part ayant son propre journal de transaction. Du fait que les tables temporaires sont créées dans cet espace, la journalisation est parallèle à celle de la base de production.

D'autres journalisations en parallèle existent :

- L'une pour chaque espace FILESTREAM, FILETABLE
- Pour chaque table « in memory »

## 2 / Gestion des espaces morts

32

### Réduction de la taille des fichiers

#### **Ne jamais systématiser**

- Au niveau base : DBCC SHRINKDATABASE<sup>1</sup>
- Au niveau fichier : DBCC SHRINKFILE

1) Évitez absolument l'utilisation de cette commande !

La réduction de la taille des fichiers d'une base doit rester une opération exceptionnelle sauf sur les installations à très faible espace (CE, LocalDB...)

Ce type d'opération est très pénalisante pour les performances et génère une fragmentation physique des fichiers qui est irréfragable.

De plus, une base étant toujours en croissance, il va falloir refaire une opération de croissance du fichier => double peine !

## 2 / Gestion des espaces morts

33

DBCC SHRINKDATABASE

```
( nom_base | database_id | 0  
  [ , cible_pourcent ]  
  [ , { NOTRUNCATE | TRUNCATEONLY } ]  
) [ WITH NO_INFOMSGS ]
```

- *cible\_pourcent* : % d'espace libre résiduel à la fin de l'opération
- NOTRUNCATE : déplace les pages vides en fin de fichier
- TRUNCATEONLY : supprime les pages vides en fin de fichier

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

NO\_INFOMSGS : évite les messages intempestifs inutiles

Exemple :

```
USE msdb;
```

```
DBCC SHRINKDATABASE (0, 10) WITH NO_INFOMSGS;
```

Compacte les fichiers de la base msdb en déplaçant les pages vides en fin de fichier et en laissant 10% de pages vides en queue.

## 2 / Gestion des espaces morts

34

### DBCC SHRINKFILE

```
( { nom_fichier | file_id }  
  { [ , EMPTYFILE ]  
    | [ [ , cible_taille ]  
      [ , { NOTRUNCATE | TRUNCATEONLY } ]  
  }  
)  
 [ WITH NO_INFOMSGS ]
```

- **cible\_taille** : taille espérée du fichier en Mo à la fin de l'opération
- **EMPTYFILE** : transfère les données de ce fichier vers les autres fichiers du même groupe

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

NOTRUNCATE, TRUNCATEONLY et NO\_INFOMSGS : voir DBCC SHRINKDATABASE

Exemple :

```
USE tempdb;  
DBCC SHRINKFILE (1, 100);
```

Résultat :

| DbId | FileId | CurrentSize | MinimumSize | UsedPages | EstimatedPages |
|------|--------|-------------|-------------|-----------|----------------|
| 2    | 1      | 12800       | 1024        | 328       | 328            |

(1 ligne(s) affectée(s))

Exécution de DBCC terminée. Si DBCC vous a adressé des messages d'erreur, contactez l'administrateur système.

A diminué le fichier 1 (données, fichier mdf) de la base tempdb à 12 800 pages (soit 102 400 Ko => 100 Mo) et indique que le minimum possible est de 1 024 pages dont 328 sont occupées.



## 2 / Structure des fichiers

35

Fichiers de données :

- Structurée par page de 8 Ko
- Pages organisées dans des extensions (extents) constituées par des blocs de 8 pages contiguës
- Les extensions peuvent être :
  - mixte : 8 pages de différents objets : table ou index
  - uniforme : 8 pages du même objet
- Lectures physique : 1 extension minimum
- Écritures physiques : 1 page minimum

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Dans SQL Server la taille des pages est fixe (8 Ko) optimal pour Windows, Azure (Microsoft) et Linux.

Les écritures sont faites de manière asynchrone par session (CHECKPOINT) environ toutes les minutes (par défaut). Réglage possible par `sp_configure 'recovery interval'` (en minutes), ou au niveau base (`TARGET_RECOVERY_TIME`)

La lecture physique par extension permet des lectures de pages « anticipées »

Par défaut, les extensions...

- mixtes sont constituées par les pages 0 des structures de stockage (table ou index), qui constitue les points d'entrée dans les objets
- uniformes sont constituées par 8 pages d'un même objet (table ou index)

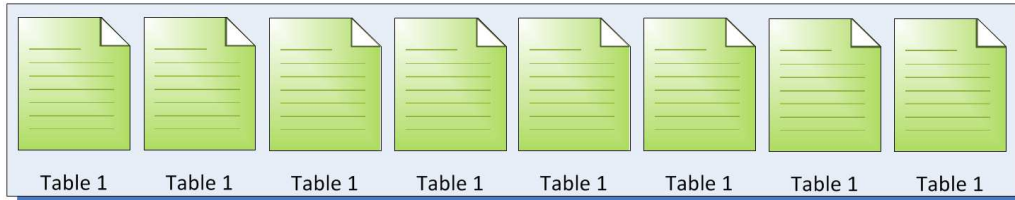
On peut changer cette organisation en utilisant les indicateurs de trace 1118 et 1119

## 2 / Structure des fichiers

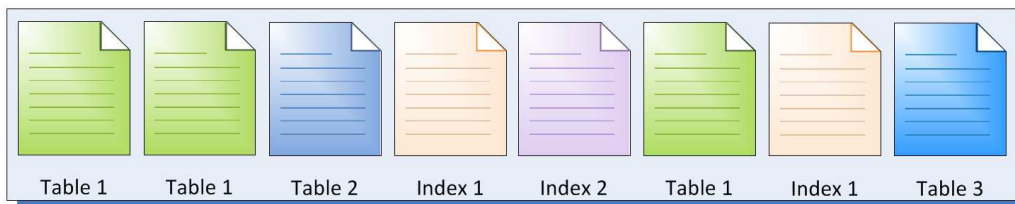
36

### Extensions mixtes et uniformes

Extension uniforme :



Extension mixte :



Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Pour imposer l'utilisation systématique d'extensions uniforme, mettez en œuvre l'indicateur de trace 1118 :

`DBCC TRACEON (1118, -1);`

Ou bien via la ligne de commande de démarrage du service SQL Server :

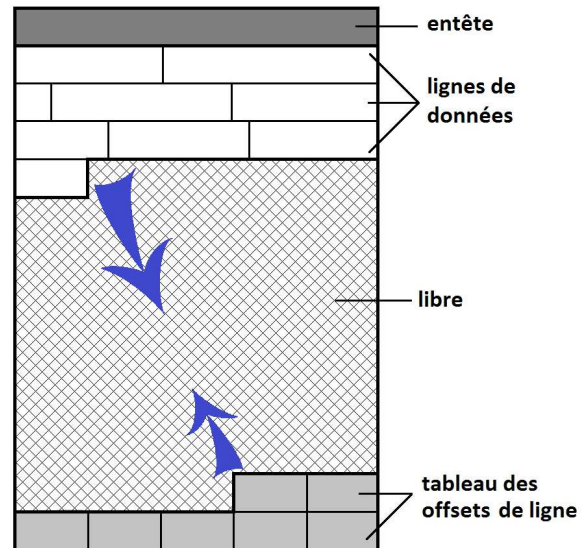
`-T1118`

## 2 / Structure des fichiers

37

Page de données (table ou index) :

- En haut, entête de 96 octets
- En bas, tableau des offsets d'emplacements de ligne
- Entre les deux : lignes de données



Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Les emplacements de ligne sont appelés « slots »

Dans Oracle la structure des pages est inversée entre le tableau d'offsets et les lignes de données.

## 2 / Structure des fichiers

38

Ligne de dans les pages :

- Pas d'ordre particulier pour une table en tas
- Triées si table organisée en cluster (par rapport à la clef de l'index clustered)
- Triées pour tout index (dit « non clustered »)

Ordre des informations dans la ligne (voir figure)



Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Des tables en tas sont appelées « heap » et sont présentées dans la vue sys.indexes come des index sans nom de type « heap ».

Des tables en cluster sont présentées dans la vue sys.indexes come des index de type « clustered » et cet index prend le nom de la contrainte de clef primaire.

Pour rappel une table « clustered » est relativement équivalent du concept d'IOT sous Oracle.

Le tag de version n'existe que si l'on met en œuvre le versionnement des lignes via le niveau d'isolation SNAPSHOT

Les données physiques ne sont pas organisées de la même façon que leur ordre logique pour des raisons d'efficacité et donc de performance...

Les tables « in memory » ont une structure très différente.

## 2 / Structure des fichiers

39

Fichier de transaction :

- WAL (Write Ahead Log) – écriture séquentielle
- Chaque entrée est repérée par un LSN (Log Segment Number)
- Structure binaire
- Lecture tabulaire possible via `sys.fn_dblog(...)`
- Utilisé en lecture en cas :
  - De ROLLBACK, pour reprendre les valeurs modifiées en mémoire;
  - D'arrêt brutal, au redémarrage pour récrire en mémoire les données modifiées par les dernières transactions.

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Le mécanisme du journal de transaction est basé sur l'algorithme ARIES que l'on trouve dans la plupart des SGBD relationnels.

À lire : <https://www.cs.berkeley.edu/~brewer/cs262/Aries.pdf>

```
SELECT * FROM sys.fn_dblog(NULL, NULL)
```

Lit l'intégralité du journal et la présente sous forme tabulaire. Vous pouvez préciser un LSN de début et un LSN de fin, afin de limiter la sortie.

À lire en complément : <http://rusanu.com/2014/03/10/how-to-read-and-interpret-the-sql-server-log/>

## 2 / Structure des fichiers

40

Fichier de transaction :

- Le plus sensible à la vitesse d'écriture physique car :
  - Les transactions sont suspendues le temps de l'écriture
    - Blocage des utilisateurs
  - Le volume peut être significatif
  - Un seul fichier utilisé simultanément
  
- Stockage dédié et rapide (RAID 0+1 ou 10)



**Point de contention n°1 dans les SGBD Relationnels**

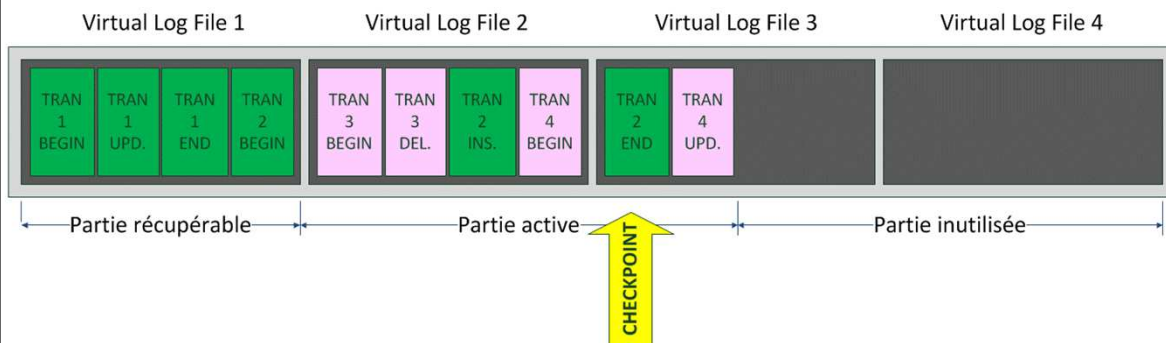
Les benchmarks du TPC.org montrent souvent des configurations à plus de 100 disques magnétiques ou encore plus de 10 SSD en parallèle (RAID 0) afin d'accélérer le débit d'écriture du journal.

## 2 / Structure des fichiers

41

Fichier de transaction, structure :

- Découpé en VLF (Virtual Log File) en mémoire
- Nombre et volume des VLFs dépendent de la taille de départ...



Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

La commande CHECKPOINT lancée environ toutes les minutes par défaut écrit les pages de données modifiées en cache dans les fichiers de données et marque le journal pour se repérer.

Avant le CHECKPOINT, c'est la partie « REDO LOG » du journal qui permet éventuellement de rejouer les transactions déjà passées.

Au CHECKPOINT et après, c'est la partie « UNDO LOG » qui permet d'annuler la transaction (ROLLBACK).

## 2 / Structure des fichiers

42

Fichier de transaction, structure :

- Le mode de journalisation détermine la volumétrie journalisée et la stratégie de reprise à entreprendre...
  - FULL (complet) : journalisation maximale et aucune purge
  - BULK LOGGED (journalisée en bloc) : journalisation minimale et aucune purge
  - SIMPLE (simple) : journalisation minimale et purge automatique
- Le journal « cycle » s'il est purgé...

**L'absence de purge fait croître le fichier indéfiniment**

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Lorsqu'aucune sauvegarde complète de la base n'a eut lieu, le journal se comporte comme si la base était en mode de journalisation simple et ne croit donc pas. Il est recommandée d'effectuer une sauvegarde complète dès création de la base.



## 2 / Structure des fichiers

43

Fichier de transaction, journalisation BULK LOGGED :

N'enregistre pas le détails des opérations reproductibles

- INSERT ... SELECT ...
- CREATE/ALTER INDEX...
- BULK INSERT...
- SELECT ... INTO ...

Généralement utilisé de manière transitoire lors des phases de maintenance

Par précaution, faire une sauvegarde du JT avant et après

Pour passer d'un mode de journalisation à l'autre, utilisez la commande  
ALTER DATABASE MaBase SET RECOVERY {FULL | BULK\_LOGGED | SIMPLE }  
Agit dynamiquement et passe d'un mode à l'autre dès que les transactions en cours sont terminées.

## 2 / Structure des fichiers

44

Fichier de transaction, purge :

- En cas de mode FULL ou BULK LOGGED, effectuer une sauvegarde régulière (planifiée) du journal

***Ceci génère des fichiers de REDO LOG***

- Les fichiers de sauvegarde du journal permettent de :
  - Compléter la restauration
  - Restaurer la base à un point dans le temps
  - Reconstituer la base sans perte en cas de crash des fichiers de données

Nous verrons qu'il faut effectuer une sauvegarde régulière des journaux de transactions des bases de production, en fonction de la volumétrie transactionnelle et de la perte maximale acceptable.

## 2 / Structure des fichiers

45

Fichier de transaction, purge :

- Ce qui peut empêcher la purge...
  - Le disque du JT plein :
    - Dans ce cas libérez quelques Mo sur le disque et passez en mode SIMPLE et repassez dans le mode initial
  - Un JT plein :
    - Augmentez la taille du fichier du JT ou créez en un autre
  - Une transaction anormalement longue :
    - Débusquez là via DBCC OPENTRAN
    - Analysez-en le code et tuez là (KILL...) si c'est possible

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

OPENTRAN peut être remplacée par des requêtes systèmes sur les vues. Exemple :

```
WITH
TO AS (SELECT MIN(database_transaction_begin_time)
      AS first_database_transaction_begin_time
      FROM sys.dm_tran_database_transactions AS TDT
      WHERE TDT.database_id = DB_ID())
SELECT TDT.transaction_id,
      TST.session_id,
      database_transaction_begin_time,
      CASE database_transaction_type
      WHEN 1 THEN 'READ WRITE'
      WHEN 2 THEN 'READ ONLY'
      WHEN 3 THEN 'SYSTEM'
      END TRANSACTION_TYPE,
      CASE database_transaction_state
      WHEN 1 THEN 'pas encore initialisée'
      WHEN 3 THEN 'initialisée, mais pas encore affecté de données'
      WHEN 4 THEN 'données affectées'
      WHEN 5 THEN 'préparée'
      WHEN 10 THEN 'validée'
      WHEN 11 THEN 'annulée'
      WHEN 12 THEN 'en cours de validation'
      END TRANSACTION_STATE,
      database_transaction_log_bytes_used,
      database_transaction_log_bytes_reserved,
      database_transaction_begin_lsn,
      database_transaction_last_lsn
FROM sys.dm_tran_database_transactions AS TDT
INNER JOIN TO
      ON TO.first_database_transaction_begin_time =
      TDT.database_transaction_begin_time
INNER JOIN sys.dm_tran_session_transactions AS TST
      ON TDT.transaction_id = TST.transaction_id;
```

## 2 / Structure des données

46

Les lignes de table peuvent être stockées dans :

- ROWS\_FILEGROUP : associé à un FILEGROUP
- PARTITION\_SCHEME : divisée en plusieurs FILEGROUPs
- FILESTREAM\_DATA\_FILEGROUP (fichiers) : structurés dans arborescence OS) sous forme de :
  - FILESTREAM
  - FILETABLE
- MEMORY\_OPTIMIZED\_DATA\_FILEGROUP : en mémoire (table « in memory »)

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Pour voir où sont stockées les données de votre base, il faut utiliser les tables système :

- sys.data\_spaces : espace de stockage des données
- sys.allocation\_units : unité d'allocation des données
- sys.partitions : liste des partitions

Le tout en partant des identifiants object\_id et index\_id

Exemple :

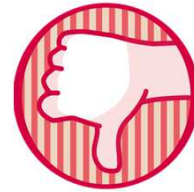
```
SELECT SCH.name AS OBJECT_SCHEMA,
       OBJ.name AS OBJECT_NAME,
       OBJ.type_desc AS OBJECT_TYPE,
       IDX.name AS INDEX_NAME,
       IDX.type_desc AS INDEX_TYPE,
       DS.name AS STORAGE_NAME,
       DS.type_desc AS STORAGE_TYPE,
       AU.type_desc AS STORAGE_MANNER,
       AU.total_pages / 128 AS TotalSizeMB,
       AU.used_pages / 128 AS UsedSizeMB,
       AU.data_pages / 128 AS DataSizeMB,
       rows AS ABOUT_NUM_ROWS,
       data_compression_desc AS DATA_COMPRESSION
FROM sys.data_spaces AS DS
     INNER JOIN sys.allocation_units AS AU
       ON DS.data_space_id = AU.data_space_id
     INNER JOIN sys.partitions AS PA
       ON (AU.type IN (1, 3) AND AU.container_id = PA.hobt_id)
          OR (AU.type = 2 AND AU.container_id = PA.partition_id)
     INNER JOIN sys.objects AS OBJ
       ON PA.object_id = OBJ.object_id
     INNER JOIN sys.schemas AS SCH
       ON OBJ.schema_id = SCH.schema_id
     LEFT JOIN sys.indexes AS IDX
       ON PA.object_id = IDX.object_id
          AND PA.index_id = IDX.index_id
ORDER BY 1, 2, 4;
```

## 2 / Structure des données

47

Les données des lignes de table peuvent être stockées en :

- IN\_ROW\_DATA : la ligne est dans la page
- LOB\_DATA : le LOB est hors ligne (pages spécialisées)
- **ROW\_OVERFLOW\_DATA** : la ligne déborde de la page (lignes de plus de 8060 octets hors LOBs)
  - Nécessite un pointeur de renvoi (24 octets).
- Évitez le Row Overflow !



Des lignes trop longues posent de multiples problèmes

La procédure `sp_tableoption` permet de stocker les données de type LOBs soit dans la page avec débordement possible ('large value types out of row' = 0, comportement par défaut) ou hors la page ('large value types out of row' = 1)

## 2 / Structure des données

48

### Compression :

- Au niveau ligne de table ou d'index :
  - DATA\_COMPRESSION = { NONE | ROW | PAGE }
- Au niveau des données :
  - Pour les nombres : vardecimal
  - Pour les colonnes « clairsemées » (sparse)

### La compression :

- gagne en place, lecture et souvent en recherche
- perd en écriture

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Pour mettre en œuvre la compression des tables et index, utilisez une reconstruction d'index par la commande :

```
ALTER INDEX ... REBUILD WITH (DATA_COMPRESSION = ... )
```

Pour mettre en œuvre la compression des nombres décimaux avec la procédure :

- sys.sp\_db\_vardecimal\_storage\_format (pour la base entière)
- sys.sp\_tableoption (avec l'option 'vardecimal storage format' pour une table particulière)

Auditez le gain préventivement avec

```
sys.sp_estimated_rowsize_reduction_for_vardecimal
```

Pour mettre en œuvre la compression des colonnes « clairsemées » (fort taux de NULL), utilisez le mot clef SPARSE dans la définition de colonne.

Exemples :

```
CREATE TABLE ... ( ... NOM_MARITAL CHAR(32) SPARSE NULL, ... )  
ALTER TABLE ... ALTER COLUMN NOM_MARITAL ADD SPARSE
```

Procédez toujours à des essais comparatifs avant et après mise en œuvre sur toutes les requêtes impactant la (ou les) table(s) compressée(s).

Les colonnes « sparse » et « vardecimal » ont de multiples restrictions d'emploi. Préférez la compression ROW ou PAGE

## 2 / Partitionnement

49

Permet de diminuer ou ventiler la plage des données scrutées pour :

- Les recherches : « diviser pour régner »
- Les mises à jour : parallélisme
- Le comportement physique :
  - ▣ Partition en READ ONLY pour archive par exemple
  - ▣ Escalade de verrou au niveau partition avant table

Une colonne NOT NULL sert pour la ventilation

S'effectue en 3 étapes...

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

La colonne sur laquelle est ventilée des données peut être une colonne calculée persistante (par exemple une concaténation de différentes colonnes ou un hachage d'autres colonnes). Ce qui permet de simuler :

- Un sous partitionnement
- Un partitionnement par hachage

## 2 / Partitionnement

50

### 1 – créer une fonction de partitionnement

```
CREATE PARTITION FUNCTION nom_fonction_partition  
  ( type_SQL_parametre )  
AS RANGE [ LEFT | RIGHT ]  
FOR VALUES ( [ limite1 [ limite2 [ ,... ] ] )
```

Ceci pose les « piquets »

□ N piquets => n + 1 partitions !

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Exemple :

```
CREATE PARTITION FUNCTION PF_DATE_FACTURATION (DATE)  
AS RANGE LEFT  
FOR VALUES ('2013-09-01', '2014-09-01', '2015-09-01');
```



## 2 / Partitionnement

51

### 2 – créer un schéma de partitionnement

```
CREATE PARTITION SCHEME nom_schéma_partition
AS PARTITION nom_fonction_partition
{ [ ALL TO nom_groupe_fichiers]
  | TO (nom_groupe_fichiers1, nom_groupe_fichiers2 [,
... ] ) ]
```

Ceci ventile les partitions dans les différents groupes de fichiers

- Créez les groupes de fichiers et les fichiers associés préalablement !

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

NOTEZ que le schéma de partition fait référence à la fonction de partition (« emboitage »)

Exemple, suite :

```
CREATE PARTITION SCHEME PS_DATE_FACTURATION
AS PARTITION PF_DATE_FACTURATION
TO (FG_OLD, FG_2013_14, FG_2014_15, FG_2015_16, FG_NEW);
```

Notez l'avertissement :

Le schéma de partition 'PS\_DATE\_FACTURATION' a été créé avec succès. 'FG\_NEW' est marqué comme étant le prochain groupe de fichiers utilisé dans le schéma de partition 'PS\_DATE\_FACTURATION'.

Ce groupe de fichier pourra être utilisé ultérieurement si l'on décide de rajouter une partition

## 2 / Partitionnement

52

### 3 – création des objets partitionnés

```
CREATE TABLE / INDEX ...  
    ON nom_schéma_partition(colonne_de_ventilation)
```

- Tables et index peuvent être ventilés de différentes manières.
- Dans un index, la colonne de ventilation doit être la première de la liste des colonnes de la clef d'index
- Une table ne peut pas être à la fois « clustered » et partitionnée

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

NOTEZ que l'on utilise le nom du schéma de partitionnement comme si c'était une fonction dont le paramètre dépend justement de la fonction de partitionnement

Pour une table existante avec un index clustered, deux possibilités :

- migrer les données dans une nouvelle table (complexe mais propre)
- migrer le stockage via la suppression de l'index clustered (si c'est la PK, supprimez la contrainte et recréez là avec un index non clustered) puis créez un index clustered sur le schéma de partitionnement et finalement, supprimez le !

Exemple :

```
ALTER TABLE ... DROP CONSTRAINT PK_...;  
ALTER TABLE ... ADD PRIMARY KEY NONCLUSTERED (...);  
CREATE CLUSTERED INDEX CX_... ON ... (...)  
    ON [PS_DATE_FACTURATION] (...);  
DROP INDEX CX... ON ...;
```

Le tout dans une transaction !

## 2 / Partitionnement

53

Vérification du partitionnement :

```
SELECT *, $partition.nom_fonction_partition  
          (colonne_partitionnée)
```

Permet dans une requête SELECT de voir dans que numéro de partition figure chaque ligne

Utilisez la vue sys.partitions en complément

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Exemple :

```
SELECT *, $partition.PF_DATE_FACTURATION(FAC_DATE) AS PART_NUMBER  
FROM ...
```

ATTENTION : cette fonction calcule dans quelle numéro de partition doit se trouver la ligne. Elle ne vérifie pas l'existence des partitions. Elle peut donc être utilisée préventivement.

## 2 / Partitionnement

54

Gestion du partitionnement :

Il est possible d'ajouter une partition. Procéder en deux temps :

```
ALTER PARTITION SCHEME nom_schema_partition  
    NEXT USED nom_file_group
```

Pour ajouter un espace de stockage (groupe de fichiers) au partitionnement

```
ALTER PARTITION FUNCTION nom_fonction_partition()  
    SPLIT RANGE (nouvelle_limite)
```

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Le groupe de fichier doit exister. Le créer préalablement.

Ces fonctions agissent en arrière plan. Elle rendent la main immédiatement. Les utilisateurs peuvent continuer à travailler.

Exemple :

```
ALTER PARTITION SCHEME PS_DATE_FACTURATION  
    NEXT USED FG_2010;  
ALTER PARTITION FUNCTION PF_DATE()  
    SPLIT RANGE ('2010-01-01');
```

## 2 / Partitionnement

55

Gestion du partitionnement :

Il est possible de fusionner deux partitions :

```
ALTER PARTITION FUNCTION nom_fonction_partition ()  
    MERGE RANGE (limite);
```

Ceci supprime la valeur pivot du partitionnement et range les lignes qualifiée dans la partition de droite ou de gauche suivant la création de la fonction de partitionnement.

## 2 / Partitionnement

56

Gestion du partitionnement :

Il est possible de déplacer une partition d'une table l'autre :

```
ALTER TABLE [nom_schema.] nom_table
    SWITCH PARTITION numero_partition_départ
        TO [autre_schéma.] autre_table
        PARTITION numero_partition_arrivée;
```

Les table doivent être identiques d'un point de vue logique et intégralement stockées dans le partitionnement (index compris) et la partition de destination doit être vide.

Par défaut toute table non partitionnée est stockée dans une partition unique de numéro 1

Exemple :

```
ALTER TABLE ...
    SWITCH PARTITION 1
        TO ... PARTITION 1;
```

## 2 / Partitionnement

57

### Métadonnées du partitionnement :

| Objet                                   | Nature   | Description                           |
|---|----------|---------------------------------------|
| sys.partitions                          | Vue      | liste des partitions                  |
| sys.partition_functions                 | Vue      | fonctions de partitionnement          |
| sys.partition_parameters                | vue      | paramètres des fonctions de partition |
| sys.partition_range_values              | vue      | limite des plages de partitionnement  |
| \$partition.nom_fonction_partition(...) | Fonction | répartition des lignes                |

## 2 / Colonnes, types, clefs

58

5 familles de types :

**Littéraux** : char, varchar, nchar, nvarchar

**Numériques** : int, smallint, bigint, tinyint, decimal, numeric, float, real

**Temporels** : date, datetime2, time, datetimeoffset

**Binaires** : bit, binary, varbinary, hierarchyid, uniqueidentifier

**LOBs** : varchar(max), nvarchar(max), varbinary(max), XML, geometry, geography, sql\_variant

Les types MONEY et SMALLMONEY sont des NUMERIC avec 4 décimales. Ils sont fortement déconseillés.

Le type DATETIME est obsolète depuis la version 2008 (à remplacer par DATETIME2)

Les types TEXT, NTEXT et IMAGE sont obsolète depuis la version 2005 (à remplacer par VARCHAR(max), NVARCHAR(max) et VARBINARY(max))

Le type TIMESTAMP est obsolète depuis la version 2005 (à remplacer par ROWVERSION)

Le type sql\_variant est à déconseillé



## 2 / Colonnes, types, clefs

59

Types littéraux :

|          | ASCII      | UNICODE     |
|----------|------------|-------------|
| FIXE     | CHAR(n)    | NCHAR(n)    |
| VARIABLE | VARCHAR(n) | NVARCHAR(n) |

Taille en octets :

|          | ASCII            | UNICODE              |
|----------|------------------|----------------------|
| FIXE     | N octets         | 2 * n octets         |
| VARIABLE | 2 à n + 2 octets | 2 à 2 * n + 2 octets |

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

N signifie NATIONAL (norme SQL)

Le coût du stockage UNIODE est double de celui de l'ASCII, mais permet l'utilisation d'alphabets non Latin (arabe, hébreu, chinois, japonais...)

## 2 / Auto incréments

60

Deux possibilités :

IDENTITY :

- Propriété d'une colonne de la table de type entière ou NUMERIC/DECIMAL
- Automatiquement alimenté
- Un seul par table

SEQUENCE :

- Indépendante de toute table
- Nécessite de prévoir l'appel dans l'insertion ou en valeur par défaut

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Syntaxes (les deux systèmes sont aujourd'hui normalisés) :

```
IDENTITY [ ( valeur_initiale , incrément ) ]
```

```
CREATE SEQUENCE [nom_schema . ] nom_séquence  
[ AS [ nom_type_sql | nom_udt ] ]  
[ START WITH <constante> ]  
[ INCREMENT BY <constante> ]  
[ { MINVALUE [ <constante> ] } | { NO MINVALUE } ]  
[ { MAXVALUE [ <constante> ] } | { NO MAXVALUE } ]  
[ CYCLE | { NO CYCLE } ]  
[ { CACHE [ <constante> ] } | { NO CACHE } ] [ ; ]
```

Les SEQUENCE sont apparue dans SQL Server 2012

L'utilisation de SEQUENCE pose plus de problèmes de « hot spot » que le recours à l'IDENTITY (un seul par table).

Une colonne IDENTITY ne peut plus être mise à jour par UPDATE.

## 2 / Clefs et contraintes

61

**PRIMARY KEY** (une seule par table) :

- création d'un index CLUSTERED par défaut

**UNIQUE** :

- création d'un index NONCLUSTERED par défaut

**FOREIGN KEY** :

- pas de création d'index

**Contrainte de table CHECK** (portée : ligne)

**Contraintes de colonne** : NOT NULL, DEFAULT, CHECK.

Il est important d'index les FOREIGN KEYS !

Une contrainte CHECK peut incorporer une UDF qui peut lire d'autres tables. Mais attention aux mises à jour réciproque. Mieux vaut utiliser des déclencheurs dans ce cas.

## 2 / Colonne calculée

62

Expression calculé introduite par le mot clef AS à la place du type.

- Peut supporter une contrainte :
  - De clef primaire
  - D'unicité
- Matérialisable par :
  - Simple persistance (mot clef PERSISTED)
  - Indexation

Pour être matérialisable, doit être déterministe

Pour éviter des dérives, mieux vaut que le calcul soit « précis »

**ATTENTION au typage !**

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Une colonne calculée non persistante et non indexée est strictement identique a ce que vous obtiendriez par une vue.

Exemple:

```
ALTER TABLE T_FACTURE
```

```
--> non persistante, non indexée
```

```
    ADD MONTANT_REMISE_HT AS MONTANT_HT * (1 - REMISE / 100.0),
```

```
--> persistante, non indexée
```

```
    MONTANT_TOTAL_HT AS MONTANT_HT * QUANTITE PERSISTED,
```

```
--> non persistante, indexée
```

```
    MONTANT_TOTAL_TTC AS MONTANT_HT * QUANTITE * (1 + TVA/100.0),
```

```
--> Persistante et indexée
```

```
    MONTANT_TOTAL_REMISE_TTC AS MONTANT_HT * QUANTITE * (1 - REMISE /  
100.0)
```

```
        * (1 + TVA/100.0) PERSISTED;
```

```
--> les index nécessaires
```

```
CREATE INDEX X_CALC_MONTANT_TOTAL_TTC ON T_FACTURE(MONTANT_TOTAL_TTC);
```

```
CREATE INDEX X_CALC_MONTANT_TOTAL_REMISE_TTC ON  
T_FACTURE(MONTANT_TOTAL_REMISE_TTC);
```

## 2 / Clichés de bases de données

63

### DATABASE SNAPSHOT :

consiste à instancier une nouvelle base copie de l'original et contenant les données telle qu'elles se trouvait à l'heure d'exécution du SNAPSHOT.

- Immédiat, quelque soit le volume de la base
- Lecture seulement
- Nécessite de prévoir le stockage des données à l'identique (sauf journal des transactions)

Syntaxe :

```
CREATE DATABASE nom_base  
ON ( description_fichiers_données )  
AS SNAPSHOT OF base_source;
```

## 2 / Clichés de bases de données

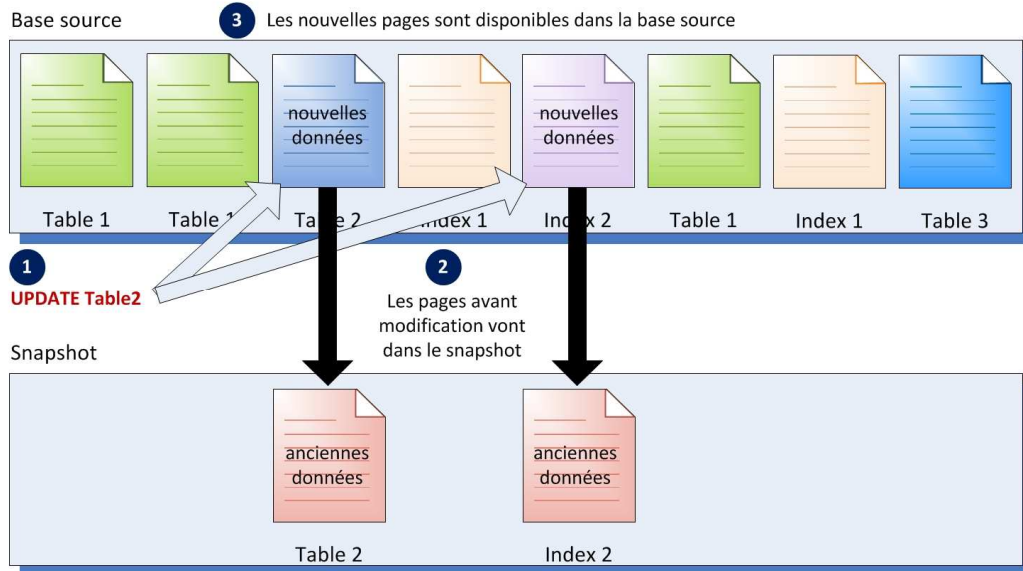
64

### DATABASE SNAPSHOT :

- Au départ, la base « clichée » ne contient pas de données.
- La mise à jour dans la base source envoie les pages sources avant modification dans la cible
- La base « clichée » est lue, pour les pages qu'elle contient et à défaut dans la base source

## 2 / Clichés de bases de données

65



Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

## 2 / Import / export de données

66

Différentes méthodes :

- BULK INSERT : commande Transact SQL
- bcp.exe : exécutable en ligne de commande
- SSIS (SQL Server Integration Services) : ETL
- Fonctions « OPEN... » : serveurs liés

BULK INSERT et SSIS utilisent bcp.exe

Insertion massivement parallèle depuis la version 2012

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Les fonctions « OPEN... » sont à déconseiller pour du chargement massif. Elles sont peu performantes et peu sécurisées.

OPENDATASOURCE : attaque une table à l'aide d'un fournisseur de données

OPENROWSET : envoie une requête à l'aide d'un fournisseur de données

OPENQUERY : lance une requête vers un serveur lié.

Exemples :

```
SELECT *
FROM OPENDATASOURCE
('SQLNCLI', --> fournisseur de données
'Data Source=MonServeur\MonInstance;Integrated Security=SSPI' --> chaine de connexion
).master.sys.databases; --> table distante
```

```
SELECT T.*
FROM OPENROWSET
('SQLNCLI', --> fournisseur de données
'Server=MonServeur\MonInstance;trusted_connection=yes;', --> chaine de connexion
'SELECT * FROM master.sys.databases' --> requête distante
) AS T; --> alias de table
```

--> création d'un serveur lié

```
USE master;
GO
EXEC master.dbo.sp_addlinkedserver @server = N'MonServeur\MonInstance',
    @srvproduct=N'SQL Server';
GO
EXEC master.dbo.sp_addlinkedsrvlogin @rmtsrvname = N'MonServeur\MonInstance',
    @locallogin = N'sa',
    @useself = N'True';
GO
```

--> utilisation du serveur lié

```
SELECT *
FROM OPENQUERY([MonServeur\MonInstance], 'SELECT * FROM master.sys.databases');
```



## 2 / Import / export de données

67

**BULK INSERT , syntaxe :**

```
BULK INSERT <matable>  
FROM <fichier>  
WITH ( <options> )
```

**bcp.exe - ligne de commande :**

```
bcp.exe [[<nom_base>.]<nom_schema>.]  
{<nom_objet>}|"<requete_SELECT>"  
{in|out|queryout|format} <fichier_donnees>
```

## 2 / Import / export de données

68

### BULK INSERT - paramètres

- CHECK\_CONSTRAINTS : vérifie les contraintes CHECK et FK,
- FIRE\_TRIGGERS : exécute les déclencheurs,
- KEEPIDENTITY : force des valeurs dans les auto incréments,
- KEEPNULLS : colonnes vides transformées en NULL,
  
- BATCHSIZE : spécifie la taille du lot de données
- KILOBYTES\_PER\_BATCH : découpe le lot par volume
- ROWS\_PER\_BATCH : découpe le lot par lignes

## 2 / Import / export de données

69

### BULK INSERT - paramètres

- ❑ FIRSTROW : détermine la première ligne
- ❑ FIELDTERMINATOR : spécifie le délimiteur de champs
- ❑ ROWTERMINATOR : spécifie le délimiteur de lignes
- ❑ LASTROW : détermine la dernière ligne
- ❑ ORDER : spécifie l'ordre de prise en compte des champs
  
- ❑ CODEPAGE : impose une page de code générique ou spécifique
- ❑ DATAFILETYPE : spécifie l'encodage du fichier
- ❑ FORMATFILE : indique un fichier de paramétrage du format

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Exemple :

```
BULK INSERT DB_SQLSERVER.dbo.T_DEPARTEMENT_ID_DPT
FROM 'C:\Temp\DepartementsIDF_id.txt'
WITH (FIELDTERMINATOR = ';',
      ROWTERMINATOR = '\n',
      FIRSTROW = 2,
      CODEPAGE = 'ACP',
      KEEPIDENTITY,
      BATCHSIZE = 7,
      ORDER (DPT_CODE),
      TABLOCK,
      ERRORFILE = 'C:\Temp\DepartementsIDF_error.log');
```

## 2 / Import / export de données

70

### Bcp.exe - commutateurs :

|                                   |                            |
|-----------------------------------|----------------------------|
| [-t <separateur_de_champs>]       | [-r <delimiteur_de_ligne>] |
| [-F <premiere_ligne>]             | [-L <derniere_ligne>]      |
| [-f <fichier_de_format>] [-x]     |                            |
| [-C {ACP OEM RAW <page_de_code>}] | [-c] [-n] [-w] [-N]        |
| [-R] [-k] [-E]                    | [-b <taille_du_lot>]       |
| [-m <nb_erreur_max>]              | [-e <fichier_erreur>]      |

- -c -n -w -N correspond au DATAFILETYPE et respectivement à char (-c), native (-n), widechar (-w), widenative (-N).
- -k correspond à KEEPIDENTITY et -E à KEEPNULLS.

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

**ATTENTION** : tous les commutateurs sont sensible à la casse !

-R spécifie que les données sont formatées aux spécifications régionales de la session SQL Server. Ceci concerne les types date, datetime, money et smallmoney.

Exemple :

```
bcp.exe DB_TEST.dbo.T_DEPARTEMENT_DPT
in C:\Temp\DepartementsIDF.txt
-S "MSSQLSERVER" -T
-t ';' -r '\n' -F 2 C ACP -c -b 7
-e C:\Temp\DepartementsIDF_error.log'
-h "ORDER (DPT_CODE), CHECK_CONSTRAINTS, TABLOCK"
```

## 2 / Import / export de données

71

### Bcp.exe - commutateurs

```
[-V (70|80|90)]  
[-S [<nom_serveur>[\<nom_instance>]]]  
[-U <compte_de_connexion>][-P <mot_de_passe>] [-T]  
[-q]  
[-d <nom_base>]  
[-a <taille_paquet_reseau_0>]  
[-h "<liste_option_supplementaires>"]
```

- -V permet d'indiquer la version de SQL Server (70 = 7, 80 = 2000 et 90 = 2005),
- -T authentification avec compte Windows
- -q active le paramètre de session QUOTED\_IDENTIFIERS à ON

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

<liste\_option\_supplementaires> ::= <option1> [, <option2> [, ...]]

<optionN> :

```
{ORDER (...)|  
|ROWS_PER_BATCH = <nombre_lignes>  
|KILOBYTES_PER_BATCH = <nombre_ko>  
|TABLOCK  
|CHECK_CONSTRAINTS  
|FIRE_TRIGGERS}
```

Exemple :

```
bcp.exe DB_TEST.dbo.T_DEPARTEMENT_DPT  
in C:\Temp\DepartementsIDF.txt  
-S "MSSQLSERVER" -T  
-t '!'; -r '\n' -F 2 C ACP -c -b 7  
-e C:\Temp\DepartementsIDF_error.log'  
-h "ORDER (DPT_CODE), CHECK_CONSTRAINTS, TABLOCK"
```

## 2 / Import / export de données

72

### BULK INSERT et bcp.exe – fichiers de format :

- Permet le mappage entre la source et la cible
- Fichiers de type texte ou XML

Peuvent être générées par bcp.exe pour un « premier jet » :

```
bcp.exe "MaTable" format nul  
-f "c:\temp\MonFichierFormat.txt"  
-S "MonServeur" -T -n [-x]
```

Utilisez `-x` pour un fichier de format XML

Administration SQL Server - CH 02 - gestion des espaces de stockage

V3.0 10/07/2017

Exemples de fichiers de format :

1) Texte :

```
10.0  
2  
1 SQLCHAR 2 3 "" 1 DPT_CODE French_BIN2  
2 SQLCHAR 2 36 "" 2 DPT_NOM French_BIN2
```

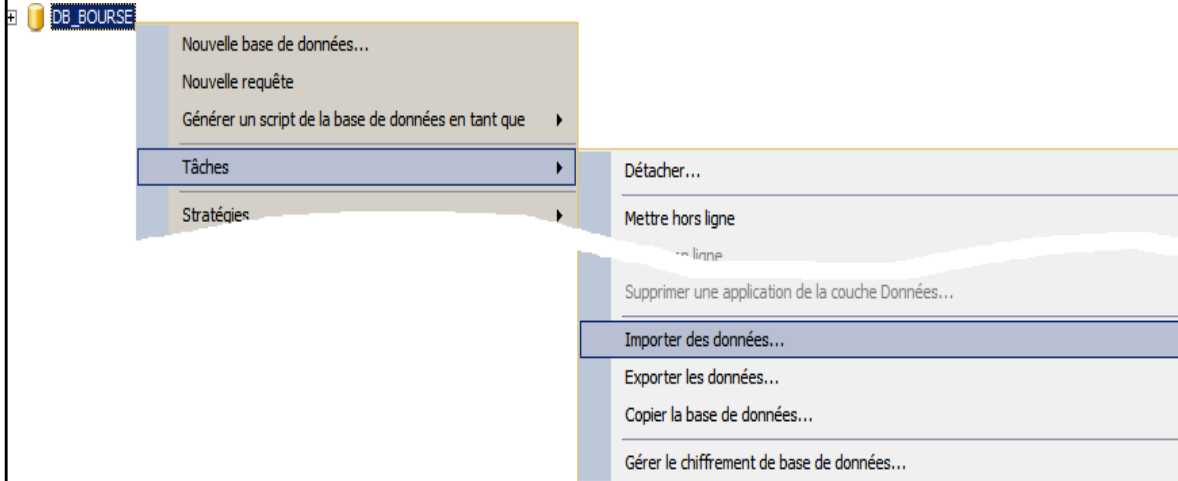
2) XML :

```
<?xml version="1.0"?>  
<BCPFORMAT xmlns="http://schemas.microsoft.com/sqlserver/2004/bulkload/format"  
xmlns:xsi="http://  
www.w3.org/2001/XMLSchema-instance">  
<RECORD>  
  <FIELD ID="1" xsi:type="CharPrefix" PREFIX_LENGTH="2" MAX_LENGTH="3"  
COLLATION="French_BIN2"/>  
  <FIELD ID="2" xsi:type="CharPrefix" PREFIX_LENGTH="2" MAX_LENGTH="36"  
COLLATION="French_BIN2"/>  
</RECORD>  
<ROW>  
  <COLUMN SOURCE="1" NAME="DPT_CODE" xsi:type="SQLCHAR"/>  
  <COLUMN SOURCE="2" NAME="DPT_NOM" xsi:type="SQLVARYCHAR"/>  
</ROW>  
</BCPFORMAT>
```

## 2 / Import / export de données

73

SSIS, peut être utilisé par l'assistant d'import/export



## 2 / Import / export de données

74

Assistant :

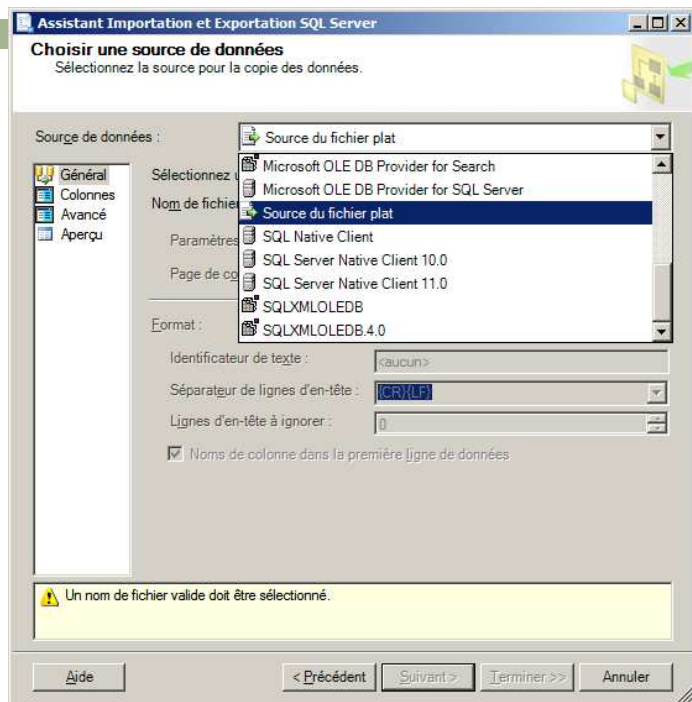




## 2 / Import / export de données

75

Laissez-vous  
guider !



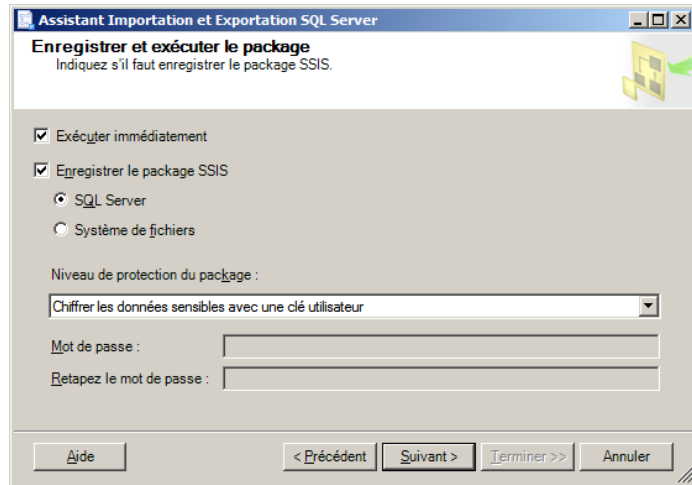
Administration SQL Server - CH 02 -

## 2 / Import / export de données

76

À la fin, vous pouvez enregistrer le paquetage d'import/export sous la forme d'un « Lot » SSIS, soit fichier, soit SQL...

... et vous en resservir !





# SQL SERVER 2012/2017 ADMINISTRATION

Frédéric Brouard (SQLpro) [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
Richard Gaillard - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)

Version 2.001  
Date 03/08/2015



**3** Sécurité et chiffrement

Administration SQL Server - CH 03 - Sécurité et chiffrement V3.0 10/07/2017

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

Illustration complémentaires Claude Leroy, Frédéric Brouard, Rudy Bruchez – Tous droits réservés

## 3 – Sécurité & chiffrement

3



- ❑ Sécurité externe
- ❑ Modèle de sécurité interne
- ❑ Authentification et connexion
- ❑ Utilisateurs SQL
- ❑ Rôles et privilèges
- ❑ « contained databases »
- ❑ Contexte de sécurité
- ❑ Chiffrement
- ❑ Audit de sécurité

## 3 / Sécurité

4

Risques encourus :

- Accès non désiré aux données :
  - Via le moteur SQL (requêtes « illégitimes »)
  - Via les fichiers de données
  - Via l'échange de données (sauvegardes, fichiers d'export)
- Interception des données sur le réseau
- Injections de code

## 3 / Sécurité

5

Injection de code SQL :

- Développez à l'aide de procédures stockées :
  - utilisez la procédure `sp_executesql` plutôt que `EXEC`
- Dans le code client :
  - vérifiez les variables (utilisez `QUOTENAME` si besoin)
  - recherchez certains motifs
  - nettoyez les chaînes de caractères
  - employez des contraintes `CHECK`
  - Évitez les fonctions « `OPEN` »
  - Ne renvoyez jamais le message d'erreur brut !

**Gérez correctement la sécurité SQL : utilisateurs et privilèges**

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Exemples de motifs à rechercher et éviter :

```
--  
/*  
*/  
EXEC  
EXECUTE  
CAST  
CONVERT  
\  
"  
;  
xp_  
WAITFOR
```

**Caractères autorisés dans un email :**

abcdefghijklmnopqrstuvwxyz0123456789@.-\_+ (en majuscule ou minuscule pour les lettres) => collation `CI_AS`

**Caractères valides pour un nom :**

abcdefghijklmnopqrstuvwxyz - ' (majuscule, minuscule, accents pour les lettres) => collation `CI_AI`

**Fonctions « OPEN » :** `OPENQUERY`, `OPENROWSET`, `OPEN`

## 3 / Sécurité

6

Attaques DoS (Denial Of Service - *déni de service*) :

- Mobilise les ressources du serveur SQL au détriment des autres utilisateurs

Principales fonctions sujettes :

- LIKE
- CONTAINS
- Expression régulières via UDF SQL CLR

Contrôle :

- `sp_configure 'query governor cost limit'`
- Gouverneur de ressource

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les opérateurs comme LIKE ou plus généralement les recherches à l'aide de motifs complexes (expressions régulières) sont les éléments les plus utilisés pour ce type d'attaques

Vous pouvez aussi auditer les connexions et rechercher les plus coûteuses (ou en mode web, les plus longues).



## 3 / Sécurité

7

Fichiers :

- Utilisez la sécurité NTFS / ReFS

Concerne :

- Les fichiers des bases (données, transactions, FileStream/FileTable)
- Les sauvegardes (.bak, .trn)
- Les exports de données (via BCP ou SSIS)

Évitez FAT32, bien qu'il soit possible de l'utiliser pour SQL Server

La réplication de données utilise des fichiers de sauvegarde ou d'export (*snapshot* de base ou de table)

Le log shipping utilise des fichiers de sauvegardes (de toutes natures).

## 3 / Sécurité

8

### Connexion :

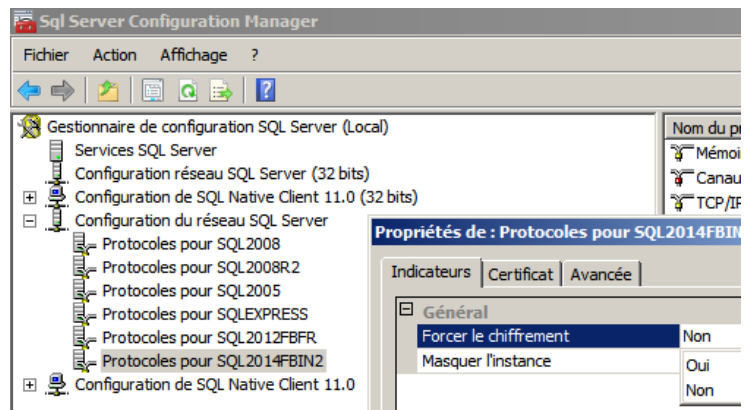
- Il est possible de chiffrer les flux entre clients et serveur :
- Dans SQL Server :
  - Propre à la connexion
  - Passe par des certificats SSL
- On peut aussi utiliser IPSec (cartes)

## 3 / Sécurité

9

chiffrer les flux entre clients et serveur...

Côté serveur :



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

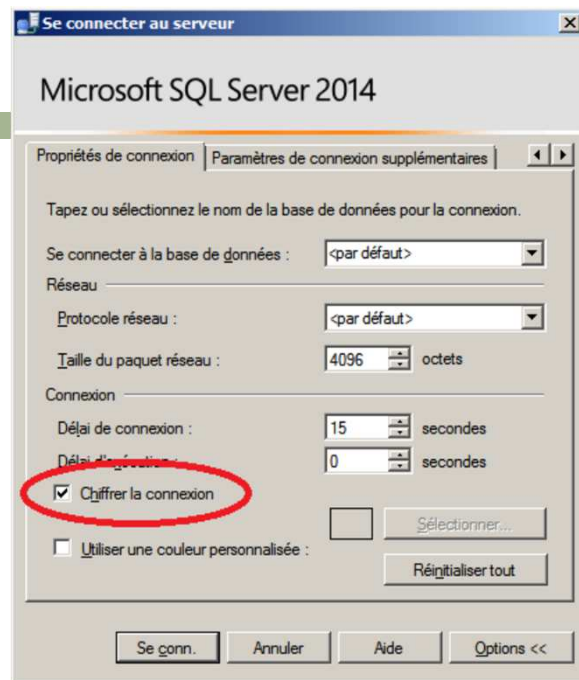
Illustration FB (copie d'écran)

## 3 / Sécurité

10

chiffrer ...

Côté client :



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Illustration FB (copie d'écran + dessin)

## 3 / Sécurité, authentification

11

Phase pendant laquelle un programme tente d'accéder à l'instance SQL

- En cas de succès : démarre une session
- En cas d'échec : message d'erreur

Mode d'authentification :

- Par compte Windows
- Par compte SQL

## 3 / Sécurité, authentification

12

Méthode d'authentification :

- Windows seulement
- Mixte (Windows + SQL)



```
EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE', N'Software\Microsoft\MSSQLServer\MSSQLServer', N'LoginMode', REG_DWORD, 1
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La paramétrage de l'authentification se fait dans les propriété du serveur, onglet « Sécurité »

Ce paramétrage modifie une clef de registre (LoginMode) de l'instance SQL Server courante;

1 => Windows seulement

2 => Windows et SQL

Il est nécessaire de redémarrer l'instance pour que ce paramétrage soit pris en compte.

Illustration FB (copie d'écran)

## 3 / Sécurité SQL, modèle

13

Du fait du multibase, le modèle de sécurité de SQL Serveur est à deux niveaux :

- Au niveau serveur : LOGIN (compte de connexion)
  - Sécurise des opérations de niveau « serveur » ou objet du serveur
- Au niveau des bases : USER (utilisateur SQL)
  - Sécurise des opérations de niveau base ou objets de la base

Un USER est associé à un LOGIN (sauf cas particulier)

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Pour savoir à quel niveau une opération est sécurisable, demandez vous s'il est possible de l'effectuer dans la base ou si elle porte sur un objet de la base.

Exemple :

- une opération de sauvegarde est faisable dans le contexte d'une base particulière et en étant dans la base => opération de niveau base de données
- Une opération de restauration n'est pas faisable dans le contexte de la base cible, car c'est en fait la création d'une nouvelle base => opération de niveau serveur

Un utilisateur SQL est au plus associé à un compte de connexion.

Il est néanmoins possible des utilisateurs SQL non associé à un compte de connexion (cas particulier) :

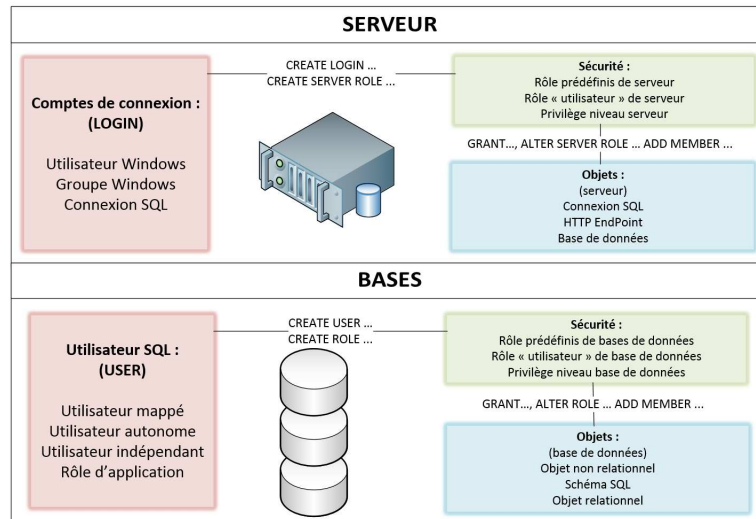
- Utilisateur SQL « orphelin »
- Utilisateur avec connexion directe à la base (sans passer par le niveau serveur)

Les comptes de connexion sont stockées dans la base master et sont accessible par la vue `sys.server_principals`

Les utilisateur SQL sont stockées dans les bases cibles et sont accessible par la vue `sys.database_principals`

## 3 / Sécurité SQL, modèle

14



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Principales commandes :

- Créer un compte de connexion : CREATE LOGIN...
- Créer un rôle de serveur : CREATE SERVER ROLE...
- Associer un compte de connexion à un rôle de serveur : ALTER SERVER ROLE ... ADD MEMBER ...
- Créer un utilisateur SQL dans une base de données : CREATE USER ...
- Créer un rôle de base de données : CREATE ROLE...
- Associer un rôle de base de données à un utilisateur : ALTER ROLE ... ADD MEMBER ...
- Donner un privilège : GRANT...
- Révoquer un privilège obtenu : REVOKE ...
- Interdire un privilège : DENY ...

Illustration FB



## 3 / Schéma SQL

15

Espace de « stockage » logique des objets relationnels de la base.

- Schéma par défaut de la base : dbo
- Schéma par défaut de l'utilisateur SQL défini lors de la création de l'utilisateur
  - À défaut, schéma SQL par défaut de la base (dbo)
- Indépendant de la notion de propriétaire (utilisateur SQL)

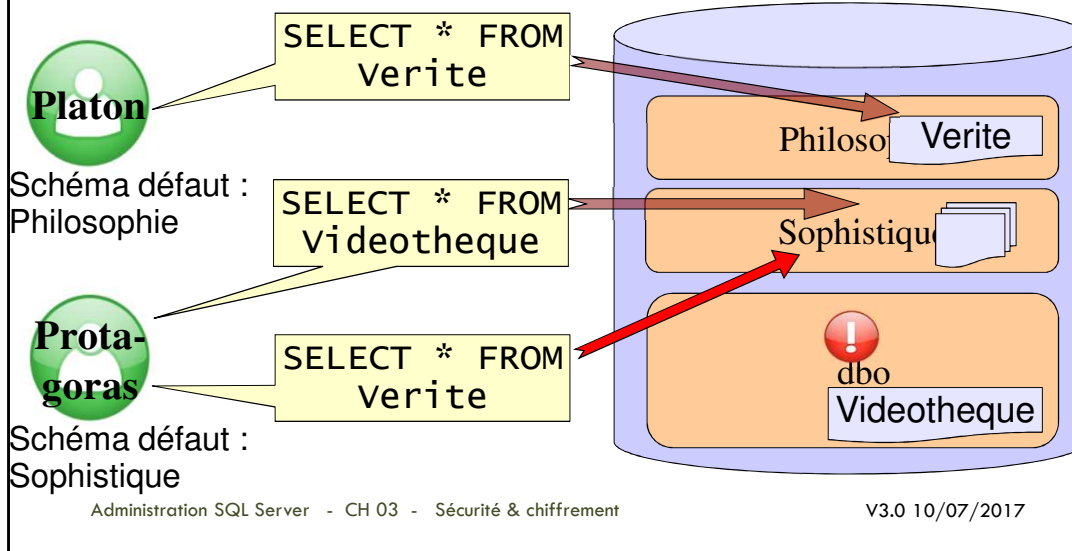
Un objet peut migrer de schéma via la commande ALTER SCHEMA ... TRANSFER

La notion de schéma dans SQL Server est conforme à la norme SQL. Elle diffère de celle d'Oracle dans laquelle propriétaire (c'est-à-dire l'utilisateur ayant créé l'objet) et schéma sont un seul et même objet. Dans SQL Server ces deux notions sont totalement disjointes et il est possible de transférer la propriété d'un schéma en utilisant la commande ALTER AUTHORIZATION (voir plus loin dans ce chapitre).

## 3 / Schéma SQL

16

Sans préfixe de schéma : phase de résolution de nom



Soyez clairs et explicites : préfixez toujours tous les objets par leur schéma (même si vous n'avez utilisé que dbo !)

Amélioration de performances...

- Pas de recherches à travers les schémas
- Pas de recompilation de procédures stockées

Une procédure doit parfois se recompiler selon l'utilisateur qui l'appelle, pour résoudre le nom des objets qu'elle contient

Illustration RB.

## 3 / Schéma SQL

17

Création, syntaxe :

```
CREATE SCHEMA nom_schema  
    [ AUTHORIZATION = nom_utilisateur ]
```

Transfert d'objet, syntaxe :

```
ALTER SCHEMA nom_schema  
    TRANSFER nom_objet
```

Suppression : DROP SCHEMA ...

Modification : ALTER SCHEMA ...

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Exemples :

**Création d'un nouveau schéma SQL dans la base contextuelle :**

```
CREATE SCHEMAS_FACTURATION  
    AUTHORIZATION = dbo;
```

**Transfer d'un objet d'un schéma SQL à l'autre :**

```
ALTER SCHEMA S_FACTURATION  
    TRANSFER dbo.T_FACTURE;
```

La table dbo.T\_FACTURE devient S\_FACTURATION.T\_FACTURE.

**ATTENTION** : le code des vues et routine restant inchangées, si cette table est mentionnée, il faudra revoir la définition des objets qui mentionne cette table.

## 3 / Compte de connexion

18

Compte « Windows » (utilisateur Windows ou groupe) ou SQL.

Permet de porter la sécurité sur les objets au niveau serveur via :

- les rôles prédéfinis de serveur;
- les rôles de serveur définis par l'utilisateur;
- les privilèges sur les objets ou conteneurs.

Lors de la création, sont activés par défaut et dotés des privilèges CONNECT et VIEW ANY DATABASE.

Les informations sur les comptes de connexion comme sur les rôles de serveur sont visibles dans la vue `sys.server_principals`.

Les informations sur l'appartenance d'un compte à un rôle de serveur sont liées dans la vue `sys.server_role_members`.

## 3 / Compte de connexion

19

Compte « Windows » (utilisateur Windows ou groupe)

Syntaxe de création :

```
CREATE LOGIN nom_compte_windows
  FROM WINDOWS
  WITH <options>
```

```
<options> ::= <options1> [ , <option2> ]
```

```
<optionN> ::= DEFAULT_DATABASE = base_cible
  | DEFAULT_LANGUAGE = langue
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Le compte Windows spécifié doit être de la forme `Domaine\Utilisateur` et spécifié entre crochets.

Il doit préalablement exister.

On peut préciser une base de donnée cible pour l'utilisateur (ceci exécute un USE `<base_cible>` après l'authentification).

Le paramètre « language » permet de configurer la session pour une langue spécifique. À défaut c'est la langue d'installation du serveur.

Ce paramètre peut être spécifié en langue locale ou en anglais et doit correspondre à l'une des 2 informations suivantes : colonne « name » ou « alias » de la table `sys.syslanguages`.

## 3 / Compte de connexion

20

Création compte « Windows », exemple :

```
CREATE LOGIN [HPZFRED\postgres]
  FROM WINDOWS
  WITH DEFAULT_DATABASE = DB_MEDICACT,
  DEFAULT_LANGUAGE = us_english
```

Modifications d'un compte :

- Changement de langue
- Changement de base cible
- Désactivation / Réactivation

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

### Syntaxes des changements :

Changement de langue et/ou de base cible :

```
ALTER LOGIN <nom_connexion> WITH ...
```

- DEFAULT\_DATABASE = <nom\_base\_cible>
- DEFAULT\_LANGUAGE = <identifiant\_langue>

Désactivation, réactivation :

```
ALTER LOGIN <nom_connexion> { DISABLE | ENABLE }
```

Suppression d'un compte :

```
DROP LOGIN <nom_connexion>
```

## 3 / Compte de connexion

21

### Création compte SQL, syntaxe :

```
CREATE LOGIN login_name
WITH PASSWORD = { 'pass' | h_pass HASHED } [ MUST_CHANGE ]
    [, <options> ]

<options> ::= <option1> [ <option2 [ , ... ] ]

<optionN> ::=
    { SID = sid
    | DEFAULT_DATABASE = base_cible
    | DEFAULT_LANGUAGE = langue
    | CHECK_EXPIRATION = { ON | OFF }
    | CHECK_POLICY = { ON | OFF }
    | CREDENTIAL = nom_accréditation }
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

### Paramètres et options :

- *pass* : mot de passe relatif au compte de connexion SQL
- *h\_pass* : valeur binaire (chaines hexadécimale) du hachage du mot de passe
- MUST\_CHANGE : indique que l'utilisateur devra changer le mot de passe à la première connexion.
- SID : GUID relatif au compte (généré par défaut)
- CHECK\_EXPIRATION = vérifie ou non la politique d'expiration des mots de passe mise en œuvre au niveau système
- CHECK\_POLICY = vérifie ou non la politique de complexité des mots de passe mise en œuvre au niveau système
- CREDENTIAL = accréditation liée à ce compte de connexion

## 3 / Compte de connexion

22

Création compte SQL, exemple :

```
CREATE LOGIN stagiaire
  WITH PASSWORD = 'Stage2015!'
  DEFAULT_DATABASE = SNAP_prod20160215
  DEFAULT_LANGUAGE = Français;
```

Désactivation : ALTER LOGIN ... DISABLE

Suppression : DROP LOGIN ...

Modification : ALTER LOGIN ...

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

**Modification au niveau mot de passe :**

```
ALTER LOGIN nom_connexion
  WITH { PASSWORD = 'pass' | h_pass HASHED }
  OLD_PASSWORD = 'ancien_pass'
  [ <options_pass> ]
```

<options\_pass> ::= <option\_pass1> [ , <option\_pass2> ]

<option\_passN> ::= { MUST\_CHANGE | UNLOCK }

**Modification de paramétrage :**

```
ALTER LOGIN nom_connexion
  WITH <options>
```

<options> ::= <option1> [ , <option2> [ , ... ] ]

```
<optionN> ::=
{ DEFAULT_DATABASE = nom_base
| DEFAULT_LANGUAGE = langue
| NAME = nom_connexion
| CHECK_POLICY = { ON | OFF }
| CHECK_EXPIRATION = { ON | OFF }
| CREDENTIAL = nom_accréditation
| NO CREDENTIAL }
```

**Modification accréditation :**

```
ALTER LOGIN nom_connexion
  WITH { ADD CREDENTIAL nom_accréditation
  | DROP CREDENTIAL nom_accréditation }
```



## 3 / Utilisateur SQL

23

Dans le contexte d'une base de données...

Sert à porter la sécurité sur la base et ses objets,  
via :

- les rôles prédéfinis de base de données;
- les rôles de base de données définis par l'utilisateur;
- les privilèges sur les objets ou les conteneurs de la base.

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les informations sur les utilisateurs SQL comme sur les rôles de base de données sont visibles dans la vue `sys.database_principals`.

Les informations sur l'appartenance d'un utilisateur à un rôle de base de données sont liées dans la vue `sys.database_role_members`

## 3 / Utilisateur SQL

24

### Utilisateurs particuliers :

- **dbo** : utilisateur particulier propriétaire du schéma par défaut de même nom. Possède tous les privilèges (db\_owner)
- **sys, INFORMATION\_SCHEMA** : utilisateurs particuliers propriétaires des schémas de mêmes noms. Ne peuvent être utilisés. Servent à Microsoft pour les objets « systèmes »
- **guest** : utilisateur par défaut d'utilisateur

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Tout compte de connexion doté du privilège CONTROL SERVER ou du rôle sysadmin est de fait utilisateur dbo dans chacune des bases (cas général pour le compte de connexion sa).

Les objets systèmes sont repérables dans les vues du schéma sys, par le fait que la colonne booléenne is\_ms\_shipped (est livré par MS) à la valeur 1.

guest est généralement utilisé dans les traitements multibase lorsqu'un compte de connexion n'est pas recensé comme utilisateur SQL dans une base dans laquelle il souhaite travailler. Il emprunte alors l'identité de guest comme utilisateur SQL et bénéficie de tous les privilèges qui lui sont affectés. Par défaut guest ne possède aucun privilège.

Exemple : lire des informations dans une base ayant des données communes tels que les codes postaux.

## 3 / Utilisateur SQL

25

Utilisateur SQL lié à un compte de connexion.

Syntaxe :

```
CREATE USER nom_utilisateur  
  { FOR | FROM } LOGIN compte_de_connexion  
  [ WITH DEFAULT_SCHEMA = nom_schema ]
```

À défaut de précision du schéma SQL, le schéma par défaut de l'utilisateur est celui de la base de données.

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Le schéma par défaut d'une base SQL Server est dbo.

**CONSEIL** : nous vous recommandons de ne pas nommer de façon identique les comptes de connexion et les utilisateurs de base de données car cela entraîne une certaine confusion dans l'attribution des privilèges. Hélas c'est ce que fait l'IHM de SSMS par défaut...

## 3 / Utilisateur SQL

26

Utilisateur SQL autonome.

Deux cas de figure :

- utilisateur sans compte de connexion;
- utilisateur se connectant directement à la base;

Pour ce dernier cas, la base doit être paramétrée en « CONTAINED ».

La notion de CONTAINED DATABASE est apparue avec la version 2012 (voir plus loin).

## 3 / Utilisateur SQL

27

Utilisateur SQL se connectant directement à la base.

Syntaxe :

```
CREATE USER { compte_windows [ WITH <options> ]  
            | nom_utilisateur  
            WITH PASSWORD = 'pass'  
            [ [ SID = sid ] , <options> ] }
```

```
<options> ::= <options1> [ , <options2> [ , ... ] ]
```

```
<optionN> ::= {DEFAULT_SCHEMA = nom_schema  
              | DEFAULT_LANGUAGE = langue }
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les « CONTAINED DATABASE » permettent pour les versions 2012 et 2014, le niveau « PARTIAL » (voir plus loin).

Une « CONTAINED DATABASE » permet de créer une base autonome et donc de se connecter directement à la base sans passer par le niveau serveur.

## 3 / Utilisateur SQL

28

Utilisateur SQL sans compte de connexion.

Syntaxe :

```
CREATE USER nom_utilisateur
  { WITHOUT LOGIN [ WITH DEFAULT_SCHEMA = nom_schema ]
  | { FOR | FROM } CERTIFICATE nom_certificat
  | { FOR | FROM } ASYMMETRIC KEY nom_clef_asymetrique
  }
```

De tels utilisateurs sont des cas particuliers nécessités par des besoins techniques spécifiques.

## 3 / Utilisateur SQL

29

### Utilisateur SQL, exemples :

```
USE DB_BOURSE;
```

```
CREATE USER USR_COTATION_CAC40  
  FROM LOGIN [cic.fr\traders]  
  WITH DEFAULT_SCHEMA = S_CAC40;
```

```
CREATE USER USR_COTATION_IBEX40  
  WITH PASSWORD = '-IBEX 40!pt',  
  DEFAULT_SCHEMA = S_TITRE,  
  DEFAULT_LANGUAGE = Portuguais;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Ces deux exemples sont lancés dans le contexte de la base DB\_BOURSE.

- Le premier exemple créé un utilisateur (USR\_COTATION\_CAC40) lié au compte de connexion Windows (cic.fr\traders) créé préalablement et lui assigne le schéma S\_CAC40 par défaut.
- Le second exemple créé un utilisateur (USR\_COTATION\_IBEX40) qui se connecte directement à la base avec le mot de passe -IBEX 40!pt dont le schéma par défaut est S\_TITRE et la langue par défaut est le portugais.

## 3 / Utilisateur SQL

30

Utilisateur SQL, modifications :

Suppression : DROP USER ...

Modification : ALTER USER ...

Options :

- NAME = nouveau\_nom
- DEFAULT\_SCHEMA = { nom\_schema | NULL }
- LOGIN = nom\_connexion
- PASSWORD = 'pass' [ OLD\_PASSWORD = '...' ]
- DEFAULT\_LANGUAGE = langue

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

L'option PASSWORD n'est possible que pour un utilisateur SQL se connectant dans une base « CONTAINED ».



## 3 / Sécurité niveau serveur

31

Concerne les comptes de connexion.

Attribution fine via la gestion de privilèges :

```
GRANT <liste_privileges>
  ON <cible>
  TO <entités_sécurité_serveur>

<liste_privileges> ::=
  <privilege1> [ , <privilege2> [ , ... ] ]

<cible> ::=
  { <nom_classe_conteneur>::<nom_conteneur>
  | <nom_objet> }
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les privilèges peuvent être attribués à un conteneur et dans ce cas, se propagent à tous les objets contenus. Les conteneurs sont :

- LOGIN
- ENDPOINT
- SERVER

Pour ce dernier (serveur), on peut omettre la clause cible (ON) car c'est toujours l'instance contextuelle.

On peut octroyer des privilèges à plusieurs entités de sécurité de niveau serveur simultanément avec la même commande. Les entités de sécurité de niveau serveur sont :

- Les comptes de connexion (LOGIN)
- Les rôles (de serveur)

Un extension possible de cette commande est la clause :  
WITH GRANT OPTION

Qui permet de transmettre les privilèges acquis. Nous vous déconseillons fortement l'usage de cette clause car elle entraîne un chaînage arborescent des privilèges difficile à contrôler.

En principe les privilèges sont accordés par le compte de connexion courant (celui qui lance la commande GRANT) et qui est appelé GRANTOR envers un compte de connexion appelé GRANTEE. Il est possible de spécifier tout autre GRANTOR via l'option AS.

## 3 / Sécurité niveau serveur

32

Liste des privilèges possible de niveau serveur :

```
SELECT * FROM sys.fn_builtin_permissions(NULL)
WHERE class_desc IN ('SERVER', 'ENDPOINT',
                    'AVAILABILITY GROUP',
                    'LOGIN', 'SERVER ROLE')
ORDER BY CASE WHEN class_desc = 'SERVER'
              THEN '' ELSE class_desc END
```

| class_desc         | permission_name        | type | covering_permission_name | parent_class_desc | parent_covering_permission_name |
|--------------------|------------------------|------|--------------------------|-------------------|---------------------------------|
| SERVER             | IMPERSONATE ANY LO...  | IAL  | CONTROL SERVER           |                   |                                 |
| SERVER             | SELECT ALL USER SEC... | SUS  | CONTROL SERVER           |                   |                                 |
| SERVER             | CONTROL SERVER         | CL   |                          |                   |                                 |
| AVAILABILITY GROUP | VIEW DEFINITION        | VW   | CONTROL                  | SERVER            | VIEW ANY DEFINITION             |
| AVAILABILITY GROUP | ALTER                  | AL   | CONTROL                  | SERVER            | ALTER ANY AVAILABILITY GROUP    |
| AVAILABILITY GROUP | TAKE OWNERSHIP         | TO   | CONTROL                  | SERVER            | CONTROL SERVER                  |

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La fonction table `sys.fn_builtin_permissions` ne possède qu'un seul argument filtrant le niveau de permission (colonne `class_desc`) et renvoie la hiérarchie des privilèges du niveau concerné ou a défaut de tous les niveaux.

Colonnes de cette table :

- `class_desc` : classe de l'entité de sécurité cible
- `permission_name` : nom du privilège
- `type` : code du type de privilège
- `covering_permission_name` : nom du privilège du groupe impliquant les autres privilèges pour cette classe.
- `parent_class_desc` : classe de l'entité de sécurité parente
- `parent_covering_permission_name` : nom du privilège parent du groupe impliquant les autres privilèges pour cette classe.

## 3 / Sécurité niveau serveur

33

GRANT, exemple :

```
GRANT ALTER TRACE ,  
      ALTER ANY EVENT NOTIFICATION ,  
      ALTER ANY SERVER AUDIT ,  
      ALTER ANY EVENT SESSION ,  
      VIEW SERVER STATE  
ON SERVER : : [HPZFRED\SQL2014FBIN2]  
TO CNX_AUDIT ;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Pour cet exemple, notez que la clause ON est inutile car la cible ne peut jamais être autre que le serveur courant.

Du fait de la hiérarchie des privilèges, cela a donné au compte de connexion CNX\_AUDIT, les privilèges suivants :

- ALTER TRACE,
- ALTER ANY EVENT NOTIFICATION,
  - CREATE DDL EVENT NOTIFICATION
  - CREATE TRACE EVENT NOTIFICATION
- ALTER ANY SERVER AUDIT,
- ALTER ANY EVENT SESSION,
- VIEW SERVER STATE

## 3 / Sécurité niveau serveur

34

REVOKE :

Commande permettant de révoquer (supprimer) un privilège préalablement attribué

DENY :

Permet d'interdire l'attribution d'un privilège

**ATTENTION** : DENY prioritaire !



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

N'utilisez le DENY qu'avec parcimonie. Sinon cela peut se traduire par des blocages du fait de la superposition des privilèges.

La révocation d'un privilège non attribué, comme l'application successive du même privilège au même destinataire ne provoque aucune erreur et n'a aucun effet.

## 3 / Sécurité niveau serveur

35

REVOKE, DENY, exemple :

```
REVOKE VIEW SERVER STATE,  
VIEW ANY DATABASE  
TO CNX_AUDIT ;
```

```
DENY SHUTDOWN  
TO CNX_AUDIT ;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La première commande (REVOKE) retire le privilège préalablement établis, de lire les informations d'état du serveur.

La seconde commande (DENY) interdit la pose du privilège d'arrêt du serveur. Notez que ces deux commandes porte sur la cible serveur (clause ON omise)

## 3 / Rôles de serveur

36

- Collection de privilèges de niveau serveur
- 2 types :
  - Prédéfinis (Microsoft)
  - Personnalisés (à définir), syntaxe de création :  

```
CREATE SERVER ROLE nom_role_serveur  
[ AUTHORIZATION nom_connexion ]
```
  - public (norme SQL), rôle dont les privilèges son hérités par tout les comptes de connexion

Les rôles de niveau serveur personnalisés (à définir par les dba et développeurs) sont apparus avec la version 2012.

## 3 / Rôles de serveur

37

Gestion globale via les rôles.

Ajout d'un rôle à un compte de connexion :

```
ALTER SERVER ROLE nom_role_serveur  
    ADD MEMBER nom_connexion
```

Retrait d'un rôle associé au compte de connexion :

```
ALTER SERVER ROLE nom_role_serveur  
    DROP MEMBER nom_connexion
```

Les procédures `sp_addsrvrolemember`, `sp_dropsrvrolemember` sont obsolètes.

## 3 / Rôles prédéfinis de serveur

38

Non modifiables, au nombre de 8 :

|                 |               |
|-----------------|---------------|
| <b>sysadmin</b> | processadmin  |
| serveradmin     | securityadmin |
| setupadmin      | bulkadmin     |
| diskadmin       | dbcreator     |

**sysadmin** possède tous les privilèges.

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

sysadmin possède le privilège CONTROL SERVER (équivalent)

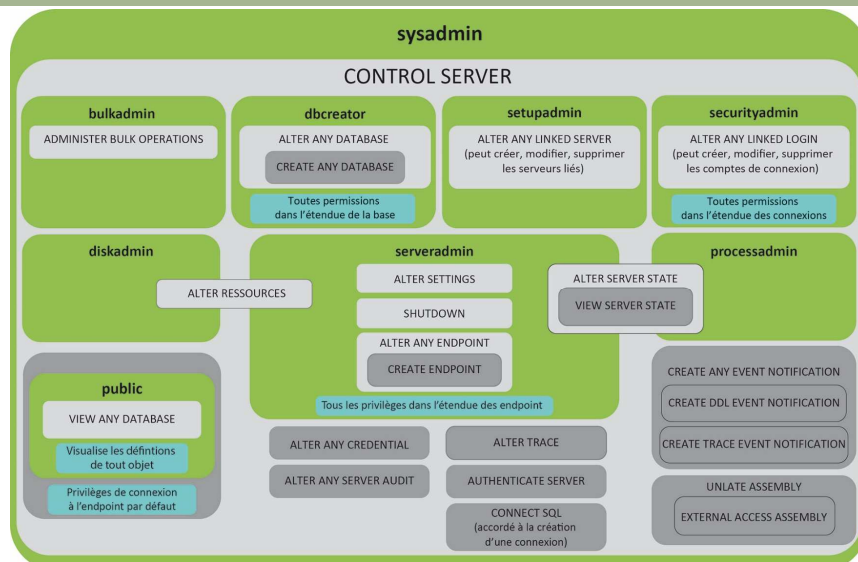
bulkadmin est un rôle permettant d'utiliser la commande BULK INSERT.

À l'exception du cas particulier du BULK INSERT réglé par le rôle bulkadmin et de la commande de sauvegardes (BACKUP) réglé par le compte de service de l'instance, seul les membres du rôle sysadmin peuvent lancer des commandes ayant des impacts en dehors du serveur, tels que la création de fichiers.



## 3 / Rôles prédéfinis de serveur

39



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Ce tableau synoptique montre tous les rôles prédéfinis de serveur et les privilèges qui leur sont associés.

Public n'est pas à proprement parler un rôle, mais en fait un super utilisateur dont les privilèges sont hérités par tous les autres utilisateurs (norme SQL). Ils se comporte en fait comme un rôle auquel tout utilisateur est affecté. Minimisez l'utilisation de public.

Illustration CL.

## 3 / Rôles personnalisés de serveur

40

Exemple :

```
CREATE SERVER ROLE SR_SECUR;  
  
GRANT ALTER ANY LOGIN,  
        ALTER ANY ENDPOINT,  
        ALTER ANY SERVER ROLE  
  TO SR_SECUR;  
  
ALTER SERVER ROLE SR_SECUR  
  ADD MEMBER CNX_SECUR;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La première commande (CREATE SERVER ROLE) crée un rôle personnalisé de serveur de nom SR\_SECUR.

La seconde commande (GRANT) attribue des privilèges ciblant le serveur au rôle SR\_SECUR.

La troisième commande (ALTER SERVER ROLE) enrôle le compte de connexion CNX\_SECUR en tant que membre du rôle SR\_SECUR.

Par conséquent le compte de connexion CNX\_SECUR hérite des privilèges ALTER ANY LOGIN, ALTER ANY ENDPOINT et ALTER ANY SERVER ROLE, ainsi que de tous ceux qui seront ajoutés à l'avenir.

## 3 / Métadonnées de sécurité serveur

41

| Vue                          | Description                                       |
|------------------------------|---|
| sys.server_principals        | Entité de sécurité niveau serveur                 |
| sys.server_role_members      | Liste des connexions membres d'un rôle de serveur |
| sys.server_permissions       | Liste des privilèges affectés                     |
| sys.sql_logins               | liste des comptes de connexion SQL                |
| sys.fn_builtin_permissions() | Liste de tous privilèges (niveau serveur et base) |

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Exemples :

### liste des comptes de connexion Windows (utilisateur ou groupes Windows) :

```
SELECT *
FROM sys.server_principals
WHERE type_desc LIKE 'WINDOWS?_%' ESCAPE '?';
```

### compte de connexion et rôle associé :

```
SELECT c.name AS CONNEXION, r.name AS ROLE_DE_SERVEUR
FROM sys.server_principals AS c
JOIN sys.server_role_members AS rm
ON c.principal_id = rm.member_principal_id
JOIN sys.server_principals AS r
ON rm.role_principal_id = r.principal_id;
```

### reconstitution de l'affectation de privilèges au serveur

```
SELECT CASE p.state WHEN 'W' THEN 'GRANT' ELSE state_desc END + ' '
+ permission_name + ' ON SERVER::[' + @@SERVERNAME + '] TO ['
+ GEE.name COLLATE Latin1_General_CI_AS_KS_WS + ']'
+ CASE p.state WHEN 'W' THEN 'WITH GRANT OPTION' ELSE '' END
+ 'AS [' + GOR.name COLLATE Latin1_General_CI_AS_KS_WS + '];'
FROM sys.server_permissions AS p
JOIN sys.server_principals AS GOR
ON p.grantor_principal_id = GOR.principal_id
JOIN sys.server_principals AS GEE
ON p.grantee_principal_id = GEE.principal_id
WHERE class_desc = 'SERVER'
```

## 3 / Sécurité niveau base

42

Concerne les utilisateur SQL de la base contextuelle.

Attribution fine via la gestion de privilèges :

```
GRANT <liste_privileges>
    ON <cible>
    TO <entités_sécurité_base>

<liste_privileges> ::=
    <privilege1> [ , <privilege2> [ , ... ] ]

<cible> ::=
    { <nom_classe_conteneur>::<nom_conteneur>
    | <nom_objet> }
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les privilèges peuvent être attribués à un conteneur (base de données ou schéma) et dans ce cas, se propagent à tous les objets contenus quelque soit le niveau du conteneur.

Les classes de conteneur sont :

- SCHEMA
- DATABASE

Pour cette dernière (base de donnée), on peut omettre la clause cible (ON) car c'est toujours la base contextuelle.

On peut octroyer des privilèges à plusieurs entités de sécurité de niveau base simultanément avec la même commande. Les entités de sécurité de niveau base sont :

- Utilisateur SQL
- Rôles (de base de données ou d'application)

Un extension possible de cette commande est la clause :

WITH GRANT OPTION

Qui permet de rétrocéder les privilèges acquis. Nous vous déconseillons fortement l'usage de cette clause car elle entraîne un chaînage arborescent des privilèges difficile à contrôler.

En principe les privilèges sont accordés par le compte de connexion courant (celui qui lance la commande GRANT) et qui est appelé GRANTOR envers un compte de connexion appelé GRANTEE. Il est possible de spécifier tout autre GRANTOR via l'option AS.

## 3 / Sécurité niveau base

43

Liste des privilèges de niveau base de données :

```
SELECT * FROM sys.fn_builtin_permissions(NULL)
WHERE class_desc NOT IN ('SERVER', 'ENDPOINT',
                        'AVAILABILITY GROUP',
                        'LOGIN', 'SERVER ROLE')
ORDER BY CASE WHEN class_desc = 'DATABASE'
              THEN '' ELSE class_desc END
```

| class_desc | permission_name  | type  | covering_permission_name | parent_class_desc | parent_covering_permi |
|------------|------------------|-------|--------------------------|-------------------|-----------------------|
| DATABASE   | CREATE TABLE     | CRTB  | ALTER                    | SERVER            | CONTROL SERVER        |
| DATABASE   | CREATE VIEW      | CRVW  | ALTER                    | SERVER            | CONTROL SERVER        |
| DATABASE   | CREATE PROCE...  | CRPR  | ALTER                    | SERVER            | CONTROL SERVER        |
| DATABASE   | CREATE FUNCTI... | CRFN  | ALTER                    | SERVER            | CONTROL SERVER        |
| DATABASE   | CREATE RULE      | CRRU  | ALTER                    | SERVER            | CONTROL SERVER        |
| DATABASE   | CREATE DEFAULT   | CRDF  | ALTER                    | SERVER            | CONTROL SERVER        |
| DATABASE   | BACKUP DATAR     | BA DR | CONTROL                  | SERVER            | CONTROL SERVER        |

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La fonction table `sys.fn_builtin_permissions` ne possède qu'un seul argument filtrant le niveau de permission (colonne `class_desc`) et renvoie la hiérarchie des privilèges du niveau concerné ou a défaut de tous les niveaux.

Colonnes de cette table :

- `class_desc` : classe de l'entité de sécurité cible
- `permission_name` : nom du privilège
- `type` : code du type de privilège
- `covering_permission_name` : nom du privilège du groupe impliquant les autres privilèges pour cette classe.
- `parent_class_desc` : classe de l'entité de sécurité parente
- `parent_covering_permission_name` : nom du privilège parent du groupe impliquant les autres privilèges pour cette classe.

## 3 / Sécurité niveau base

44

GRANT, exemples :

```
GRANT SELECT TO USR_STAGE , USR_TRADER ;
```

```
GRANT INSERT , UPDATE , DELETE  
ON SCHEMA : : S_OPERATION  
TO USR_TRADER ;
```

```
GRANT EXECUTE ON dbo.P_AUDIT_DML  
TO USR_TRADER ;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Un tel script doit être lancé dans la base cible. Utiliser USE pour ce faire.

La première commande donne le privilège de lecture (SELECT) sur toutes les tables, vues et fonctions table opérationnelles de la base courante (et pas sur les vues systèmes) aux utilisateurs SQL USR\_STAGE et USR\_TRADER.

La seconde commande attribue les privilèges de mise à jour (INSERT, UPDATE, DELETE) sur toutes les tables, vues et fonctions table opérationnelles du schéma SQL S\_OPERATION, à l'utilisateur USR\_TRADER.

La troisième commande octroie le privilège d'exécution de la procédure dbo.P\_AUDIT\_DML à l'utilisateur USR\_TRADER.

## 3 / Sécurité niveau base

45

GRANT, particularité :

Vous pouvez limiter les privilèges :

- SELECT
- UPDATE

À certaines colonnes

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Exemple :

```
GRANT SELECT (MDC_CODE, MDC_NOM, MDC_MOLECULE)
ON S_ACT.T_MEDICAMENT_MDC
TO USR_PHARMA;
```

## 3 / Sécurité niveau base

46

DENY, REVOKE, exemples :

```
REVOKE DELETE
```

```
ON SCHEMA::S_OPERATION
```

```
TO USR_TRADER;
```

```
DENY VIEW DEFINITION
```

```
TO USR_TRADER;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La première commande révoque le privilège de suppression (DELETE) précédemment obtenu sur le schéma SQL S\_OPERATION, à l'utilisateur USR\_TRADER.

La seconde commande interdit le privilège de lecture des définitions des objets de toute la base à l'utilisateur USR\_TRADER.



## 3 / Rôles de base de données

47

- Collection de privilèges de niveau base de données
- 2 types :
  - Prédéfinis (Microsoft)
  - Personnalisés (à définir), syntaxe de création :  
`CREATE ROLE nom_role_bd`  
`[ AUTHORIZATION nom_utilisateur_sql ]`
  - public (norme SQL), rôle dont les privilèges sont hérités par tout les comptes de connexion

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les procédures `sp_addrole`, `sp_droprole` sont obsolètes.

## 3 / Rôles de base de données

48

Gestion globale via les rôles.

Ajout d'un rôle à un utilisateur SQL :

```
ALTER ROLE nom_role_bd  
    ADD MEMBER nom_utilisateur_sql
```

Retrait d'un rôle associé à un utilisateur SQL :

```
ALTER ROLE nom_role_bd  
    DROP MEMBER nom_utilisateur_sql
```

Les procédures `sp_addrolemember` et `sp_droprolemember` sont obsolètes.

## 3 / Rôles prédéfinis de BD

49

Non modifiables, au nombre de 9 :

**db\_owner**

db\_datareader

db\_datawriter

db\_denydatareader

db\_denydatawriter

db\_securityadmin

db\_accessadmin

db\_ddladmin

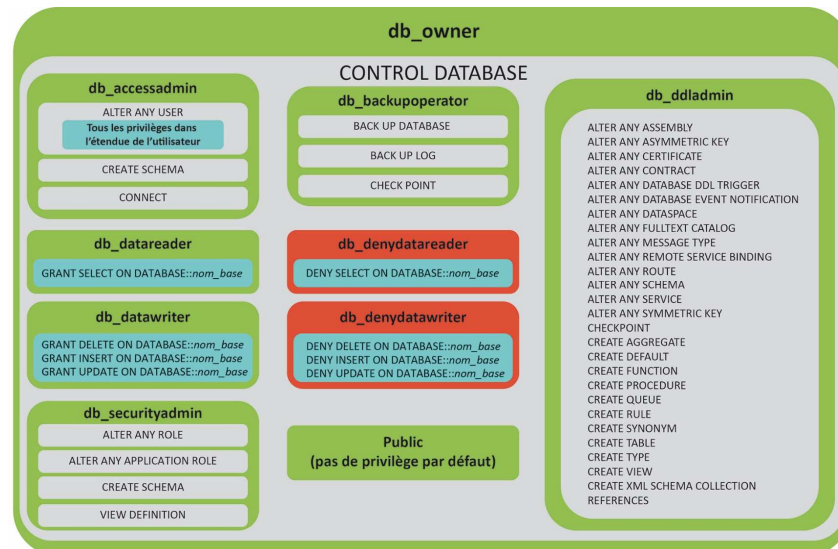
db\_backupoperator

db\_owner possède tous les privilèges dans la base.

db\_owner possède le privilège CONTROL DATABASE (équivalent)

## 3 / Rôles prédéfinis de BD

50



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Ce tableau synoptique montre tous les rôles prédéfinis de base de données et les privilèges qui leur sont associés.

Public n'est pas à proprement parler un rôle, mais en fait un super utilisateur dont les privilèges sont hérités par tous les autres utilisateurs (norme SQL). Ils se comporte en fait comme un rôle auquel tout utilisateur est affecté. Minimisez l'utilisation de public.

Illustration CL.

## 3 / Rôles personnalisés de BD

51

Exemple :

```
CREATE ROLE RDB_STAGIAIRE ;

GRANT SELECT TO RDB_STAGIAIRE ;
DENY SELECT ON SCHEMA::S_CLIENT
    TO RDB_STAGIAIRE ;
GRANT EXECUTE ON SCHEMA::S_REPORT
    TO RDB_STAGIAIRE ;

ALTER ROLE RDB_STAGIAIRE
    ADD MEMBER USR_STAGIAIRE ;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Un tel script doit être lancé dans la base cible. Utiliser USE pour ce faire.

La première commande (CREATE ROLE) crée le rôle RDB\_STAGIAIRE.

Les trois commandes suivantes octroient des privilèges au rôle RDB\_STAGIAIRE : le SELECT sur toute la base sauf le schéma S\_CLIENT (DENY) et l'EXECUTE sur le schéma S\_REPORT.

La dernière commande affecte le rôle RDB\_STAGIAIRE à l'utilisateur USR\_STAGIAIRE.

ATTENTION : les procédures sp\_addrole, sp\_droprole, sp\_addrolemember sont obsolètes.

## 3 / Bases autonomes

52

Les « CONTAINED DATABASE » pompeusement appelées *bases de données à relation contenant-contenu* (!) permettent de rendre autonome l'accès à la base :

- Plus de passage imposé par le compte de connexion
- Règle le problème de collation avec tempdb
- Mais interdit toute interaction avec l'extérieur...

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Le problème de la collation vient du fait que si une base A a été créée avec une collation X et que l'instance a été installée avec la collation Y, la création d'une table temporaire repose sur la collation de tempdb même si elle est créée dans le contexte de la base A, à moins de spécifier pour chaque colonne littérale la collation générique *database\_default*, ou de créer cette table via la commande `SELECT ... INTO`.

Au contraire, pour une base autonome B, la collation des littéraux d'une table temporaire créé dans le contexte de la base B est bien celle de la base B.

## 3 / Bases autonomes

53

Pour autoriser la fonctionnalité de bases « autonomes » dans l'instance :

```
EXEC sp_configure 'contained database authentication', 1;  
GO  
RECONFIGURE;
```

Pour faire qu'une base soit « autonome » :

```
ALTER DATABASE CURRENT SET CONTAINMENT = PARTIAL
```

Seul le niveau « partiel » est actuellement supporté.

Dans le futur les bases « autonomes » seront plus autonomes encore (niveau FULL) la principale difficulté étant l'intégration de l'Agent SQL dont les données sont dans la base système msdb, au sein de la base courante, une autre étant les messages d'erreur personnalisés créés via `sp_add_message` et stockés dans la table `sys.messages` de master.

## 3 / Bases autonomes

54

Pour vérifier qu'une base n'a pas d'interaction avec l'extérieur, utilisez la vue :

- `sys.dm_db_uncontained_entities`

Pour migrer un utilisateur SQL ayant un compte de connexion à un utilisateur se connectant directement à la base, utilisez la procédure :

- `sp_migrate_user_to_contained`

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Syntaxe :

EXEC sp\_migrate\_user\_to\_contained

[ @username = ] N'user' ,

[ @rename = ] { N'copy\_login\_name' | N'keep\_name' } ,

[ @disablelogin = ] { N'disable\_login' | N'do\_not\_disable\_login' }



## 3 / Fonctions de sécurité

55

Les fonctions suivantes peuvent vous être utiles pour tester vos privilèges :

- IS\_ROLEMEMBER(nom\_role\_bd)
- IS\_SRVROLEMEMBER(nom\_role\_server)
- HAS\_PERMS\_BY\_NAME(objet, nature, privilège)
- HAS\_DBACCESS(nom\_base)

Ces fonctions renvoient 0 ou 1

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

**Syntaxes complètes :**

```
IS_ROLEMEMBER ( 'role' [ , 'user' ] )  
IS_SRVROLEMEMBER ( 'role' [ , 'login' ] )  
HAS_PERMS_BY_NAME ( 'objet', 'classe', 'privilège' [ , 'sous-objet' ] [ ,  
'classe-sous-objet' ] )
```

Classes :

```
APPLICATION ROLE | ASSEMBLY | ASYMMETRIC KEY | CERTIFICATE  
| CONTRACT | DATABASE | ENDPOINT | FULLTEXT CATALOG | LOGIN  
| MESSAGE TYPE | OBJECT | REMOTE SERVICE BINDING | ROLE |  
ROUTE  
| SCHEMA | SERVER | SERVER ROLE | SERVICE | SYMMETRIC KEY |  
TYPE  
| USER | XML SCHEMA COLLECTION
```

Sous-objet par exemple *column*

## 3 / Fonctions de sécurité

56

Fonction table :

`sys.fn_my_permissions(objet, classe)`

Renvoie la liste des  
privilèges accordés  
à la classe et  
l'éventuelle  
sous-classe

| entity_name            | subentity_name | permission_name |
|------------------------|----------------|-----------------|
| [dbo].[T_E_IMPORT_IPT] |                | SELECT          |
| [dbo].[T_E_IMPORT_IPT] |                | UPDATE          |
| [dbo].[T_E_IMPORT_IPT] |                | REFERENCES      |
| [dbo].[T_E_IMPORT_IPT] |                | INSERT          |
| [dbo].[T_E_IMPORT_IPT] |                | DELETE          |
| [dbo].[T_E_IMPORT_IPT] | IPT_ID         | SELECT          |
| [dbo].[T_E_IMPORT_IPT] | IPT_CODE       | SELECT          |
| [dbo].[T_E_IMPORT_IPT] | IPT_NUMBER     | SELECT          |
| [dbo].[T_E_IMPORT_IPT] | IPT_TYPE       | SELECT          |
| [dbo].[T_E_IMPORT_IPT] | IPT_ID         | UPDATE          |
| [dbo].[T_E_IMPORT_IPT] | IPT_CODE       | UPDATE          |

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

**Exemple :**

```
SELECT *  
FROM sys.fn_my_permissions('dbo.T_E_IMPORT_IPT', 'OBJECT')
```

## 3 / Aides sur la sécurité

57

| Procédure            | Description                             |
|----------------------|---|
| sp_helpntgroup       | Liste les groupes NT autorisés          |
| sp_helplogins        | Liste les bases accessibles au login    |
| sp_helpsrvrole       | Liste les rôles de serveur              |
| sp_helpsrvrolemember | Liste les membre des rôles de serveur   |
| sp_helpdbfixedrole   | Liste les rôles fixe de base de données |
| sp_helprole          | Liste les rôles de base de données      |
| sp_helprolemember    | Liste les membres des rôles de BD       |

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Ces fonctions fournissent une aide sous forme de listes tabulaires.  
sp\_helplogins nécessite en paramètre un nom de connexion

# 3 / Synoptique niveaux sécurité

58



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Voir aussi les posters : « Database Engine Permissions », par exemple pour 2014 =>  
[http://social.technet.microsoft.com/wiki/cfs-file.ashx/\\_\\_\\_key/communityserver-wikis-components-files/00-00-00-00-05/0245.Permissions\\_5F00\\_Poster\\_5F00\\_2014.pdf](http://social.technet.microsoft.com/wiki/cfs-file.ashx/___key/communityserver-wikis-components-files/00-00-00-00-05/0245.Permissions_5F00_Poster_5F00_2014.pdf)

Illustration Microsoft.

## 3 / Rôle d'application

59

Plus que les utilisateurs, ce sont des applications qui se connectent à l'instance SQL.

Le but des rôles d'application est que l'application s'identifie pour gérer sa propre sécurité.

Moyen simple pour de « petites » applications...

Opérer en deux temps :

- Création du rôle d'application
- Établissement du lien entre l'application et l'instance

## 3 / Rôle d'application

60

Un rôle d'application est propre à une base.

Syntaxe de création :

```
CREATE APPLICATION ROLE nom_role_application
    WITH PASSWORD = 'pass'
    [ , DEFAULT_SCHEMA = nom_schema ]
```

Attribuer les privilèges nécessaire (GRANT)

Dans l'application, lancer la procédure :

```
sys.sp_setapprole
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Syntaxe de la procédure sys.sp\_setapprole

sp\_setapprole

```
[@rolename= ] 'role',
[@password = ] { encrypt N'password' } | 'password'
[ , [@encrypt= ] { 'none' | 'odbc' } ]
[ , [@fCreateCookie= ] true | false ]
[ , [@cookie = ] @cookie OUTPUT ]
```

@rolename : nom du rôle d'application

@password : mot de passe associé

@encrypt : valeur possible none ou odbc. Si odbc, chiffre le mot de passe pour odbc ou OLE DB

@fCreateCookie : booléen indiquant si un cookie doit être généré

@cookie : paramètre en sortie, contenant les données du cookie (type SQL VARBINARY (800) mais en fait seuls les 50 premiers octets sont utilisés)

Une fois l'authentification réussie, la session se termine par la déconnexion physique ou par le lancement de la commande sys.sp\_unsetapprole avec le cookie d'origine.

## 3 / Sécurité au niveau ligne

61

Permet de filtrer implicitement les lignes des tables en fonction :

- De l'utilisateur SQL
- Du compte de connexion
- D'un identifiant de contexte
- D'un prédicat utilisant différents paramètres d'identification

Nécessite une colonne contenant le nom de l'entité de sécurité dans la table.

Procède par :

- Création d'une fonction de sécurité
- Application d'une police liant la fonction et un objet

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

« Row Level Security »

Concernent les :

- Tables
- Vues avec SCHEMABINDING

Exemple de fonction de sécurité :

```
CREATE FUNCTION S_SEC.F_FILTRE_SECURITE(@USER AS sysname)
RETURNS TABLE
WITH SCHEMABINDING
AS
RETURN SELECT 1 AS SECURITE
WHERE @USER = USER_NAME();
```

La police de sécurité se définit à l'aide de la syntaxe suivante :

```
CREATE SECURITY POLICY <nom_police>
ADD FILTER PREDICATE <nom_fonction>(<arguments_col_table>)
ON <nom_objet>
WITH (STATE = { ON | OFF });
```

## 3 / Propriétaire

62

Chaque objet créé est la propriété :

- D'un compte de connexion pour un objet de niveau serveur (base incluse)
- D'un utilisateur SQL dans la base

Les propriétaires ont tous les droits sur les objet qui leur appartient

Le transfert de propriété est possible via la commande : ALTER AUTHORIZATION

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La syntaxe détaillée de la commande ALTER AUTHORIZATION est la suivante :

```
ALTER AUTHORIZATION
  ON [ <classe_objet>:: ] nom_objet
  TO { SCHEMA OWNER | nom_propriétaire }
```

```
<classe_objet> ::=
{
  OBJECT
  ASSEMBLY
  ASYMMETRIC KEY
  CERTIFICATE
  CONTRACT | TYPE
  DATABASE
  ENDPOINT
  FULLTEXT CATALOG
  FULLTEXT STOPLIST
  MESSAGE TYPE
  REMOTE SERVICE BINDING
  ROLE
  ROUTE
  SCHEMA
  SEARCH PROPERTY LIST
  SERVER ROLE
  SERVICE
  SYMMETRIC KEY
  XML SCHEMA COLLECTION }
```



## 3 / Propriétaire

63

Dans la base :

- Tous les schémas ont un propriétaire.
- Un objet relationnel :
  - ▣ peut avoir un propriétaire particulier
  - ▣ à défaut, c'est le même que celui du schéma

Si un objet est manipulé par le biais du compte du propriétaire, tous les objets sous jacents le sont aussi et C'est le chaînage des propriétaires.

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Au niveau des objets serveur, le propriétaire est un compte de connexion.

Au niveau des objets de base, le propriétaire est un utilisateur SQL

**ATTENTION** : nous vous recommandons de ne pas créer des bases avec comme propriétaire un compte de connexion « système » (compte ou groupe Windows). En effet en cas de portage de la base vers un autre serveur qui n'aurait pas la visibilité sur ce compte (hors domaine ou compte local), des problèmes peuvent survenir...

Dans la vue sys.objects, la colonne principal\_id indique le propriétaire (utilisateur SQL). Si cette colonne est vide, le propriétaire est le même que celui du schéma dans lequel se trouve l'objet.

**NOTA** : les procédures stockées :

- sp\_changeobjectowner
- sp\_changedbowner

Anciennement destinées aux transfert de propriété sont obsolète. La première ne fonctionne qu'avec la version 2000.

## 3 / Propriétaire

64

### Chaînage des propriétaires entre les bases

D'une base à l'autre :

```
ALTER DATABASE nom_base  
SET DB_CHAINING ON;
```

À faire sur aux moins 2 bases

Sur tout le serveur :

```
EXECUTE sp_configure  
'cross db ownership chaining', 1;
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

**ATTENTION** à l'option de chaînage au niveau serveur !

La chaînage des propriétaire est particulièrement intéressant lors de l'utilisation de vues ou de procédures stockées.

Une exemple simple consiste à se demander comment gérer une base e données contenant des données communes de type

- code postaux
- code insee
- nomenclature de sécurité sociale

...

Lorsque vous naviguez de base en base (USE ...), vous pouvez à tout moment connaître la base originelle dans laquelle vous vous êtes connecté à l'aide de la fonction ORIGINAL\_DB\_NAME()

## 3 / Accréditations

65

Une accréditation permet à un compte SQL d'emprunter un compte système Windows pour effectuer une tâche en dehors du scope du moteur SQL. Syntaxe :

```
CREATE CREDENTIAL nom_accrédiation
    WITH IDENTITY = N'compte_systeme',
    SECRET = N'mot de passe'
```

Il faut ensuite l'assigner aux connexions voulues :

```
ALTER LOGIN nom_connexion
    ADD CREDENTIAL nom_accrédiation
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

En anglais « credential » et dans l'IHM SSMS « information d'identification ».

Par exemple pour envoyer un fax en lançant depuis SQL Server un exécutable.

## 3 / Masquage

66

Il est possible de masquer des données (Dynamic Data Masking - v. 2016)

- Attribut MASKED de la colonne.

Exemple :

```
PRS_EMAIL VARCHAR(256)
MASKED WITH (FUNCTION =
              partial(3, XXX@XXX, 4)
```

| PRS_EMAIL      |
|----------------|
| gigXXX@XXXe.fr |
| b.IXXX@XXXp.fr |
| optXXX@XXXt.fr |
| jeaXXX@XXXo.fr |
| MarXXX@XXX.com |
| eybXXX@XXX.net |

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

MASKED comporte 4 fonctions préprogrammées :

- DEFAULT : lettre transformées en X et nombres en 0, etc...
- PARTIAL : présente le début et la fin, et masque le milieu
- RANDOM : nombres remplacés par valeurs aléatoires
- EMAIL : présente le premier caractère et l'extension (.com, .fr...)

## 3 / Masquage du code

67

Il est possible de créer des objets dont le code est masqué :

- Option ENCRYPTION (vues, routines)

```
CREATE nom_objet ...  
WITH ENCRYPTION
```

**NOTA** : bien que son nom semble indiquer un chiffrement, l'option ENCRYPTION sert à masquer le code et non à le chiffrer. L'algorithme utilisé est sujet à une attaque massive et les données sont en clair dans certaines tables système.

Routines : procédures stockées, déclencheurs, fonctions.

## 3 / Masquage (chiffrement signé)

68

Permet de masquer et d'authentifier le code d'une routine et lui donner des privilèges.

Syntaxe :

```
ADD [ COUNTER ] SIGNATURE TO nom_objet  
BY <liste_chiffreurs>
```

- Un objet signé n'est authentifié que lors de son utilisation directe (elle est perdue si le module est appelé par ailleurs)
- Une contre signature (COUNTER) fait perdre l'authentification.

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Routines : procédures stockées, déclencheurs, fonctions.

Le chiffreur peut être un certificat ou une clef asymétrique.

La signature peut être multiple

La liste des objets signables ou signés est disponible dans la fonction système `sys.fn_check_object_signatures`

## 3 / Impersonalisation

69

Consiste à se faire passer pour un autre :

- Compte de connexion (et donc son éventuel utilisateur associé)
- Utilisateur SQL (et donc son éventuel compte de connexion associé)

**EXECUTE AS** : pour se faire passer pour autrui

**REVERT** : pour revenir en arrière

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

**ATTENTION** : REVERT ne revient en arrière que d'une seule impersonnalisation effectuée par EXECUTE AS. Il est en effet possible de se faire passer pour quelqu'un qui se fait passer pour quelqu'un qui se fait passer pour quelqu'un qui se fait passer pour quelqu'un qui se fait passer pour quelqu'un qui se fait passer pour... !

Nécessite à minima le privilège IMPERSONATE

**ATTENTION** : les privilèges obtenus par impersonnalisation sont stricts et non pas limité par les phases successives de dépersonnalisation. Ainsi toute personne dotée de ce privilège peut obtenir les privilèges sysadmin si elle se fait passer pour un compte de connexion de ce niveau.

Pour savoir qui vous êtes lorsque vous utilisez l'impersonnalisation, vous pouvez lancer la requête :  
SELECT SYSTEM\_USER AS NOM\_CONNEXION, USER AS UTILISATEUR\_SQL,  
ORIGINAL\_LOGIN() AS CONNEXION\_ORIGINELLE

## 3 / Impersonalisation

70

Deux niveaux :

- Dans un script SQL
- Pour dépersonnaliser l'exécution d'une routine

Noms génériques (dans les routines) :

- CALLER : contexte de l'appelant du module
- SELF : personne qui à crée dernier à avoir modifié le module
- OWNER : propriétaire du module

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

**Syntaxe pour un script :**

```
EXECUTE } AS { LOGIN | USER } = 'nom_entité_sécurité' [  
WITH { NO REVERT | COOKIE INTO @cookie } ]  
REVERT [ WITH COOKIE = @cookie ]
```

NO REVERT, signifie qu'il n'est pas possible de revenir au contexte précédent

La variable de cookie (varbinary(8000), facultative) sert à transmettre un jeton afin de sécuriser le retour en arrière

**Syntaxes pour une routine :**

Fonctions (sauf table en ligne assimilées à des vues),  
procédures stockées et déclencheurs DML :  
{ EXEC | EXECUTE } AS { CALLER | SELF | OWNER |  
'nom\_utilisateur SQL' }

Déclencheurs DDL de niveau bases de données :  
{ EXEC | EXECUTE } AS { CALLER | SELF | 'nom\_utilisateur  
SQL' }

Déclencheurs DDL de niveau serveur et déclencheur FOR LOGON  
:  
{ EXEC | EXECUTE } AS { CALLER | SELF | 'nom\_connexion' }



## 3 / Chiffrement

71

SQL server propose différentes méthodes et moyens pour assurer le chiffrement des données :

- Chiffrement de colonnes :
  - Permanent (Always Encrypted, v. 2016)
  - À déchiffrer
- Chiffrement du stockage (TDE)

Les clés de chiffrement peuvent être dans :

- SQL Server
- Un HSM

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

HSM : Hardware Security Module (voir par la suite)

Le chiffrement est utilisé en interne par SQL Server :  
Service Broker, Mirroring, AlwaysOn...

**ATTENTION** : le chiffrement des données induit systématiquement une baisse des performances dans les manipulations de données :

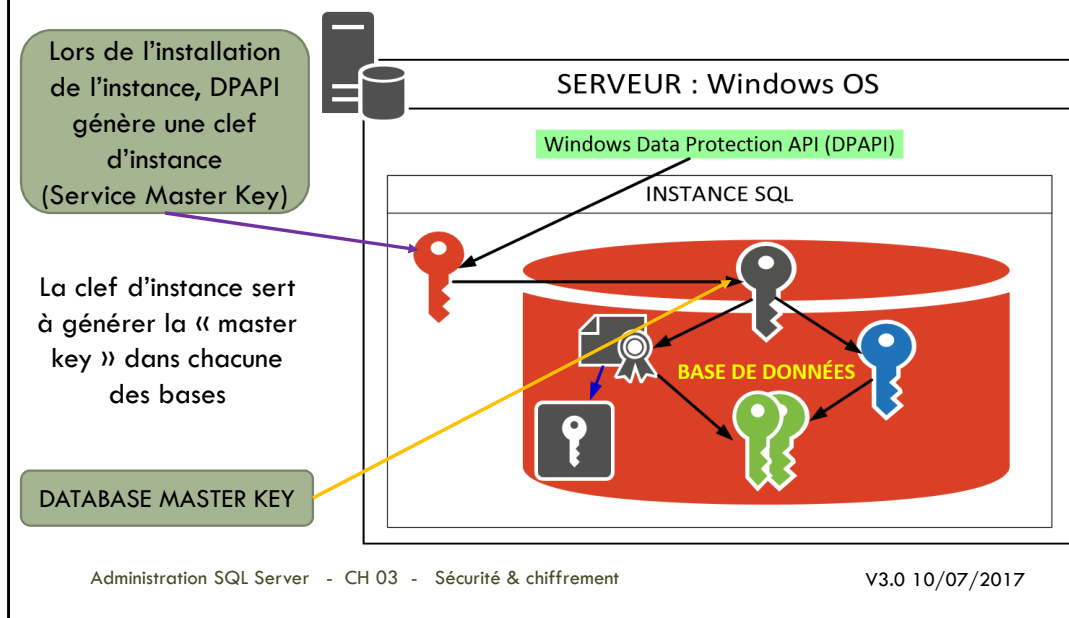
- Lecture par balayage et déchiffrement pour rechercher une donnée précise
- Volumétrie du chiffrement

Le choix de l'algorithme influe sur les performances. Certaines techniques permettent de rendre plus rapide l'accès aux données (ajout de colonnes hachées, technologie Always Encrypted...)

On peut opter pour un chiffrement partiel (par exemple pour les cartes de crédit, présenter les 4 derniers numéros).

## 3 / Chiffrement, architecture

72



### Dépendances des clefs dans l'instance SQL Server.

Chaque base comporte ses propres clefs.

Au niveau de l'instance, la clef principale de l'instance (Service master key) est automatiquement créée par l'API DPAPI de Windows. Elle sert à créer d'autres objets de chiffrement situé en dehors de la base (niveau instance)

La service master key est généré par l'algorithme Triple DES dans les versions 2014 et antérieures. À partir de la version 2016 c'est l'algorithme AES qui est utilisé.

Pour augmenter la sécurité de vos données, en cas de migration de bases vers 2016, il est souhaitable de régénérer vos clefs.

Il est possible de régénérer la Service Master Key via la commande :

```
ALTER SERVICE MASTER KEY
{ [ FORCE ] REGENERATE
  | { WITH OLD_ACCOUNT = 'account_name' , OLD_PASSWORD = 'password'
    | WITH NEW_ACCOUNT = 'account_name' , NEW_PASSWORD =
'password' } }
```

La clef d'instance étant générée au moment de l'installation, elle dépend du compte de service du service SQL Server. En cas de modification du compte de service utilisez le gestionnaire de configuration de SQL Server qui dispose d'une copie de la clef associée au compte. En cas de migration d'une instance vers une autre machine (VM par exemple) vous devrez migrer la clef à l'aide des commandes BACKUP et de RESTORE spécifique à ce type de clef. Cette clef est utilisée pour chiffrer les mots de passe des connexions SQL et les informations de connexion des serveurs liés.

## 3 / Chiffrement, architecture

73

Service Master KEY (SMK)... Algorithme :

- SQL Server 2016 : AES
- Précédemment : 3DES

En cas de migration de bases de données, régénérez toutes les clefs des bases migrées à l'aide de la commande :

```
ALTER MASTER KEY [ FORCE ] REGENERATE
```

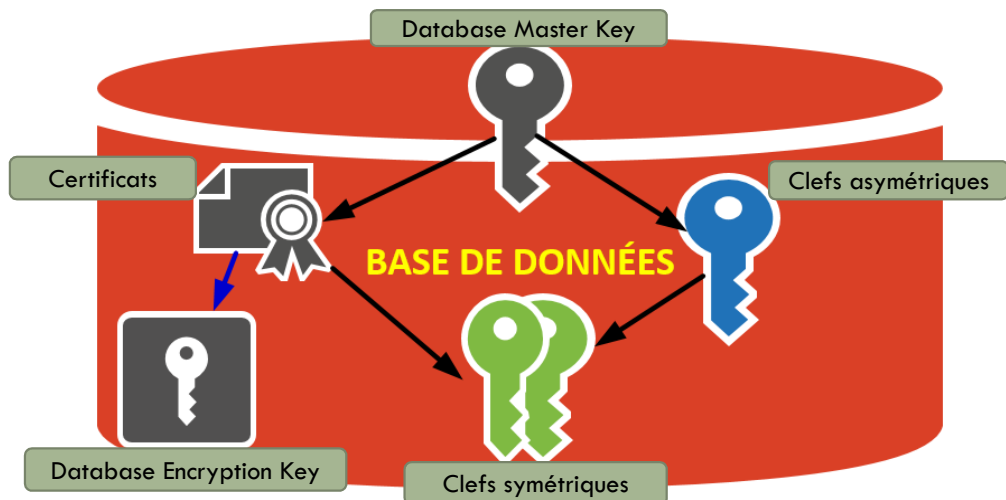
Abréviations :

- Service Master Key : SMK
- master key : DMK (pour Database Master Key)

Vous pouvez aussi en cas de déplacement d'instance, régénérer la « service master key » par la commande ALTER SERVICE MASTER KEY [ FORCE ] REGENERATE. Cependant, ceci peut briser la chaîne arborescente de dépendance des clefs, auquel cas les données chiffrées peuvent être irrémédiablement perdues. Dans un tel cas une migration des données en clair est un préalable fortement souhaitable. Dans tous les cas, pour une migration d'instance, il est important que le compte de service du service SQL Server soit le même.

## 3 / Chiffrement, architecture

74



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La clef principale de la base de données est générée d'après la clef d'instance (Service master key).

La database master key est générée par l'algorithme Triple DES dans les versions 2014 et antérieures. À partir de la version 2016 c'est l'algorithme AES qui est utilisé.

Pour augmenter la sécurité de vos données, en cas de migration de bases vers 2016, il est souhaitable de régénérer vos clefs.

Elle sert ensuite à générer :

- Des certificats
- Des clefs asymétriques

Clefs asymétriques et certificats servent à générer des clefs symétriques.

Une clef particulière appelée « Database Encryption Key » est générée par certificat pour le chiffrement du stockage (TDE). Voir plus loin.

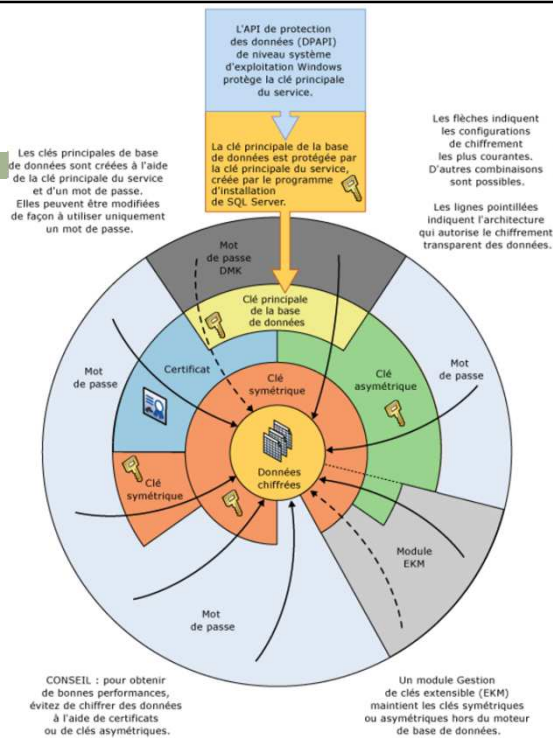
## 3 / Chiffrement

75

Cette hiérarchie de chiffrement permet de sécuriser l'ensemble des outils de chiffrement et de les tracer

Il faudra penser à sauvegarder toutes vos clefs !

Administration SQL Server - CH 03 - Sécurité & chiffrement



V3.0 10/07/2017

**ATTENTION** : il n'est pas possible de sauvegarder vos clefs symétriques et asymétriques de base. Mais vous pouvez les régénérer à l'identique sur un autre serveur à condition d'en connaître les paramètres et par exemple la phrase de passe...

Seules peuvent être sauvegardées :

- La clef maître de l'instance (Service Master Key - SMK)
- Les clefs maîtres des bases (Database Master Key - DMK)
- Les certificats

## 3 / Chiffrement

76

Clef maître de la base :

- Vous devez la créer avant mise en place de tout processus de chiffrement dans la base<sup>1</sup>.

```
USE nom_base;  
CREATE MASTER KEY  
    ENCRYPTION BY PASSWORD = 'mot de passe';
```

En cas de « vol », utilisez ALTER MASTER KEY pour régénérer les clefs

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

<sup>1</sup> sauf cas particulier du chiffrement par phrase de passe (déconseillé).

Autres syntaxes :

```
ALTER MASTER KEY  
{ [ FORCE ] REGENERATE WITH ENCRYPTION BY PASSWORD =  
'password'  
  | { [ ADD ENCRYPTION BY { SERVICE MASTER KEY | PASSWORD =  
'password' }  
    | DROP ENCRYPTION BY { SERVICE MASTER KEY | PASSWORD  
= 'password' } } }
```

Pour créer une clef maître de la base, vous devez avoir le privilège CONTROL sur la base.

Vous ne pourrez supprimer une clef maître de la base que si aucun autre processus de chiffrement n'est plus en vigueur dans la base. Voir <sup>1</sup>

Toutes les clefs maîtres des bases sont stockées en sus dans une table système de la base master visible dans la vue sys.databases

## 3 / Chiffrement



77

Clef asymétriques :

- Paire de clefs, l'une chiffre (publique, clef révélée), l'autre déchiffre (privée).
- Performances ↓ Sécurité ↑

Syntaxe :

```
CREATE ASYMMETRIC KEY nom_clef
  [ AUTHORIZATION propriétaire ]
  { [ FROM <source> ]
    | WITH <option>
    [ ENCRYPTION BY PASSWORD = 'password' ]
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

### Détail de la syntaxe

<source>::=

```
FILE = 'chemin_fichier'
| EXECUTABLE FILE = 'chemin_fichier_exe'
| ASSEMBLY nom_assembly
| PROVIDER nom_EKM
```

<option> ::=

```
ALGORITHM = <algorithme>
| PROVIDER_KEY_NAME = 'nom_clef_EKM'
| CREATION_DISPOSITION = { CREATE_NEW | OPEN_EXISTING }
```

<algorithme> ::= { RSA\_512 | RSA\_1024 | RSA\_2048 }

Une clef peut être extraite d'un fichier et plus particulièrement d'un exécutable ou d'une « assembly ». Elle peut être générée par un HSM (nom\_EKM, PROVIDER). Voir plus loin. CREATE DISPOSITION n'est utilisable que dans ce cas.

## 3 / Chiffrement



78

Clef symétriques :

□ La même clef sert à chiffrer et déchiffrer

□ Performances ↑ Sécurité ↓

```
□ CREATE SYMMETRIC KEY nom_clef
  [ AUTHORIZATION propriétaire ]
  [ FROM PROVIDER nom_EKM ]
  WITH <liste_options>
  | ENCRYPTION BY <chiffrement_clef>
```

**NOTA** : évitez les algorithmes de faible chiffrement (RC2, RC4, RC4\_128)

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Détail de la syntaxe :

<options> ::=

```
KEY_SOURCE = 'phrase_de_passe'
| ALGORITHM = <algorithme>
| IDENTITY_VALUE = 'phrase_identifiante'
| PROVIDER_KEY_NAME = 'nom_clef_EKM'
| CREATION_DISPOSITION = {CREATE_NEW | OPEN_EXISTING }
```

<algorithme> ::= DES | TRIPLE\_DES | TRIPLE\_DES\_3KEY | RC2 | RC4 | RC4\_128 | DESX | AES\_128 | AES\_192 | AES\_256

<chiffrement\_clef> ::=

```
CERTIFICATE nom_certificat
| PASSWORD = 'mot_de_passe'
| SYMMETRIC KEY nom_clef_symétrique
| ASYMMETRIC KEY nom_clef_asymétrique
```

**NOTA** : DESX est obsolète.

KEY\_SOURCE : phrase secrète à partir de laquelle la clé est dérivée.

IDENTITY\_VALUE : phrase identifiante à partir de laquelle un GUID est généré pour baliser les données qui sont chiffrées à l'aide d'une clé temporaire.

La clef peut être générée par un HSM (nom\_EKM, PROVIDER). Voir plus loin. CREATE DISPOSITION n'est utilisable que dans ce cas.  
chiffrement\_clef : Cette clause permet de protéger la clef



## 3 / Chiffrement



79

### Clef symétriques :

- Elle peut être protégée par un autre mécanisme de chiffrement :
  - ▣ Clef asymétrique
  - ▣ Clef TRIPLE\_DES générée par mot de passe
  - ▣ Certificat
  - ▣ Clef symétrique
- Car elle, est vulnérable !

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Exemple :

```
CREATE SYMMETRIC KEY KS_SANTE  
WITH ALGORITHM = AES_256  
ENCRYPTION BY CERTIFICATE CERT_SANTE;
```

## 3 / Chiffrement

80

### Certificats :

Associe une clef asymétrique à une identité

Conforme à la norme X.509

```
CREATE CERTIFICATE nom_certificat
  [ AUTHORIZATION propriétaire ]
  { FROM <clef_existante>
    | <nouvelle_clef> }
  [ ACTIVE FOR BEGIN_DIALOG = { ON | OFF } ]
```

**NOTA** en l'absence de clause de dates, le début est la date courante et la fin dans un an (format UTC)

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Algorithme RSA utilisé. Clef de 1024 bits (128 bits) par défaut. Par import d'une source externe, clef de 384 à 4096 bits.  
ACTIVE FOR BEGIN\_DIALOG n'est valable que pour les certificats associés à Service Broker.

### Détails de la syntaxe :

```
<clef_existante> ::=
  ASSEMBLY nom_assembly
  | { [ EXECUTABLE ] FILE = 'chemin_fichier'
    [ WITH PRIVATE KEY ( <options_clef_privée> ) ] }
  | { BINARY = certificat_ASN
    [ WITH PRIVATE KEY ( <options_clef_privée> ) ] }

<options_clef_privée> ::=
  { FILE = 'chemin_clef_privée'
    [ , DECRYPTION BY mot_de_passe = 'mot_de_passe' ]
    [ , ENCRYPTION BY mot_de_passe = 'mot_de_passe' ] }
  | { BINARY = clef_privée_binaire
    [ , DECRYPTION BY mot_de_passe = 'mot_de_passe' ]
    [ , ENCRYPTION BY mot_de_passe = 'mot_de_passe' ] }

<nouvelle_clef> ::=
  [ ENCRYPTION BY PASSWORD = 'mot_de_passe' ]
  WITH SUBJECT = 'sujet_du_certificate'
  [ , <date_options> [ ,...n ] ]

<date_options> ::=
  START_DATE = 'datetime' | EXPIRY_DATE = 'datetime'
```

**NOTA** : un certificat doit toujours avoir un sujet. Format de date AAAAMMJJ[HH:MM[:SS[.nnn]]]

À SAVOIR... ci vous possédez des certificat au format PFX, vous pouvez les transformer au format PVK avec PVKConverter.exe

## 3 / Chiffrement

81

La clef du certificat, peut être extraite :

- d'un fichier encodé DER (FILE)
- Une DLL signée (EXECUTABLE FILE)
- Extraite d'un autre fichier (EXECUTABLE FILE + WITH PRIVATE KEY, si BINARY format binaire PVK)
- Extrait d'un certificat binaire au format ASN (BINARY)

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

### Exemple :

```
CREATE CERTIFICATE CERT_ENCODING_TRANSIT
FROM FILE = 'C:\securite\certifs\postage.cert'
WITH PRIVATE KEY (FILE = 'C:\securite\K\mail.pvk',
DECRYPTION BY PASSWORD = 'Le p@ß 2016.');
```

Ceci créé un certificat extrait d'un autre certificat (postage.cert) et charge une paire de clef d'un autre fichier mail.pvk qui est protégée par le mot de passe « Le p@ß 2016. »

## 3 / Chiffrement

82

En cas de perte d'une clef, il n'y a pas d'issue secrète pour « retrouver » la clef ou déchiffrer les donnés. La sauvegarde est une nécessité !

- ❑ BACKUP SERVICE MASTER KEY TO FILE
- ❑ BACKUP MATER KEY TO FILE
- ❑ BACKUP CERTIFICATE

**Il n'y a pas de BACKUP pour les clefs !**

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

En fait, SQL Server étant sa propre autorité de certification, tout est conservé et protégé dans les bases de données, sous la responsabilité des clefs et certificats des hiérarchies supérieures.

Une clef qui a été créé sans spécification de source et d'identité ne peut en aucun cas être scriptée ou copier.

Une clef créée avec une source et une identité générera la même valeur de clef quelque soit le serveur. Pas besoin de sauvegarder il suffit de connaître les éléments d'identification.

Option impérative : ENCRYPTION BY PASSWORD = 'mot de passe'

Protège le fichier de sauvegarde. Il sera nécessaire de donner cette clef lors de la restauration.

La restauration entrainera le rechiffrement de toutes les données !

Restaurer une clef nécessite le privilège CONTROL SERVER

Exemple :

```
USE Mabase;
```

```
OPEN MASTER KEY
```

```
    DECRYPTION BY PASSWORD = 'Le p@ß 2016.'
```

```
BACKUP Clé Principale TO FILE = 'C:\securite\mk\DB_SANTE_MK.BKK'
```

```
    ENCRYPTION BY PASSWORD = 'aqwAZE951';
```

```
GO
```

```
USE MaBase
```

```
RESTORE MASTER KEY FROM FILE = 'C:\securite\mk\DB_SANTE_MK.BKK'
```

```
    DECRYPTION BY PASSWORD = 'aqwAZE951'
```

```
    ENCRYPTION BY PASSWORD = 'New p@ß 2016.'
```

```
GO
```

NOTEZ que la clef doit être ouverte pour pouvoir s'en servir...

## 3 / Chiffrement via EKM

83

EKM (Extensible Key Management) :

Permet l'utilisation d'un HSM (Hardware Security Module)

- boîtier électronique de génération et protection des clefs de chiffrement;
- autodétruit ses données en cas de manipulation physique



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Image : Thalès nShield

Le boîtier est situé sur le réseau et doit être atteint par un exécutable sous forme de DLL

## 3 / Chiffrement via EKM

84

Procéder en 3 temps :

- Autoriser l'utilisation de l'EKM

```
EXEC sp_configure 'EKM provider enabled', 1
```

- Créer l'objet fournisseur de chiffrement avec l'ordre SQL :

```
CREATE CRYPTOGRAPHIC PROVIDER nom_EKM  
FROM FILE = 'fichier_DLL';
```

- Créez vos objets de cryptographie avec l'option PROVIDER...

## 3 / Chiffrement

85

### Fonctions de chiffrement/déchiffrement

| Chiffrement         | Déchiffrement           | clef |
|---------------------|-------------------------|------|
| EncryptByPassPhrase | DecryptByPassPhrase     | SYM  |
| EncryptByKey        | DecryptByKey            | SYM  |
|                     | DecryptByKeyAutoCert    | SYM  |
|                     | DecryptByKeyAutoAsymKey | SYM  |
| EncryptByAsmKey     | DecryptByAsmKey         | ASYM |
| EncryptByCert       | DecryptByCert           | ASYM |

**ATTENTION : le chiffrement est salé !**

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

**REMARQUE** : en fait ces fonctions sont originellement écrites en majuscule...

Dans la cas particulier la « passPhrase » pas de référence de clef à passer, mais la clef asymétrique est générée à la volée par la phrase de passe.

Pour chiffrer avec une clef symétrique il faut d'abord l'ouvrir, sinon utiliser les fonctions Auto.

Les données chiffrées sont en binaire ([VAR]BINARY...). Lors du déchiffrement il faut les transtyper dans le type d'origine (CAST)

Les clefs étant identifiées par un GUID, c'est cette référence qu'il faut passer aux fonctions de chiffrement. Si vous ne la savez pas, vous pouvez utiliser la fonction KEY\_GUID.

Pour pouvoir utiliser une clef symétrique ou une « master key » il faut d'abord l'ouvrir et la refermer au plus tôt :

- OPEN SYMMETRIC KEY
- CLOSE SYMMETRIC KEY
- OPEN MASTER KEY
- CLOSE MASTER KEY

## 3 / Chiffrement

86

Always Encrypted (v. 2016), architecture :

- Attribut de colonne
- Chiffrement/déchiffrement depuis application cliente
- Utilise le middleware ADO.net
- Deux modes :
  - Déterministe (permet les recherches =, groupages, tris)
  - Aléatoire (salée comme dans le chiffrement classique)

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Nécessite :

le framework .net version 4,6 à minima

Un certificat stocké sur la machine cliente dans son espace personnel (Current User/Personal/...)

Une clef maître de colonne (CMK : Columns Master Key), créée dans SQL Server (CREATE COLUMN MASTER KEY...)

Une clef de chiffrement de colonne (CEK : Column Encryption Keys) créée dans SQL Server (CREATE COLUMN ENCRYPTION KEY...)

Des colonnes chiffrées « Always Encrypted » dans les tables.

Informé la connexion de l'application de l'existence de ce type de chiffrement (Column Encryption Setting=Enabled)

Possible avec JDBC 6.0 au lieu de ADO



## 3 / Chiffrement

87

Always Encrypted, syntaxe de la colonne dans la table :

```
<nom_colonne> <type_SQL>  
  ENCRYPTED WITH  
    ( ENCRYPTION_TYPE =  
      { DETERMINISTIC, RANDOMIZE }  
      ALGORITHM =  
        'AEAD_AES_256_CBC_HMAC_SHA_256',  
        COLUMN_ENCRYPTION_KEY = clef_CEK )
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les performances sont correctes : doublement du temps de réponse au pire.

La mécanique de recherche, tri, groupage... devient transparente; Elle est assumée par le biais du middleware et de SQL Server. Les données restent cryptées sur le réseau.

## 3 / Chiffrement, signature

88

Permet d'authentifier des données.

Un texte situé en clair dans une colonne et la signature dans une autre colonne.

Fonctions de signature :

| Signature     | Vérification         |
|---------------|----------------------|
| SignByAsymKey | VerifySignedByAsmKey |
| SignByCert    | VerifySignedByCert   |

## 3 / Chiffrement et hachage

89

### Fonctions complémentaires :

| Fonction        | Description   |
|-----------------|---|
| CERTPRIVATEKEY  | Extrait la clef d'un certificat au format binaire PVK                       |
| CERTENCODED     | Extrait la partie publique d'un certificat au format binaire                |
| PWDENCRYPT      | Retourne un hachage en utilisant l'algorithme courant                       |
| PWDCOMPARE      | Compare un hachage avec le texte à hacher en utilisant l'algorithme courant |
| CHECKSUM        | Calcule logiquement une somme de contrôle                                   |
| BINARY_CHECKSUM | Calcule binairement une somme de contrôle                                   |
| CHECKSUM_AGG    | Calcule la somme de contrôle d'une expression sur groupage                  |
| HASHBYTES       | Fonction de hachage généraliste retournant un binaire                       |

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Le résultat de SERTPRIVATEKEY peut être utilisé pour créer un nouveau certificat avec cette même clef par la commande CREATE CERTIFICATE avec l'option WITH PRIVATE KEY BINARY.

Le résultat de CERTENCODED peut être réutilisé pour créer un nouveau certificat par la commande CREATE CERTIFICATE avec l'option WITH BINARY.

PWDENCRYPT est annoncée comme « deprecated » et risque de ne plus fonctionner dans une version future

CHECKSUM et BINARY\_CHECKSUM peut être utilisé en colonne calculée persistante pour soulager certaines recherches pour des données très longues (indexation de hachage).

Ces deux fonctions prennent en argument soit avec l'opérateur \* soit une ou plusieurs expressions et renvoie un entier (INT)

CHECKSUM\_AGG prend en argument une expression et renvoie un entier (INT)

HASHBYTES : renvoie un VARBINARY de 8 000 octets au maximum.  
Algorithmes supportés :

- MD2, MD4, MD5
- SHA, SHA1, SHA2\_256, SHA2\_512

## 3 / Chiffrement TDE

90

Transparent Data Encryption :

- Chiffrement transparent pour l'utilisateur, d'une base de donnée entière
- Fichiers de données et de journal sont chiffrés sur disque, et en clair en mémoire
- Le travail s'effectue à la volée dans les opérations d'IO physiques
- Protège contre le vol des fichiers et de la sauvegarde
- Clef particulière : Database Encryption Key (DEK)

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

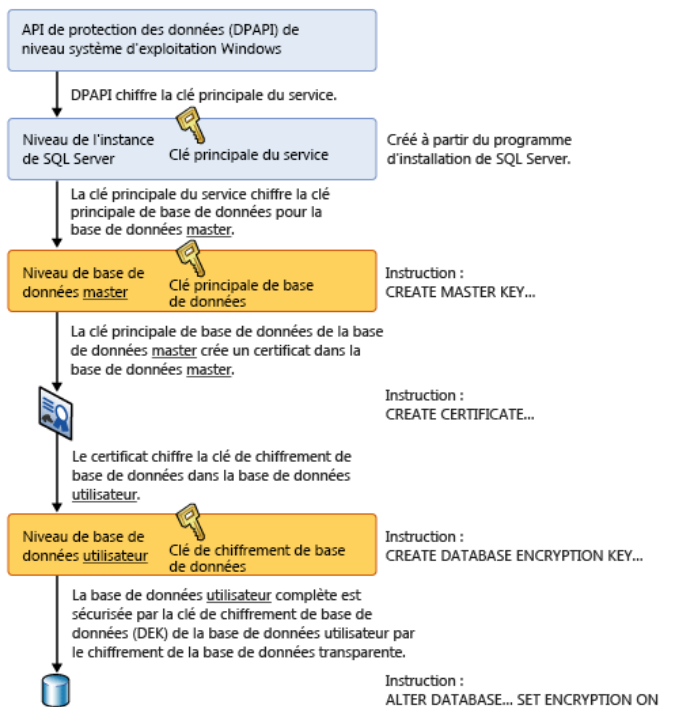
**ATTENTION** : seule les nouvelles transactions seront chiffrées dans la journal. Pour être sûr, effectuez une sauvegarde transactionnelle avant, suivi d'une troncation et d'un redimensionnement du journal.

La Database Encryption Key ne peut pas être sauvegardée. Mais, elle est incorporée à la base. Pensez à sauvegarder les clefs de niveau supérieur et le certificat.

# 3 / TDE

91

```
USE master;  
GO  
CREATE MASTER KEY ...  
go  
CREATE CERTIFICATE ...  
go  
USE MaBase  
GO  
CREATE DATABASE  
ENCRYPTION KEY  
...  
GO  
ALTER DATABASE Mabase  
SET ENCRYPTION ON;  
GO
```



Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La mise en place de TDE prends du temps. Elle est faite de manière transparente. Les utilisateurs peuvent continuer à travailler.

**NOTA** : la mise en place de TDE impose que le stockage de la base de données tempdb soit lui aussi chiffré.

D'après Microsoft, l'impact sur les performances est de 3 à 5 %.

## 3 / Chiffrement

92

### Métadonnées :

| Vue                             | Description                                |
|---------------------------------|--|
| sys.dm_database_encryption_keys | État du chiffrement dans une base (TDE)    |
| sys.asymmetric_keys             | Liste des clefs asymétriques               |
| sys.symmetric_keys              | Liste des clefs symétriques                |
| sys.certificates                | Liste des certificats                      |
| sys.crypt_properties            | Propriété des objets chiffrés              |
| sys.key_encryptions             | Liste de chiffrement des clefs symétriques |
| sys.openkeys                    | Clefs ouvertes dans la session             |

## 3 / Chiffrement

93

### Métadonnées pour les EKM :

| Vue                                      | Description                               |
|--|---|
| sys.cryptographic_providers              | Liste des EKM installés                   |
| sys.dm_cryptographic_provider_keys       | Liste les clefs d'un EKM                  |
| sys.dm_cryptographic_provider_algorithms | Liste les algorithmes disponible d'un EKM |
| sys.dm_cryptographic_provider_properties | Retourne les propriétés d'un EKM          |
| sys.dm_cryptographic_provider_sessions   | Retourne des sessions ouvertes d'un EKM   |

Les 4 vues dynamique de gestion (dm\_) prennent en argument un nom d'EKM

## 3 / Audit de sécurité

94

### DATABASE AUDIT :

permet de surveiller les accès de manière asynchrone :

- au serveur (SERVER AUDIT)
- à une base (DATABASE AUDIT)

...de manière fine et précise

Filtre sur :

- utilisateur
- objet
- commande SQL

Version Enterprise seulement.



## 3 / Audit de sécurité

95

Procéder en trois temps...

1 - créer l'espace d'audition (fichier ou journal d'événement)

CREATE SERVER AUDIT ...

2 - créer un objet de suivi relatif à l'espace d'audition

□ de niveau serveur :

CREATE SERVER AUDIT SPECIFICATION ...

□ de niveau base :

CREATE DATABASE AUDIT SPECIFICATION ...

3 - Activez l'audit :

ALTER SERVER AUDIT ... WITH ( STATE = ON )

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Les données d'audition sont découplées du serveur SQL et ne sont pas stockées dans le serveur pour diverses raisons :

- 1) Ne pas être annulées en cas de ROLLBACK
- 2) Pouvoir être accessibles avec d'autres comptes de sécurité que ceux du serveur SQL
- 3) Pouvoir y accéder même en cas de crash du serveur

...

## 3 / Audit de sécurité

96

### Syntaxe de création de l'espace de travail :

```
CREATE SERVER AUDIT nom_audit
TO { FILE ( <options_fichiers> [ , ...n ] )
      | APPLICATION_LOG
      | SECURITY_LOG }
[ WITH ( <options_audit> [ , ...n ] ) ]

<options_fichiers> ::=
{ FILEPATH = 'os_file_path'
  [ , MAXSIZE = { limite { <unit> } | UNLIMITED } ]
  [ , MAX_ROLLOVER_FILES = { nombre | UNLIMITED } ]
  [ , RESERVE_DISK_SPACE = { ON | OFF } ] }
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

FILE = emplacement (répertoire) des fichiers « textes » de l'audit.  
APPLICATION\_LOG : journal d'application de Windows. SECURITY\_LOG : journal de sécurité de Windows.

Pour auditer un fichier, spécifiez à minima son emplacement.

Préférez réserver l'espace du fichier pour le temps de l'audit.

Pour un volume important prévoyez de créer une succession de fichiers (option ROLLOVER)

**ATTENTION** : cette commande doit être passée dans la base master exclusivement.

**NOTE** : à la création, l'audit de serveur est dans un état désactivé.

## 3 / Audit de sécurité

97

Syntaxe de création de l'espace de travail, suite :

```
<unit> ::=  
{ MB | GB | TB }  
  
<options_audit> ::=  
{  
  [ QUEUE_DELAY = délai_ms ]  
  [ , ON_FAILURE = { CONTINUE | SHUTDOWN } ]  
  [ , AUDIT_GUID = uniqueidentif ]  
}
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

QUEUE\_DELAY indique le délai maximal admissible pour l'écriture asynchrone.

ON\_FAILURE indique que faire en cas de panne :

- soit continuer sans enregistrer l'audit,
- soit arrêter l'instance SQL

AUDIT\_GUID sert à taguer les sessions d'audit en cas de distribution de charge (*Mirroring* et *Always On* en particulier)

## 3 / Audit de sécurité

98

Syntaxe de création d'événements **serveur** à auditer :

```
CREATE SERVER AUDIT SPECIFICATION nom_spec
FOR SERVER AUDIT nom_audit
ADD    ( <groupe_action_server1> )
        [, ( <groupe_action_server2> )
          [ ... ] ]
WITH ( STATE = { ON | OFF } ) ]
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La clause FOR SERVER AUDIT n'est pas obligatoire s'il n'existe qu'un seul espace de travail pour les audits.

NOTEZ que pour le niveau serveur seuls des groupes d'actions sont disponible.

Parmi les groupes d'actions de niveau serveur les plus classiques :

```
FAILED_LOGIN_GROUP,
BACKUP_RESTORE_GROUP,
SERVER_OPERATION_GROUP,
DATABASE_OPERATION_GROUP,
SERVER_STATE_CHANGE_GROUP,
DATABASE_CHANGE_GROUP
```

## 3 / Audit de sécurité

99

Syntaxe de création d'événements **database** à auditer :

```
CREATE DATABASE AUDIT SPECIFICATION nom_spec
FOR SERVER AUDIT nom_audit
ADD ( { <action_spec1> | <action_group1> } )
      [, ( { <action_spec2> | <action_group2>
            } )
      [, ... ] ]
[ WITH ( STATE = { ON | OFF } ) ];
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

La clause FOR SERVER AUDIT n'est pas obligatoire s'il n'existe qu'un seul espace de travail pour les audits.

Parmi les **groupes d'actions** de niveau base les plus classiques :

DATABASE\_OPERATION\_GROUP,  
DATABASE\_CHANGE\_GROUP  
DATABASE\_OBJECT\_CHANGE\_GROUP  
DATABASE\_PERMISSION\_CHANGE\_GROUP  
DATABASE\_OBJECT\_ACCESS\_GROUP  
SCHEMA\_OBJECT\_ACCESS\_GROUP

...

Les **actions** de niveau base sont :

INSERT, UPDATE, DELETE, SELECT pour les tables, fonctions table et vues

REFERENCES pour l'utilisation d'objets dépendants

EXECUTE pour les procédures stockées

RECEIVE pour les QUEUES (Service Broker)

## 3 / Audit de sécurité

100

Syntaxe de création d'événements **database** à auditer, suite :

```
<audit_action_spec> ::=  
    action1 [ , action2 [ , ... ] ]  
    ON [ type_classe:: ] securable  
    BY principal1 [ , principal2 [ , ... ] ]
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Le type de classe n'est à spécifier que pour les objets « conteneur » (base, schéma...)

Le « securable » est l'objet à auditer (table, vue, routine SQL...)

Les « principals » sont les utilisateurs SQL et toutes les entités de sécurité (certificat, rôle d'application...)

## 3 / Audit de sécurité

101

Dépouillement des données...

Pour un audit stocké en fichiers utiliser la fonction table `fn_get_audit_file`, syntaxe :

```
SELECT *
FROM sys.fn_get_audit_file (
    motif_fichier,
    fichier_initial,
    offset_fichier )
```

Administration SQL Server - CH 03 - Sécurité & chiffrement

V3.0 10/07/2017

Elle renvoie une table.

Les motifs de fichiers peuvent être :

<chemin\_d'accès>\\* : tous les fichiers à l'emplacement spécifié.

<chemin\_d'accès>\LoginsAudit\_{GUID} : tous les fichiers ayant la paire nom/GUID spécifiée.

<chemin\_d'accès>\LoginsAudit\_{GUID}\_00\_29384.sqlaudit : un fichier d'audit spécifique.

## 3 / Audit de sécurité

102

### Métadonnées de l'audit de bases de données :

| Vue                                      |   |
|--|---|
| sys.server_audits                        | Liste les audits                            |
| sys.server_audit_specifications          | Liste les spécif. d'audit de niveau serveur |
| sys.server_audit_specifications_details  | Détails des audits de niveau serveur        |
| sys.database_audit_specifications        | Liste les spécif. d'audits de niveau base   |
| sys.database_audit_specification_details | Détails des audits de niveau base           |
| sys.server_file_audits                   | Fichiers de stockage des audits             |
| sys.dm_audit_actions                     | Liste des actions d'audit                   |
| sys.dm_server_audit_status               | Statut du système d'audit                   |
| sys.dm_audit_class_type_map              | Jointure type d'audit à action              |





# SQL SERVER 2012/2017 ADMINISTRATION

Frédéric Brouard (SQLpro) [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
Richard Gaillard - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)

Version 3.001  
Date 10/07/2017



**4** Tâches d'administration

Administration SQL Server - CH 04 - Agent SQL V3.0 10/07/2017

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

## 4 – Tâche d'administration

3



- ❑ Présentation de l'Agent SQL
- ❑ Database mail
- ❑ Gestion des opérateurs
- ❑ Les travaux et leurs étapes
- ❑ Sécurité des travaux
- ❑ Planification des travaux
- ❑ Les alertes
- ❑ Monitorer l'Agent SQL

## 4 – Tâche d'administration

3



- Déclencheurs DDL...
- L'utilitaire DBCC
- Vérification d'intégrité de la base
- Réparer une base endommagée
- Les index : structure, audit, mise en place
- Les statistiques
- Les plans de maintenance

## 4 / Agent SQL

5

Service permettant :

- L'automatisation et la planification de travaux
- La gestion des alertes en réponse à des événements

Outil important pour le DBA

Basé sur le service SQLAgent (dépend du service SQL Server)

- Le mettre en, lancement automatique

Données de l'Agent stockées dans la base msdb

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

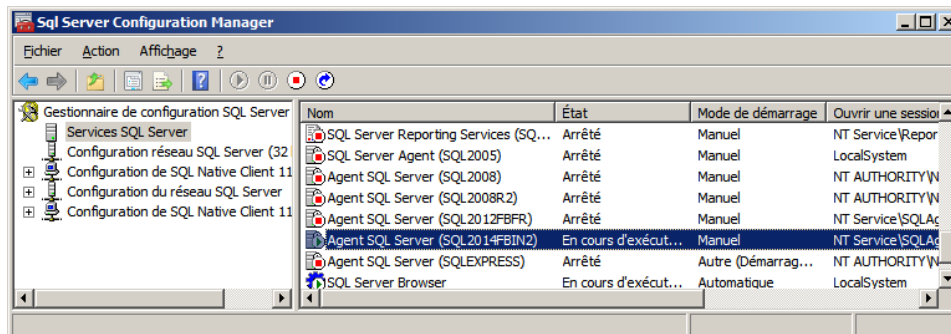
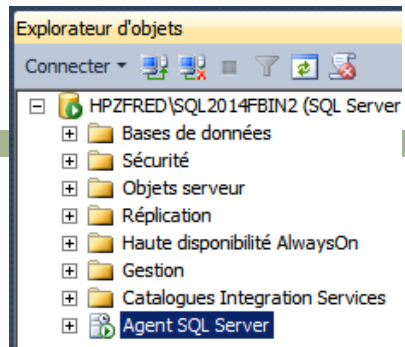
L'Agent SQL est utilisé de manière interne par différents mécanismes dont la gestion par politiques (Policy Management), la réplication de données, le « log shipping »...

Pensez à sauvegarder la base msdb (en général une fois par jour).

# 4 / Agent SQL

6

L'Agent SQL dans SSMS  
et dans le gestionnaire de  
configuration de SQL Server



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

## 4 / Agent SQL

7

Travaux (jobs) :

- Ensemble d'étapes enchainées par des règles;
- Doivent être planifiés.

Alertes (alerts) :

- Survenance d'un événement conduisant à :
  - Déclencher un travail;
  - Notifier un opérateur;
  - *Ou bien les deux.*

Les travaux peuvent être de différentes natures (SQL, systèmes, etc...)

Les événements déclenchant les alertes sont des événements relatifs au fonctionnement de SQL Server

## 4 / Database Mail

8

Permet d'envoyer des mails par le biais du moteur SQL

Utilise des profils (au moins un)

Un profil qui route les mails vers un (ou plusieurs) serveur SMTP :

- Interne

- Externe

Maintient un journal d'envoi :

- msdb.dbo.sysmail\_log

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

En français « Messagerie de base de données »

Très utile et important afin d'être informé des échecs des travaux. Peut aussi être utilisé applicativement par les développeurs (dans ce cas il est préférable de séparer les profils)

Un profil peut compter plusieurs serveurs SMTP pour plus de sûreté d'envoi (redondance). Ils sont essayés l'un après l'autre en cas d'échec d'envoi jusqu'à ce que le premier réussisse.



## 4 / Database Mail

9

Exécution en dehors du moteur SQL

Cluster aware

Redondance de serveurs SMTP

Envoi asynchrone via Service Broker

□ ATTENTION : l'envoi fait partie de la transaction

□ donc, annulé au rollback

Contrôles de sécurité (types d'attachement, taille, ...)

Service broker est une messagerie de données assurant des services de routage de données entre serveur SQL ou pour des opérations particulières (Database Mail, mirroring, AlwaysOn...)

## 4 / Database Mail

10

### Configuration...

- Activer Database Mail via sp\_configure (1)
- Configurer les paramètres de routage dans SSMS (Gestion / Messagerie de base de données)
- Configurer l'envoi de mail dans les propriétés de l'Agent SQL (onglet Système d'alerte)
- Créer au moins un opérateur avec contact par email
- Testez !

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

(1) Activé automatiquement pas l'assistant de configuration de la messagerie de base de données (message : « Le composant de messagerie de base de données n'est pas disponible. Voulez-vous activer ce composant ? »)

Sinon, lancez :

```
EXEC sp_configure 'Database Mail XPs', 1;
```

```
GO
```

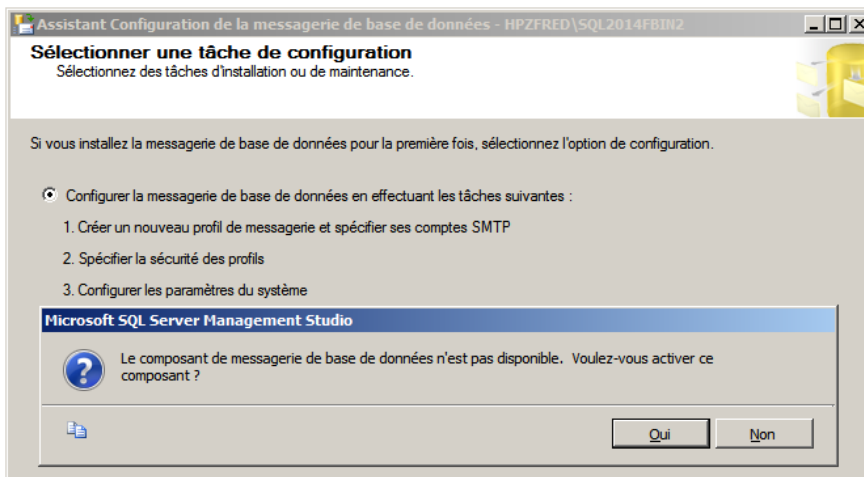
```
RECONFIGURE
```

```
GO
```

## 4 / Database Mail

11

### Assistant de configuration de Database Mail :



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Ceci effectue la modification de `sp_configure` pour autoriser l'utilisation des objets du « database mail » (Database Mail XPs).

## 4 / Database Mail

12

### Assistant de configuration de Database Mail :

Assistant Configuration de la messagerie de base de données - HPZFRID\SQL2014FRID02

**Nouveau profil**  
Spécifiez le nom, la description, les comptes et la priorité de basculement du profil.

Nom de profil : ProfilMailAgent  
Description : Ce profil est destiné à envoyer des mails pour les travaux de L'Agent SQL

**Nouveau compte de messagerie de base de données**  
Spécifiez le nom, la description et les attributs de votre compte SMTP.

Nom du compte : CompteAgentMail  
Description :  
Serveur de courrier sortant (SMTP)  
Adresse de messagerie : serveur\_sql@ma\_compagnie.com  
Nom complet :  
Répondre au courrier :  
Nom du serveur : smtp.orange.fr Numéro du port : 25  
 Ce serveur nécessite une connexion sécurisée (SSL).

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Donnez un nom au profil et une éventuelle description puis cliquez sur le bouton « Ajouter » pour définir l'accès à un serveur SMTP

Les paramètres impératif sont :

- Un nom de compte SMTP
- Une adresse mail (même bidon) qui sera celle de l'expéditeur des messages
- Le nom du serveur SMTP et son port
- L'authentification (voir ci après)

## 4 / Database Mail

13

Choisissez la méthode d'authentification au serveur SMTP :

- Windows si SMTP dans domaine (transparent)
- Anonyme, si possible
- Sinon renseignez les informations du compte d'accès

Authentification SMTP

Authentification Windows à l'aide d'informations d'identification du service Moteur de base de données

Authentification de base

Nom d'utilisateur : db\_admin@ma\_compagnie.com

Mot de passe : \*\*\*\*\*

Confirmer le mot de passe : \*\*\*\*\*

Authentification anonyme

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

L'authentification Windows NT ou anonyme est utilisé en interne lorsque le serveur SMTP est dans l'infrastructure informatique de l'entreprise.

## 4 / Database Mail

14

Un profil d'envoi de mail peut être :

- Public ou privé
- Par défaut ou non

Profil publics | Profils privés

Tous les utilisateurs de n'importe quelle base de données hôte de courrier peuvent accéder à un profil public.

Sélectionnez des profils publics. Vous pouvez également spécifier le profil public par défaut.

| Public                   | Nom de profil   | Profil par défaut |
|--------------------------|-----------------|-------------------|
| <input type="checkbox"/> | ProfilMailAgent | Non               |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

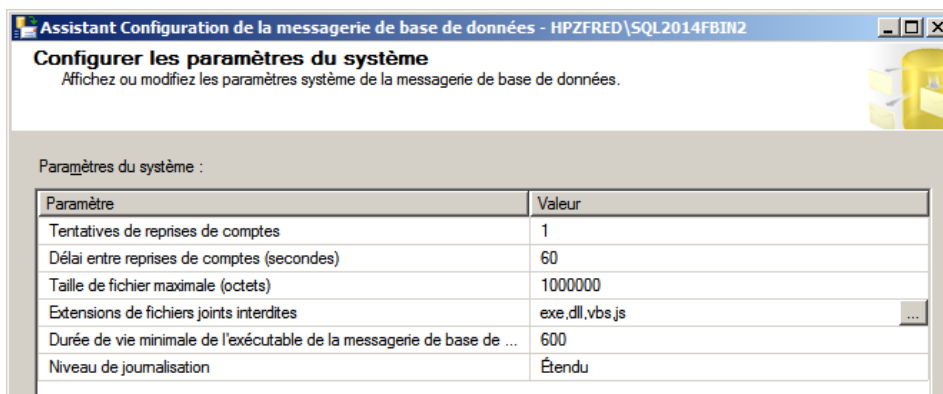
Un profil public peut être utilisé par n'importe quel utilisateur.

Un profil par défaut sera utilisé si l'on omet le compte de préciser le service d'envoi

## 4 / Database Mail

15

Il vous reste à régler les dernier détails :



Administration SQL Server - CH 04 - Agent SQL

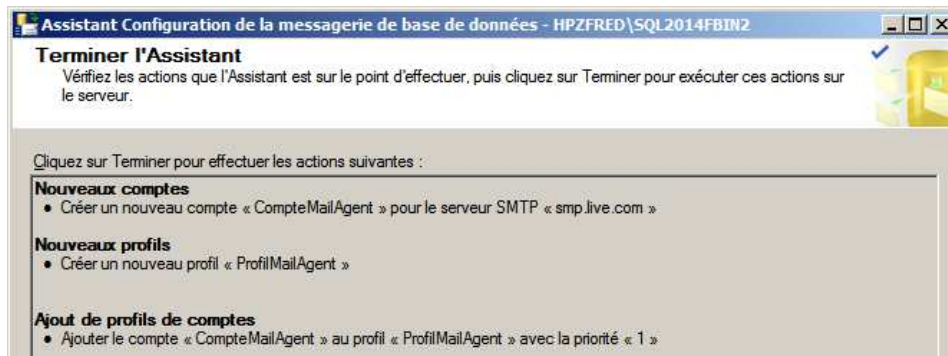
V3.0 10/07/2017

Montre les différents paramètres de sécurité des pièces jointes et du pilotage de l'envoi des mails

## 4 / Database Mail

16

Et finalement accepter de lancer cette création :



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Le script exécuter pour créer ce profil utilise les procédures suivantes de msdb :

- sysmail\_add\_account\_sp
- sysmail\_update\_account\_sp
- sysmail\_add\_profile\_sp
- sysmail\_add\_profileaccount\_sp

du schéma dbo

Si vous désirez scripter la chose, allez dans le menu et sélectionnez l'item « Affichage » puis « Explorateur de modèles. ».

Dans la fenêtre de l'explorateur de modèle, choisissez le dossier « Messagerie de base de données. », double-cliquez sur « Configuration simple de la messagerie de base de données ».

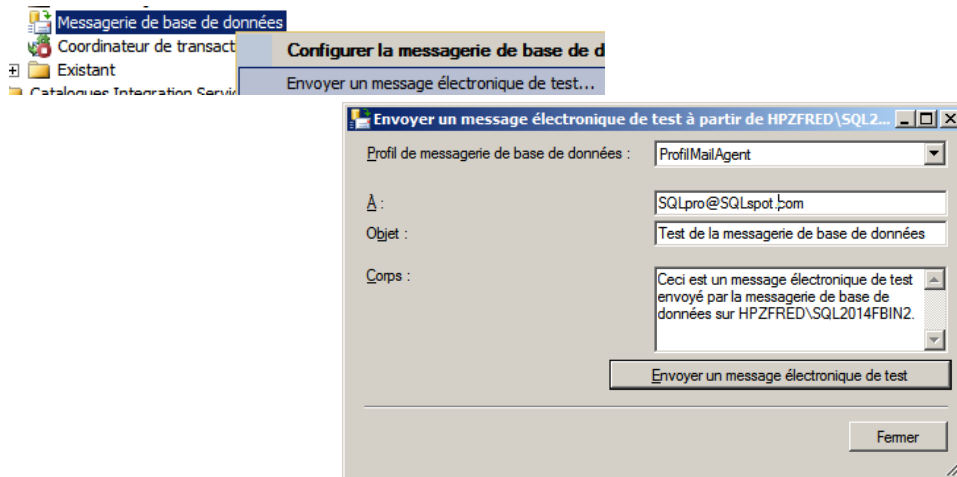
Un modèle de script Transact SQ s'ouvre dans une nouvelle fenêtre de requête.



## 4 / Database Mail

17

Il ne vous reste plus qu'à tester...



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Le départ du premier message peut prendre un peu de temps.

## 4 / Database Mail

18

En cas de panne...

- Database mail est-il lancé ? :
  - EXEC msdb.dbo.sysmail\_help\_status\_sp;
- Quel est l'état de la file d'attente des emails ?
  - EXEC msdb.dbo.sysmail\_help\_queue\_sp 'mail';
- Quels messages ont été envoyés :
  - SELECT \* FROM msdb.dbo.sysmail\_sentitems;
- Y a t-il des message d'erreurs ?
  - SELECT \* FROM msdb.dbo.sysmail\_event\_log;

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

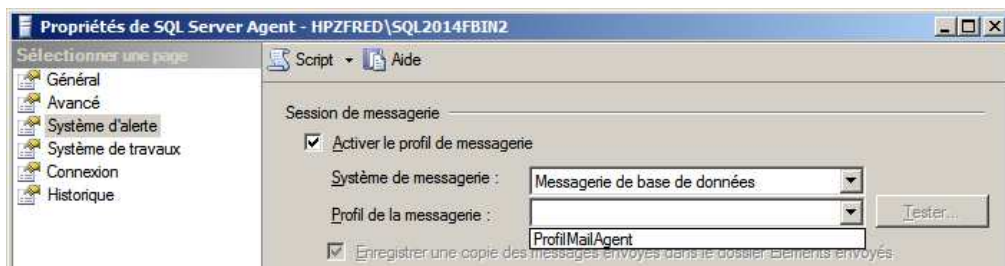
Pour arrêter ou démarrer Database Mail, lancez l'un des deux procédures :

- EXEC msdb.dbo.sysmail\_stop\_sp;
- EXEC msdb.dbo.sysmail\_start\_sp;

## 4 / Database Mail

19

Paramétrer l'Agent SQL pour envoyer des mails :  
Dans l'onglet « Système d'alerte » des propriétés de l'Agent SQL...



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

**ATTENTION** : pour les versions antérieures à 2012, veuillez relancer le service SQL Agent pour que ce paramètre soit pris en compte.

Il faut maintenant définir un ou plusieurs « opérateurs. »

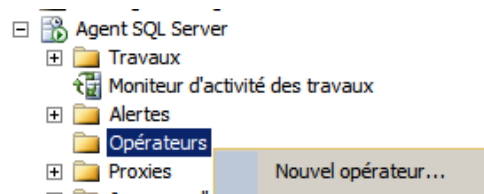
## 4 / Opérateur

20

Personne de votre organisation joignable par :

- Email
- Pager
- Net Send

Recensé en tant qu'Opérateur dans l'Agent SQL



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

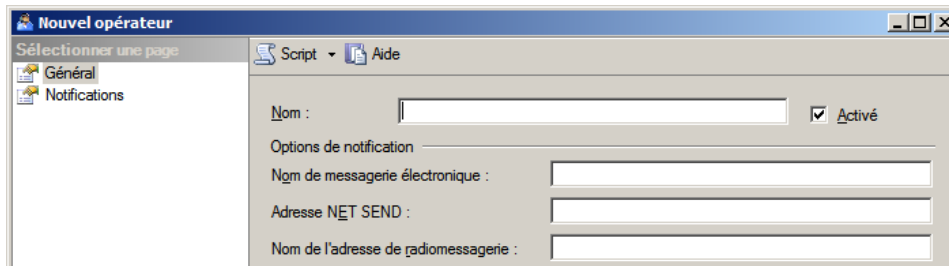
Pour le « pager », cela passe par des comptes mails

**ATTENTION** : les opérateurs par Radiomessagerie et net send sont considéré comme obsolètes.

# 4 / Opérateur

21

## Création



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Le nom doit être unique.

La procédure `msdb.dbo.sp_add_operator` est utilisée pour définir un opérateur et les données sont stockées dans `msdb.dbo.sysoperators`.

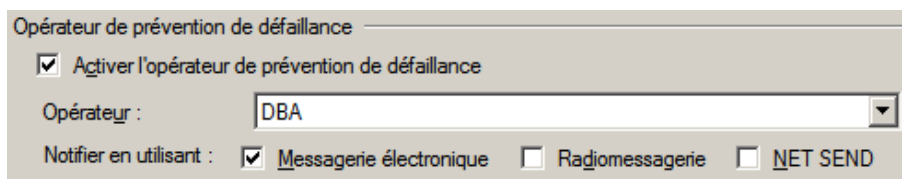
## 4 / Opérateur « fail safe »

22

Vous pouvez dans les propriétés de l'Agent SQL définir un opérateur de prévention de la défaillance.

Il lui sera envoyé un email en cas de défaillance :

- Base msdb inatteignable
- Un opérateur n'a pu être joint



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

En français, Microsoft a traduit « fail safe operator » par Opérateur de prévention de la défaillance.

Ce paramétrage se trouve dans l'onglet « Système d'alerte »

## 4 / Travaux de l'Agent

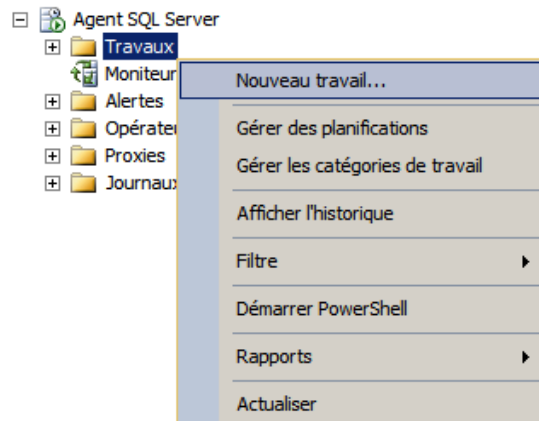
23

Lot d'instructions, commandes, script... composées d'une ou plusieurs étapes

À créer dans l'IHM

Procédure

`MsdB.dbo.sp_add_job`

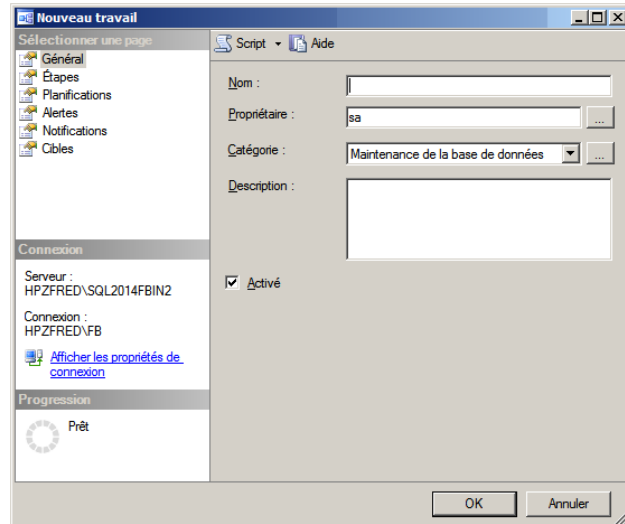


## 4 / Travaux de l'Agent

24

### Créer un travail :

- Nom
- Propriétaire
- Catégorie
- Description
- Activé



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Le nom doit être unique pour le serveur.

Le propriétaire est un compte de connexion (sécurité niveau serveur).

La catégorie est destinées à « ranger » les différents travaux.

La description est facultative.

Par défaut, lors de la création un travail est activé.

Lors de la création par la procédure `msdb.dbo.sp_add_job`, un travail est identifié par un GUID.

Les données des travaux sont stockées dans la table `msdb.dbo.sysjobs`

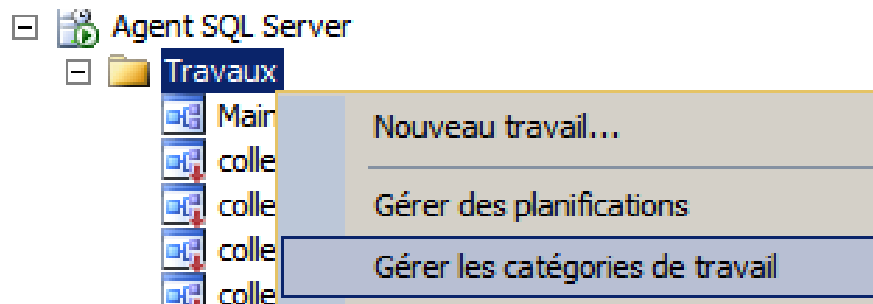


## 4 / Travaux de l'Agent

25

Catégorie de travaux :

- Des catégories préexistent
- Vous pouvez gérer les catégories



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Pour gérer les catégories de travail, cliquez droit au niveau « Travaux » de l'arborescence de l'Agent et sélectionnez l'item « Gérer les catégories de travail » dans le menu contextuel

## 4 / Travaux de l'Agent

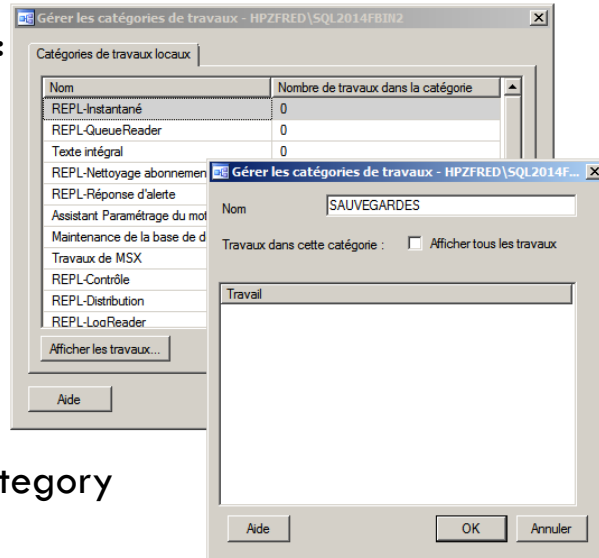
26

Catégorie de travail :

- Ajoutez
- Supprimez

Les catégories que  
Vous souhaitez

Procédure  
`msdb.dbo.sp_add_category`



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

En cas de suppression d'une catégorie assignée à des travaux, ces derniers passent en catégorie « N'appartenant à aucune catégorie »

## 4 / Travaux de l'Agent

27

Étapes d'un travail :

- Exécute un script (lot de commandes) d'un type spécifique dans un contexte de sécurité particulier

L'enchaînement des étapes est géré par des règles de précedence :

- En cas de succès
- En cas d'échec

Possibilité de différer l'enchaînement à l'étape suivante

Procédure msdb.dbo. sp\_add\_jobstep

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

*En cas de ...* choix possibles :

- Passer à l'étape suivante
- Quitter le travail en signalant la réussite
- Quitter le travail en signalant l'échec

On peut différer l'enchaînement de chaque étape par :

- Un multiplication des tentatives
- Un délai entre les différentes tentatives

L'IHM utilise la procédure msdb.dbo.sp\_add\_jobstep

## 4 / Travaux de l'Agent

28

### Étapes d'un travail :

Propriétés de l'étape du travail - Verify that automation is enabled.

Sélectionner une page

- Général
- Avancé

Script Aide

Nom de l'étape :  
Verify that automation is enabled.

Type :  
Script Transact-SQL (T-SQL)

Exécuter en tant que :

Base de données :  
master

Commande :  
IF (msdb.dbo.fn\_sypolicy\_is\_automation\_enabled() != 1)  
BEGIN  
RAISERROR(34022, 16, 1)  
END

Ouvrir...

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

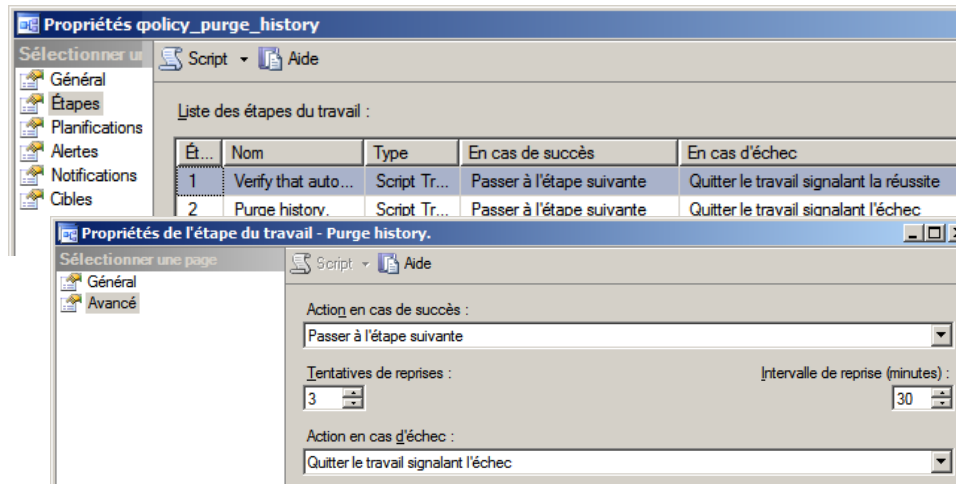
Les paramètres d'une étape sont :

- Le nom de l'étape
- Le type de commande (voir plus loin dans ce chapitre)
- Le contexte d'exécution (sécurité système)
- La base de données cible
- Le script ou lot de commande à exécuter

## 4 / Travaux de l'Agent

29

### Enchaînement des étapes :



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

L'enchaînement des étapes se règle dans l'onglet « Avancé » de la boîte de dialogue de paramétrage de chaque étape.

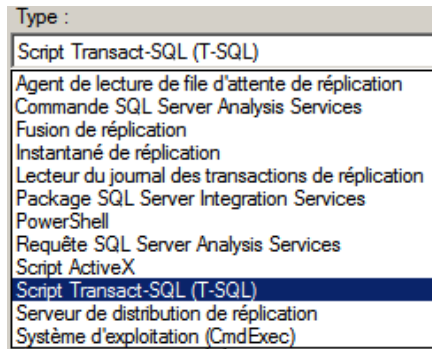
**ATTENTION** : l'enchaînement n'est pas forcément linéaire. Vous pouvez boucler, revenir à n'importe quelle étape et l'étape de départ n'est pas forcément la première...

## 4 / Travaux de l'Agent

30

Types des étapes :

- Lot d'instruction Transact SQL
- Commandes systèmes Windows
- Script PowerShell
- Script ActiveX
- Travaux spécifiques :
  - Réplication
  - OLAP (SSAS)
  - ETL (SSIS)



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

**ATTENTION** : le type d'étape Script ActiveX est obsolète, mais laissé pour rétro compatibilité

## 4 / Travaux de l'Agent

31

Informations de sortie d'étape :

Permet de définir comment les informations d'exécution seront enregistrées :

- dans un fichier
- dans une table
- dans l'historique de l'Agent

Fichier de sortie :  ... Afficher

Ajouter la sortie au fichier existant

Enregistrer un journal dans la table

Ajouter la sortie à l'entrée existante dans la table

Inclure la sortie de l'étape dans l'historique

Exécuter en tant qu'utilisateur  ...

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Le fichier de sortie est généralement un fichier texte.

**ATTENTION** : seuls les comptes de connexion membre du rôle sysadmin peuvent utiliser une sortie de type fichier.

L'option « Enregistrer un journal dans la table » consigne la sortie de l'étape de travail dans la table msdb.dbo.sysjobstepslogs

Le bouton « Afficher » montre la dernière sortie d'étape enregistrée dans la table

L'option « Ajouter la sortie à l'entrée existante dans la table » ajoute la sortie au contenu précédent dans la table. Sinon, il y a écrasement.

L'option « Inclure la sortie de l'étape dans l'historique » rajoute la sortie de l'étape à l'historique des travaux.

L'option « Exécuter en tant qu'utilisateur » permet de définir une autre connexion pour exécution de l'étape (réservé aux connexion membre du rôle sysadmin) travail.

## 4 / Sécurité des travaux

32

Par défaut les travaux de l'Agent SQL s'exécutent

- sous le compte du service de l'Agent pour les travaux externe (commande système, PowerShell)
- sous celui du Service SQL Serveur pour ce qui est des scripts Transact SQL

Certaines étapes peuvent parfois nécessiter une extension des droits pour aller chercher des ressources hors d'atteinte du compte de service.

Dans ce cas il faut utiliser un **"proxy"**

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les objets « proxy » sont dédiés à l'utilisation de l'Agent SQL lorsque les travaux doivent se dérouler dans le contexte de sécurité d'un compte de connexion SQL qui n'a pas accès à l'extérieur du serveur SQL



## 4 / Sécurité des travaux

33

Un proxy permet d'utiliser un compte système venant d'une accréditation (*credential*) afin d'effectuer une étape d'un type précis dans le contexte de sécurité relatif aux droits du compte système.

Par exemple pour atteindre un répertoire distant afin d'y placer des sauvegardes, vous pouvez utiliser un proxy. Pour cela vous devez procéder en trois temps :

- Créer l'accréditation (information d'identification)
- Créer le proxy
- Utiliser le proxy pour l'étape de travail considérée

Dans l'IHM SSMS, les accréditations sont accessible dans l'arborescence de l'explorateur d'objets, item « Sécurité / Informations d'identification »

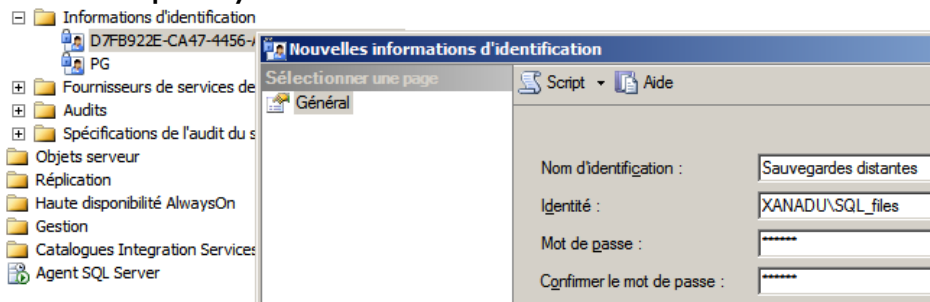
## 4 / Sécurité des travaux

34

Créer l'accréditation (information d'identification) :

```
CREATE CREDENTIAL nom_accréditation  
    WITH IDENTITY = N'compte système',  
    SECRET = N'mot de passe' ;
```

□ Le compte système doit avoir les droits sur les ressources



## 4 / Sécurité des travaux

35

Créer le proxy, exemple :

```
EXEC msdb.dbo.sp_add_proxy
    @proxy_name=N'Proxy file SQL',
    @credential_name=N'Sauvegardes distantes',
    @enabled=1;
EXEC msdb.dbo.sp_grant_proxy_to_subsystem
    @proxy_name=N'Proxy file SQL',
    @subsystem_id=3;
EXEC msdb.dbo.sp_grant_login_to_proxy
    @proxy_name=N'Proxy file SQL',
    @login_name=N'CLRLogin';
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Le « subsystem » 3 est celui dédié aux commandes systèmes (CmdExec).  
Pour avoir la liste des subsystems : `SELECT * FROM msdb.dbo.syssubsystems`

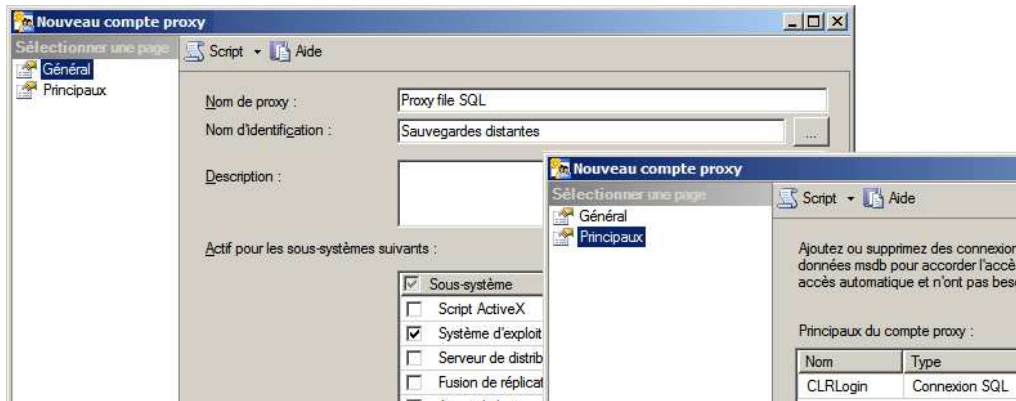
Autres métadonnées des proxies :

- `msdb.dbo.sysproxylogin` : compte de connexion associés aux proxies
- `msdb.dbo.sysproxysubsystem` : liste des liens subsystems / proxies
- `msdb.dbo.sysproxyloginsubsystem_view` : vue globale

## 4 / Sécurité des travaux

36

Créer le proxy, exemple (IHM) :



Administration SQL Server - CH 04 - Agent SQL

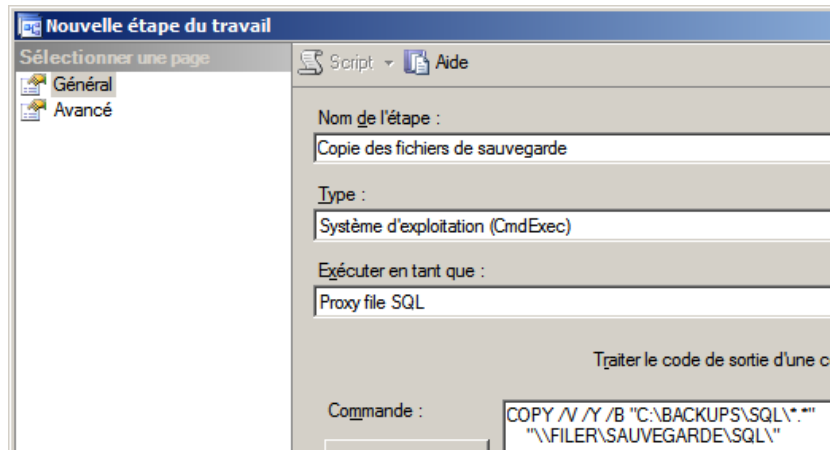
V3.0 10/07/2017

Les « proxy » sont regroupés dans l'arborescence de l'Agent SQL par catégorie

## 4 / Sécurité des travaux

37

Utilisez le proxy pour l'étape de travail considérée :



## 4 / Sécurité des travaux

38

3 rôles spécifiques à msdb pour les travaux :

### **SQLAgentUserRole**

- Peuvent créer des Travaux et voir seulement les Travaux leur appartenant

### **SQLAgentReaderRole**

- SQLAgentUserRole + affichage de toutes les Travaux et historique

### **SQLAgentOperatorRole**

- SQLAgentReaderRole + exécution des Travaux locales d'autres propriétaires

## 4 / Planification d'un travail

39

4 possibilités :

- périodique
- unique
- sur inactivité des processeurs
- au démarrage de l'Agent SQL

Les planifications sont partageables

Procédure : `msdb.dbo.sp_add_jobschedule`

Le moment où les processeurs sont considérés comme inactif (idle) est paramétrable dans les propriétés de l'Agent. Voir plus loin.

Il est aussi possible de démarrer des procédures suite au démarrage de l'instance par le biais d'un paramétrage particulier, mais ceci est généralement une mauvaise idée, car il n'est pas sûr que toutes les bases soient « montées » dès le démarrage. Pour ce faire vous devez :

- Créez une procédure dans la base master, schéma dbo
- Utilisez la procédure `sp_procoption` avec le paramètre `startup` à ON pour définir que votre procédure doit démarrer au lancement de l'instance

## 4 / Planification d'un travail

40

Nouvelle planification du travail

Nom : Planif TravDoc Travaux planifiés

Type de planification : Périodique  Activé

Une seule occurrence

Date : 28/02/2016 Heure : 09:54:28

Fréquence

Périodicité : Hebdomadaire

Répéter toutes les : 1 semaine(s) le

Lundi  Mercredi  Vendredi  Samedi  
 Mardi  Jeudi  Dimanche

Fréquence quotidienne

Une fois le : 12:00:00  
 Toutes les : 1 heure(s) Début à : 00:00:00  
Fin : 23:59:59

Durée

Date de début : 28/02/2016  Date de fin : 28/02/2016  
 Aucune date de fin :

Résumé

Description : A lieu toutes les semaines le Lundi, Mardi, Mercredi, Jeudi, Vendredi à 12:00:00. La planification

Administ

OK Annuler Aide

0/07/2017

Comme une planification peut être partagée, vous pouvez reprendre une planification existante en utilisant le bouton « Travaux planifiés »

Il est possible de définir une double planification. Exemple :

- Quotidienne
- Toutes les 30 minutes de 8h à 20h



## 4 / Notification d'un travail

41

Permet d'être informé du déroulement d'un travail.

Définir quel opérateur sera joint lors de :

- l'échec
- la réussite
- la fin

Du travail, et par quel média (email, pager, net send).

Onglet « Notifications » du travail

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Évitez de vous tenir informer de la réussite ou de la fin, mais uniquement des échecs :

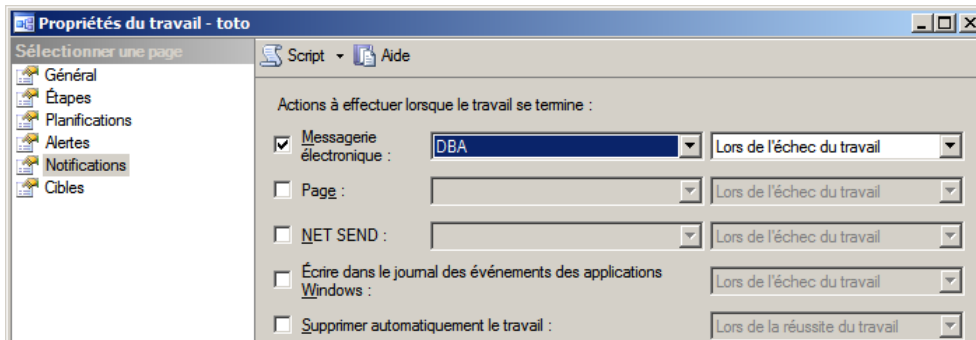
- Vous risquez de ne pas voir les quelques échecs parmi le flots des messages des réussites
- Vous pouvez être considéré comme spammer par les robots de routage si votre SMTP est externe

Les paramètres de notification d'un travail sont définis lors de la création du travail (sp\_add\_job / sp\_update\_job)

## 4 / Notification d'un travail

42

Vous pouvez aussi rajouter une information dans le journal d'événement pour les applications Windows



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Il est possible de définir un travail qui sera supprimé une fois exécuté (dans ce même onglet)

## 4 / Alertes

43

Événement, qui, lorsqu'il survient déclenche au choix :

- Un travail
- Un message envoyer à un opérateur
- Les deux

Peut être

- Une exception SQL
- Une condition de performance
- Un paramètre WMI

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les alertes sont des objets accessible directement dans l'arborescence de l'Agent SQL Server

La procédure `msdb.dbo.sp_add_alert` est utilisée pour créer une alerte et les données sont stockées dans `msdb.dbo.sysalerts`

WMI (Windows Management Instrumentation) est l'infrastructure de gestion des opérations effectuées par le système et permet de créer des scripts d'exécution pilotant l'OS ou des requêtes de contrôle afin d'obtenir des métriques. Les alertes WMI sont constituées par des requêtes WQL (un langage proche de SQL).

Présentation de WMI : <https://technet.microsoft.com/fr-fr/library/bb742445.aspx>

WQL : <https://msdn.microsoft.com/fr-fr/library/windows/desktop/aa392902%28v=vs.85%29.aspx>

## 4 / Alertes

44

### Sur exception SQL :

The screenshot shows the 'Définition d'une alerte d'événement' (Define Event Alert) dialog box in SQL Server Enterprise Manager. The 'Nom' (Name) field is empty, and the 'Activer' (Activate) checkbox is checked. The 'Type' (Type) dropdown is set to 'Alerte d'événement SQL Server'. Under 'Définition d'une alerte d'événement', the 'Nom de la base de données' (Database name) dropdown is set to '<toutes les bases de données>'. The 'Les alertes seront déclenchées selon' (Alerts will be triggered according to) section has three options: 'Numéro d'erreur' (Error number) with a radio button and a text box containing '1'; 'Gravité' (Severity) with a radio button and a dropdown menu showing '001 - Informations système diverses'; and 'Déclencher une alerte quand le message contient' (Trigger alert when message contains) with an unchecked checkbox and an empty text box. The 'Texte du message' (Message text) field is also empty.

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Donnez un nom (unique) à votre alerte

Le type d'alerte doit être : « Alertes d'événement SQL Server »

Choisissez une base de données cible, ou bien toutes les bases.

Les alertes seront déclenchées soit :

- Pour une erreur précise : donnez ce numéro (voir la table master.sys.sysmessages)
- Pour un niveau de gravité précis : choisissez dans la liste

Vous pouvez affiner en précisant un mot clef de filtration

**ATTENTION** : seules les erreurs journalisées soit

- naturellement (sévérité supérieure ou égale à 19)
- forcées (par un RAISERROR ... WITH LOG – considéré comme obsolète)
- lancées par la procédure de xp\_logevent

Seront captées par le système d'alerte.

Pour informations sur les niveaux d'erreurs (SEVERITY, STATE), lire :

[http://blog.developpez.com/sqlpro/p10296/ms-sql-server/parametrage\\_des constantes\\_severity\\_et\\_s](http://blog.developpez.com/sqlpro/p10296/ms-sql-server/parametrage_des constantes_severity_et_s)

## 4 / Alertes

45

Sur condition de performance, déclenchée par égalisation ou dépassement de seuil concernant une métrique SQL Server.

Métriques visibles dans la table :

- `sys.dm_os_performance_counters`  
La colonne `cntr_type`, définit l'unité de mesure.

La documentation est peu claire sur l'utilisation des métriques de compteurs. Certaines étant en valeur directe ou cumulative, d'autres en pourcentage. Voir les articles suivants pour plus d'information :

[https://blogs.msdn.microsoft.com/psssql/2013/09/23/interpreting-the-counter-values-from-sys-dm\\_os\\_performance\\_counters/](https://blogs.msdn.microsoft.com/psssql/2013/09/23/interpreting-the-counter-values-from-sys-dm_os_performance_counters/)  
[http://troubleshootingsql.com/2011/03/03/what-does-cntr\\_type-mean/](http://troubleshootingsql.com/2011/03/03/what-does-cntr_type-mean/)

## 4 / Alertes

46

Sur condition de performance, exemple :

The screenshot shows the 'Definition d'une alerte de condition de performances' dialog box. The fields are filled as follows:

- Nom : JT plein
- Type : Alerte de condition de performances SQL Server
- Objet : Databases
- Compteur : Percent Log Used
- Instance : DB\_MEDICACT
- Alerte si le compteur : s'élève au-dessus
- Valeur : 70

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

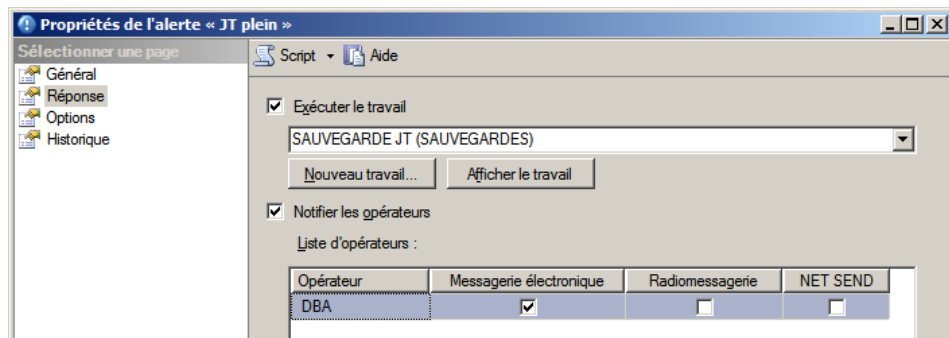
Ici on cherche à alerter si l'utilisation du journal dépasse 70% de la taille du fichier du journal des transactions.

## 4 / Alertes

47

En cas de déclenchement, exemple :

- Lancement d'un travail
- Information d'un opérateur



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Si notre alerte est levée, le travail est exécuté et un informateur contacté par email.

## 4 / Alertes

48

### Options supplémentaire :

Inclure le texte d'erreur de l'alerte dans :

Messagerie électronique

Radiomessagerie

NET SEND

Message de notification supplémentaire à envoyer :

Délai entre les réponses :

minutes  secondes

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Dans l'onglet « Options », vous pouvez :

- enrichir le message envoyé à l'opérateur
- ajouter un message personnalisé
- disposer d'un délai entre les relances de l'alerte

**NOTA** : nous vous conseillons fortement cette dernière option, car bien des seuils d'alerte se maintiennent longtemps et vous risquer de voir déferler un nombre important de message et d'être considéré comme spammeur !

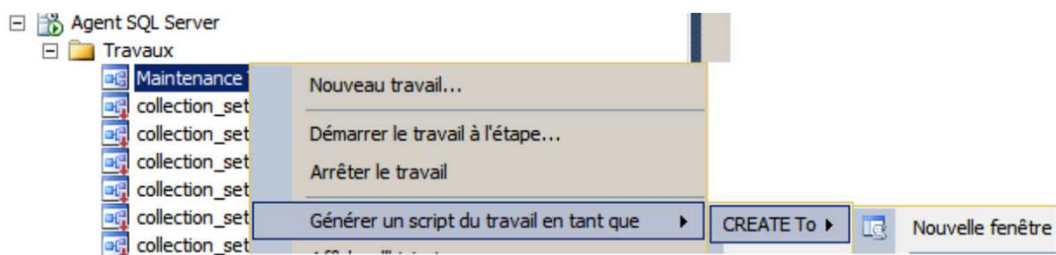


## 4 / Scripts

49

Tous les objets de l'Agent SQL, sont scriptables :

- Opérateurs
- Travaux
- Alertes



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Ceci peut être utile pour reprendre un travail générique et le faire exécuter sur plusieurs serveurs

## 4 / Monitorer l'Agent SQL

50

Journal d'exécution :

- Permet de voir l'historique d'exécution des travaux
- La santé de l'Agent SQL

L'historique peut être limité

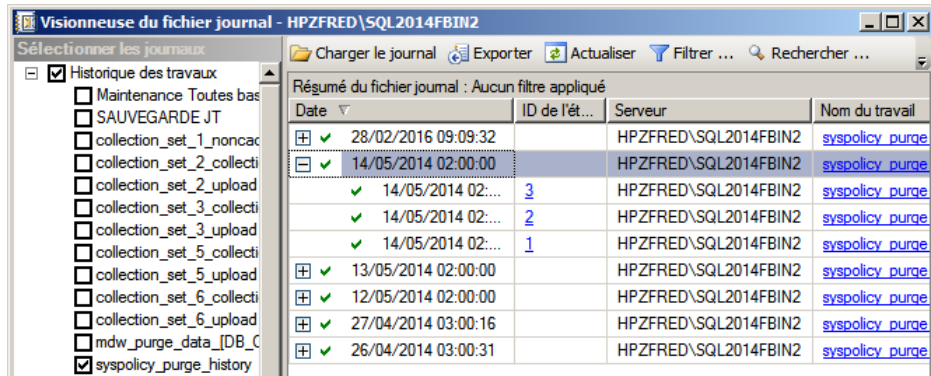
Voir onglet « Historique » dans les propriétés du serveur

- **CONSEIL** : rehausser les seuils !

## 4 / Monitorer l'Agent SQL

51

Journal d'exécution :



The screenshot shows a window titled 'Visionneuse du fichier journal - HPZFRED\SQL2014FBIN2'. It features a left sidebar with a tree view of job categories, including 'Historique des travaux' and various 'collection\_set' jobs. The main area displays a table with columns for 'Date', 'ID de l'ét...', 'Serveur', and 'Nom du travail'. The table lists several jobs, with the most recent one on 14/05/2014 at 02:00:00 being expanded to show three sub-entries with IDs 3, 2, and 1.

| Date                | ID de l'ét... | Serveur              | Nom du travail                  |
|---------------------|---------------|----------------------|---------------------------------|
| 28/02/2016 09:09:32 |               | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 14/05/2014 02:00:00 |               | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 14/05/2014 02:...   | 3             | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 14/05/2014 02:...   | 2             | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 14/05/2014 02:...   | 1             | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 13/05/2014 02:00:00 |               | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 12/05/2014 02:00:00 |               | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 27/04/2014 03:00:16 |               | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |
| 26/04/2014 03:00:31 |               | HPZFRED\SQL2014FBIN2 | <a href="#">syspolicy_purge</a> |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

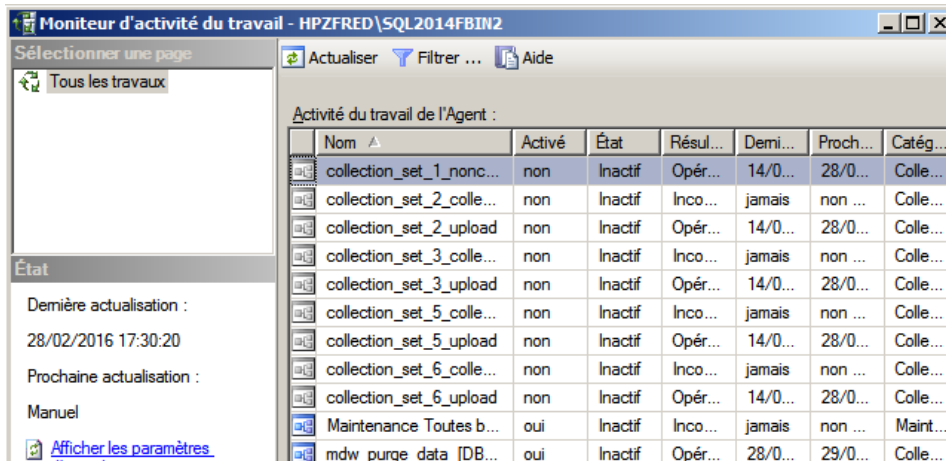
Disponible dans le menu contextuel de chaque travail (clic droit, item « Afficher l'historique »)

On peut filtrer, exporter, importer...

## 4 / Monitorer l'Agent SQL

52

### Moniteur d'exécution :



The screenshot shows a window titled 'Moniteur d'activité du travail - HPZFRED\SQL2014F8IN2'. It features a sidebar on the left with 'Sélectionner une page' (containing 'Tous les travaux') and 'État' (containing 'Dernière actualisation : 28/02/2016 17:30:20', 'Prochaine actualisation : Manuel', and a link 'Afficher les paramètres'). The main area displays 'Activité du travail de l'Agent :' with a table of jobs.

| Nom                       | Activé | État    | Résul... | Demi... | Proch... | Catég... |
|---------------------------|--------|---------|----------|---------|----------|----------|
| collection_set_1_nonc...  | non    | Inactif | Opér...  | 14/0... | 28/0...  | Colle... |
| collection_set_2_colle... | non    | Inactif | Inco...  | jamais  | non ...  | Colle... |
| collection_set_2_upload   | non    | Inactif | Opér...  | 14/0... | 28/0...  | Colle... |
| collection_set_3_colle... | non    | Inactif | Inco...  | jamais  | non ...  | Colle... |
| collection_set_3_upload   | non    | Inactif | Opér...  | 14/0... | 28/0...  | Colle... |
| collection_set_5_colle... | non    | Inactif | Inco...  | jamais  | non ...  | Colle... |
| collection_set_5_upload   | non    | Inactif | Opér...  | 14/0... | 28/0...  | Colle... |
| collection_set_6_colle... | non    | Inactif | Inco...  | jamais  | non ...  | Colle... |
| collection_set_6_upload   | non    | Inactif | Opér...  | 14/0... | 28/0...  | Colle... |
| Maintenance Toutes b...   | oui    | Inactif | Inco...  | jamais  | non ...  | Maint... |
| mdw_purge_data_IDB...     | oui    | Inactif | Opér...  | 28/0... | 29/0...  | Colle... |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Permet de savoir les travaux actif, inactif, le dernier résultat, la prochaine exécution...

Objet « Moniteur d'activité des travaux » dans l'arborescence de l'Agent SQL.

## 4 / Procédure « système »

53

Une procédure peut être rendue générique afin de pouvoir s'exécuter dans toutes les bases dans le contexte locale.

Pour cela :

- La procédure doit être créée dans master
- Son schéma doit être dbo
- Son nom commencer par sp\_
  - La « taguer » à l'aide de la procédure :  
sp\_MS\_marksystemobject

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Exemple :

```
USE master;
GO
CREATE PROCEDURE dbo.sp__ADMIN_CONSTRAINTS_VERIFY
AS
    DBCC CHECKCONSTRAINTS WITH ALL_CONSTRAINTS,
    NO_INFOMSGS;
GO
EXEC sp_MS_marksystemobject
'sp__ADMIN_CONSTRAINTS_VERIFY';
```

## 4 / Déclencheurs DDL...

54

Déclencheur spécial « FOR LOGON » :

- Se déclenche à la connexion

Déclencheur DDL :

- Sur événement DDL ou groupe d'événement DDL
- Au niveau serveur (ALL SERVER) ou au niveau base (DATABASE)
- Transmet un « paquet » informatif sous forme XML :
  - EVENTDATA()

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

**Syntaxe simplifiée du déclencheur sur connexion :**

```
CREATE TRIGGER nom_déclencheur
ON ALL SERVER
[ WITH ENCRYPTION ]
FOR LOGON
AS ...
```

**Syntaxe simplifiée des déclencheurs DDL :**

```
CREATE TRIGGER nom_déclencheur
ON { ALL SERVER | DATABASE }
FOR { événement_type | groupe_événement } [ ,...n ]
AS ...
```

## 4 / Déclencheurs DDL...

55

Exemple de déclencheur « FOR LOGON » :

```
CREATE TRIGGER E_LOGON_LIMIT_SA
ON ALL SERVER
FOR LOGON
AS
BEGIN
IF ORIGINAL_LOGIN()= 'sa'
    AND EXISTS(SELECT 1
                FROM sys.dm_exec_sessions
                WHERE is_user_process = 1
                AND original_login_name = 'sa'
                HAVING COUNT(*) > 1)
    ROLLBACK;
END;
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

La fonction ORIGINAL\_LOGIN renvoie le compte de connexion original  
La vue `sys.dm_exec_sessions` renvoie les sessions actuellement connectées  
ROLLBACK annule la transaction de connexion ce qui a pour effet d'interdire l'accès à l'instance

## 4 / Déclencheurs DDL...

56

Fonction EVENTDATA(), exemple :

```
<EVENT_INSTANCE>
  <EventType>CREATE_TABLE</EventType>
  <PostTime>2016-02-28T17:21:22.493</PostTime>
  <SPID>70</SPID>
  <ServerName>HPZFRED\SQL2014FBIN2</ServerName>
  <LoginName>HPZFRED\FB</LoginName>
  <UserName>dbo</UserName>
  <DatabaseName>DB_TEST_COL</DatabaseName>
  <SchemaName>dbo</SchemaName>
  <ObjectName>toto</ObjectName>
  <ObjectType>TABLE</ObjectType>
  <TSQLCommand>
    <SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON" ANSI_PADDING="ON" ... />
    <CommandText>CREATE TABLE toto (c int)</CommandText>
  </TSQLCommand>
</EVENT_INSTANCE>
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Extraire les parties intéressantes de l'événement qui à déclenché le trigger à l'aide des fonctions xml de SQL Server et de Xquery / Xpath...

La composition et la nature des informations du XML dépend des actions trappées.



## 4 / Déclencheurs DDL...

57

### Déclencheur DDL, exemple :

```
CREATE TRIGGER E_DDL_TABLE_CREATE
ON DATABASE
FOR CREATE_TABLE
AS
BEGIN
DECLARE @XML XML; SET @XML = EVENTDATA();
IF @XML.value('/EVENT_INSTANCE/SchemaName)[1]', 'sysname')
= 'dbo'
OR @XML.value('/EVENT_INSTANCE/ObjectName)[1]', 'sysname')
NOT LIKE 'T_?' ESCAPE '?'
    ROLLBACK;
END;
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Ce déclencheur interdit la création des tables dans le schéma dbo et qui ne commencent pas par T\_.

On peut intercepter des événements DDL spécifique comme des groupes d'événements. À lire :

DDL Event Groups : <https://msdn.microsoft.com/en-us/library/bb510452.aspx>

DDL Events : <https://msdn.microsoft.com/en-us/library/bb522542.aspx>

## 4 / L'utilitaire DBCC

58

### *DataBase Console Commands*

utilitaire spécifique au moteur de stockage pour :

- Obtenir des informations
- Modifier le comportement du moteur
- Effectuer la maintenance de bas niveau

À tendance à être remplacé par les DMV pour la partie information et parfois par des commandes Transact SQL.

Présente les informations de manière « libre », sauf si **TABLERESULTS** indiqué en paramètre

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

### **Quelques exemples de commandes DBCC remplacées par des DMV :**

DBCC SHOWCONTIG (...) : sys.dm\_db\_index\_physical\_stats(...)

DBCC IND (...) : sys.dm\_db\_database\_page\_allocations

DBCC USEROPTIONS : sys.dm\_exec\_sessions

### **Quelques exemples de commandes DBCC remplacées par des commandes Transact SQL :**

DBCC DBREINDEX (...) : ALTER INDEX ... REBUILD

DBCC INDEXDEFRAG (...) : ALTER INDEX ... REORGANIZE

DBCC DETACHDB (...) : EXEC sp\_detach\_db ...

Nous avons déjà vu les commandes suivantes :

DBCC SHRINKFILE : réduction d'un fichier

DBCC SHRINDATABASE : réduction des fichiers de la base

DBCC SQLPERF (LOGSPACE) : taille et remplissage des journaux de transactions

Pour rappel : DMV (Data Management View). Données dynamiques d'exécution de SQL Server, collectées et présentées sous forme de vues.

## 4 / L'utilitaire DBCC

59

Lorsque le résultat est tabulaire il ne peut pas faire l'objet de clauses SQL (WHERE par exemple).

On peut cependant réinjecter le résultat tabulaire dans une tables :

- De base
- Temporaire
- Ou une variable table

À l'aide d'un code SQL dynamique

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Exemple :

```
DECLARE @TRACESTATUS TABLE
(TRACE_FLAG INT,
TRACE_STATUS BIT,
TRACE_GLBAL BIT,
TRACE_SESSION BIT);
```

```
INSERT INTO @TRACESTATUS
EXEC ('DBCC TRACESTATUS'); --> DBCC TRACESTATUS est appelé dynamiquement
```

```
SELECT * FROM @TRACESTATUS WHERE TRACE_FLAG > 1000;
```

## 4 / L'utilitaire DBCC

60

### Principales commande d'information :

| DBCC              | Description  |
|-------------------|--|
| HELP              | Donne de l'aide sur DBCC                                   |
| OPENTRAN          | Montre la plus ancienne transaction dans la base courante  |
| PAGE              | Renvoie le contenu d'une page d'un fichier de données      |
| SHOWCONTIG        | Montre la fragmentation d'un index                         |
| SHOW_STATISTICS   | Montre la distribution d'une statistique                   |
| TRACESTATUS       | Montre les traces actives                                  |
| USEROPTIONS       | Montre les options active de l'utilisateur courant         |
| SQLPERF(LOGSPACE) | Montres la taille et le remplissage des journaux des bases |
| LOGINFO           | Informations sur les VLF du journal de la base courante    |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

### Syntaxes :

DBCC HELP ( 'dbcc\_statement' | @dbcc\_statement\_var | '?' ) [ WITH NO\_INFOMSGS ]

DBCC OPENTRAN

DBCC PAGE ( { 'dbname' | dbid }, filenum, pagenum [, printopt={0|1|2|3} ] [, cache={0|1} ] )

DBCC SHOW\_STATISTICS ( 'table\_name', 'target\_name' )

DBCC SHOWCONTIG ( table\_id | table\_name [, index\_id | index\_name ] [ WITH FAST, ALL\_INDEXES, TABLERESULTS [, ALL\_LEVELS ] ] )

DBCC TRACESTATUS ( trace# [, ...trace# ] )

## 4 / L'utilitaire DBCC

61

### Vérification d'intégrité des bases :

| DBCC             | Description   |
|------------------|---|
| CHECKALLOC       | Vérifie l'intégrité physique des pages techniques         |
| CHECKCATALOG     | Vérifie l'intégrité physique des tables systèmes          |
| CHECKTABLE       | Vérifie l'intégrité physique des tables utilisateur       |
| CHECKFILEGROUP   | Effectue une vérification globale d'un groupe de fichiers |
| CHECKDB          | Effectue une vérification globale de la base              |
| CHECKCONSTRAINTS | Vérifie les contraintes FOREIGN KEY et CHECK              |

Opération indispensable pour pouvoir déceler une base endommagée.

À faire régulièrement (1 fois par jour par exemple)

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Ces opérations s'effectuent sur une copie des données générée dans la tempdb et ne gêne pas la production. L'option ESTIMATEONLY permet de savoir quelle volumétrie de la base tempdb sera utilisée.

#### Syntaxes :

```
DBCC CHECKALLOC [( database_name | database_id | 0 [, NOINDEX ] )
  [ WITH { [ ALL_ERRORMSG ] [, NO_INFOMSGS ] [, TABLOCK ] [, ESTIMATEONLY ] } ] ]
DBCC CHECKCATALOG [( database_name | database_id | 0 )
  [ WITH NO_INFOMSGS ] ]
DBCC CHECKTABLE ( table_name | view_name [, { NOINDEX | index_id } ] )
  [ WITH { [ ALL_ERRORMSG ] [, NO_INFOMSGS ] [, TABLOCK ] [, ESTIMATEONLY ] [, { PHYSICAL_ONLY |
  DATA_PURITY } ] } ] ]
DBCC CHECKFILEGROUP [ [ ( { filegroup | filegroup_id | 0 } [, NOINDEX ] ) )
  [ WITH { [ ALL_ERRORMSG | NO_INFOMSGS ] [, TABLOCK ] [, ESTIMATEONLY ] [, PHYSICAL_ONLY ] } ] ]
DBCC CHECKDB [ [ ( database_name | database_id | 0 [, NOINDEX ] ) )
  [ WITH { [ ALL_ERRORMSG | NO_INFOMSGS ] [, TABLOCK ] [, ESTIMATEONLY ] [, PHYSICAL_ONLY ] } ] ]
DBCC CHECKCONSTRAINTS [ ( table_name | table_id | constraint_name | constraint_id )
  [ WITH { [ ALL_CONSTRAINTS | ALL_ERRORMSG ] } ] [, ] [ NO_INFOMSGS ] ]
```

**NOTA** : les options PHYSICAL\_ONLY | DATA\_PURITY permettent de vérifier plus ou moins profondément :

- PHYSICAL\_ONLY : (moins profond) vérification limitée à l'intégrité physique des pages et des en-têtes de ligne
- DATA\_PURITY : vérifie que certaines limites ne sont pas dépassées en fonction du type de données

CHECKFILEGROUP et CHECKDB effectuent successivement un CHECKALLOC + CHECKCATALOG + CHECKTABLE sur toutes les tables de la base (y compris vues indexées et tables interne de Service Broker)

CHECKDB vérifie de manière parallèle les tables. Il est possible de désactiver ce parallélisme à l'aide de l'indicateur de trace 2528.

Sous Oracle l'équivalent de cette fonctionnalité de vérification est l'utilitaire en ligne de commande DBVERIFY.

**NOTA** : DBCC CHECKIDENT contrôle le comportement des autoincrément de type IDENTITY

## 4 / L'utilitaire DBCC

62

### Réparation base endommagée

Lire le diagnostic de DBCC, **mais...**

- Ne vous précipitez pas sur les commandes REPAIR !
- Commencez par revoir le support (disques)
- Déplacer la base si besoin
- Si l'objet endommagé est un index non clustered, supprimez-le et recréez-le
- Si c'est une table vous pouvez peut-être récupérer les pages endommagées (RESTORE PAGE ...)

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les commandes :

- DBCC CHECKTABLE
- DBCC CHECKALLOC
- DBCC CHECKDB

Proposent l'option de réparation qui peut être

- REPAIR\_ALLOW\_DATA\_LOSS : tente de réparer les données y compris par suppression des erreurs irrécupérables
- REPAIR\_FAST : obsolète
- REPAIR\_REBUILD : tente de récupérer les données endommagées sans perte de données

Article à lire : <http://blog.developpez.com/sqlpro/files/2015/08/Correction-des-bases-de-donnees-SQL-Server-corrompues.pdf>

La récupération de page endommagées se fait à l'aide d'une restauration de page. Voir le chapitre consacré aux sauvegardes et restaurations.

## 4 / L'utilitaire DBCC

63

### Réparation base endommagée

Si REPAIR\_ALLOW\_DATA\_LOSS est utilisé :

- Vérifiez les contraintes (notamment FK) sur les tables réparées
- Supprimez les lignes orphelines à l'aide de requêtes SQL (semi anti-jointure gauche)
  - Vous devrez probablement désactiver préalablement les contraintes FK

Tentez de récupérer les données perdues depuis d'anciennes sauvegardes

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

SQL Server permet de restaurer une base à côté de celle de production. Il ne vous suffit plus que de comparer à coup de requête ou via un utilitaire comme TableDiff les données qui peuvent avoir été perdues pour les réinjecter

## 4 / L'utilitaire DBCC

64

### Réparation automatique de base endommagée

Dans le cas où vous auriez installé un PRA/PCA via :

- Le mirroring de bases de données
- AlwaysOn

Ce service assure la réparation automatique des bases.

Contrôlez la table :

`msdb.dbo.suspect_pages`

Pour savoir si des opérations de réparation ont eu lieu

Dans le cas où des pages ont été réparées, intéressez-vous à la santé de vos disques ou pire... à celle du contrôleur.



## 4 / L'utilitaire DBCC

65

### Jouer avec le cache :

| DBCC             | Description  |
|------------------|--|
| PROCCACHE        | Informations relatives au cache des procédures           |
| SQLPERF          | Informations diverses suivant paramètre                  |
| MEMORYSTATUS     | Informations des strates des différents niveaux de cache |
| SQLMGRSTATS      | Information de performance du cache                      |
| DROPCLEANBUFFERS | Vide le cache des données (sauf pages sales)             |
| FREESYSTEMCACHE  | Vide le cache système                                    |
| FREEPROCCACHE    | Vide le cache des procédures                             |
| FLUSHPROCINDB    | Vide le cache des procédure d'une base particulière      |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

#### Syntaxes :

DBCC PROCCACHE [ WITH NO\_INFOMSGS ]

DBCC FREEPROCCACHE [ ( { plan\_handle | sql\_handle | pool\_name } ) ] [ WITH NO\_INFOMSGS ]

FLUSHPROCINDB (db\_id)

SQLPERF (*paramètre*)

#### Paramètres pour SQLPERF :

- UMSSTATS : SQL Server thread management.
- WAITSTATS : wait types for SQL Server resources.
- IOSTATS : outstanding SQL Server reads and writes.
- RASTATS : SQL Server read-ahead activity.
- THREADS : I/O, CPU, and memory usage per SQL Server thread.

#### Pour vider la mémoire de SQL Server :

CHECKPOINT;

DBCC DROPCLEANBUFFERS;

DBCC FREEPROCCACHE;

## 4 / L'utilitaire DBCC

66

### Modifier le comportement du moteur :

De nombreux « flags » aussi appelés indicateur de trace permettent de modifier le comportement du moteur SQL.

Soit globalement, à l'aide des commandes :

| DBCC        | Description                      |
|-------------|----------------------------------|
| TRACEON     | Met en place une trace           |
| TRACEOFF    | Supprime une trace mise en place |
| TRACESTATUS | Informe des traces présentes     |

Soit localement dans une requête (clause OPTION)

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Syntaxes :

```
DBCC TRACEON ( trace# [ ,...n ] [ , -1 ] ) [ WITH NO_INFOMSGS ]
```

```
DBCC TRACEOFF ( trace# [ ,...n ] [ , -1 ] ) [ WITH NO_INFOMSGS ]
```

```
DBCC TRACESTATUS ( ( [ trace# [ ,...n ] ] [ , ] [ -1 ] ] ) [ WITH NO_INFOMSGS ]
```

**NOTA** : certains indicateurs de trace sont locaux (session) et d'autres globaux. Le paramètre -1 permet fixer la trace globalement (pour celles qui ne sont actives que pour la session)

On peut aussi fixer la trace pour le serveur en ajoutant dans la ligne de commande de lancement du service SQL Server l'option de démarrage -T suivie des n° de traces que l'on veut forcer.

## 4 / L'utilitaire DBCC

67

### Flags (indicateurs de trace) :

| DBCC |  |
|------|--|
| 3604 | Revoie les informations de certaines commandes DBCC vers la console        |
| 3605 | Revoie les informations de certaines commandes DBCC vers l'ERRORLOG        |
| 1211 | Désactive l'escalade de verrous en fonction de la pression mémoire         |
| 1224 | Désactive l'escalade de verrous en fonction du nombre de verrous           |
| 1204 | Alimente l'ERRORLOG avec les information de deadlock                       |
| 1222 | Ajoute des informations de forme XML dans l'ERRORLOG pour les deadlocks    |
| 610  | Minimalise les entrées du journal pour les opérations BULK                 |
| 834  | Utilise des pages « larges » pour l'allocation mémoire (64 bits seulement) |
| 835  | Verrouillage des pages mémoire (nécessite une autorisation système)        |

## 4 / L'utilitaire DBCC

68

### Flags (indicateurs de trace) :

| DBCC |   |
|------|---|
| 2528 | Désactive le parallélisme de vérification des commandes CHECK                     |
| 1806 | Désactive l'« instant file initialization » si elle est définie au niveau système |
| 3014 | Complète l'ERRORLOG d'info. supplémentaires pour les sauve.s/restau.              |
| 3226 | Évite les informations de sauvegardes de l'ERRORLOG                               |
| 3502 | Écrit des informations dans l'ERRORLOG sur les CHECKPOINT                         |
| 1118 | Interdit l'usage d'extensions mixtes  |
| 2371 | Change le seuil de déclenchement du recalcul des statistiques                     |
| 9481 | Utilise l'ancien estimateur de cardinalité (pour les bases de version >=2014)     |
| 2313 | Utilise le nouvel estimateur de cardinalité (pour les bases de version <2014)     |
| 1117 | Oblige tous les fichiers d'un même groupe à croître identiquement                 |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

En savoir plus :

[https://victorisakov.files.wordpress.com/2011/10/sql\\_pass\\_summit\\_2011-important\\_trace\\_flags\\_that\\_every\\_dba\\_should\\_know-victor\\_isakov.pdf](https://victorisakov.files.wordpress.com/2011/10/sql_pass_summit_2011-important_trace_flags_that_every_dba_should_know-victor_isakov.pdf)  
<http://sqlturbo.com/the-most-important-trace-flags-for-sql-server/>  
<http://www.sqlservice.se/updated-microsoft-sql-server-trace-flag-list/>

## 4 / Les index

69

Index relationnels :

- B-tree+ (arbres équilibrés chaînés)
- Hash (limité aux tables in memory)
- BW-Tree (B Tree pour tables « in memory »)

Index Analytiques :

- ColumnStore (indexation « verticale » - pour OLTP et OLAP)

Index particuliers :

- XML
- Géo (SIG)
- Textuels (« full text »)

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

SQL Server n'implémente pas d'index de type BITMAP (facilement simulable par une table de référence) ni d'index en hachage pour les tables relationnelles (facilement simulable par une colonne calculée persistante indexée).

Ces deux types d'index ne pouvant d'ailleurs servir que pour des opérations de recherches par égalité.

B-Tree+ => « balanced tree » avec chaînage des nœuds

BW-Tree => index B-tree sans verrou qui exploite la structure du journal de transaction en mémoire via un magasin de paires clé/valeur versionné. Il est spécialement conçu pour tirer partie des architecture multicœurs et du stockage SSD.

[https://www.researchgate.net/publication/266656619\\_Indexing\\_on\\_modern\\_hardware\\_Hekaton\\_and\\_beyond](https://www.researchgate.net/publication/266656619_Indexing_on_modern_hardware_Hekaton_and_beyond)

## 4 / Les index

70

| <b>Définition :</b>                        | <i>explications</i>                                      |
|--|--|
| <b>Structure de données</b>                | <i>Ce n'est pas une table</i>                            |
| <b>redondante</b>                          | <i>Copie des informations</i>                            |
| <b>spécialement organisée</b>              | <i>Les lignes d'un index sont « triées »</i>             |
| <b>pour accélérer certaines recherches</b> | <i>Toutes les recherches en bénéficient pas de gains</i> |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

La structure d'un index et la façon de ranger les données obéissent à l'algorithme utilisé. Dans notre cas c'est l'arbre équilibré chaîné.

Le principe algorithmique général de l'indexation fait partie de la classe des algorithmes « diviser pour régner »

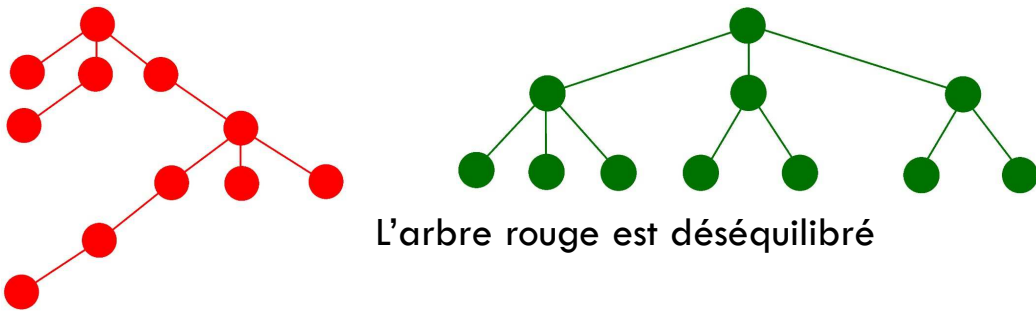
Comme les tables, les index sont constituées de pages qui ne diffèrent que par leur contenu et leur rangement

## 4 / Les index

71

Arbre équilibré :

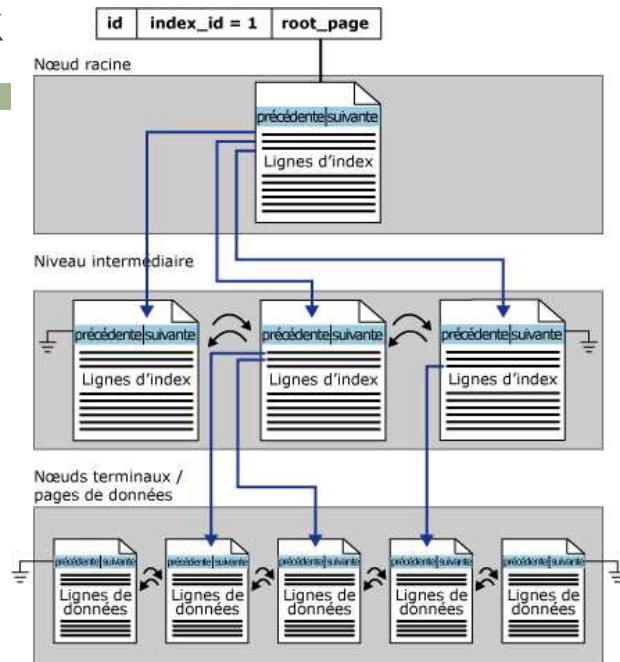
Arbre dont la distance entre la racine et les feuilles est identique pour toute feuille (en vert)



# 4 / Les index

72

## B-Tree+ : chainage des pages





## 4 / Les index

73

Index relationnels : B-tree+ (arbres équilibrés chaînés)

Utilisés pour la recherche d'information par :

- =
- > < >= <=
- BETWEEN
- LIKE 'blablabla%'

La clause GROUP BY

La clause ORDER BY

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

On dit qu'un index est « sargable » (cherchable) lorsque le prédicat de filtrage peut utiliser pleinement la recherche.

Les opérateurs de comparaison ici montrés sont tous « sargable » y compris toutes combinaisons avec l'opérateur logique ET

Ne sont pas « sargable » !

- <>
- LIKE '%blabla'
- LIKE 'bla%bla'
- IN (...)

Et toutes combinaisons avec NOT et OR

Néanmoins certains peuvent être rendus recherchables par le biais de l'indexation de colonnes calculées.

## 4 / Les index

74

Index relationnel, 2 types...

**CLUSTERED** : c'est la table triée par la clef d'index

- Un seul index clustered par table (*c'est la table !*)

**NONCLUSTERED** : index avec copie des données.

- Nécessite un pointeur de renvoi pour revenir à la ligne originelle de la table
  - Soit un « rowid » (en l'absence d'index clustered)
  - Soit la valeur de la clef clustered (si index clustered)

Une table peut donc être :

- En HEAP (tas) les lignes n'ont aucun ordre particulier
- CLUSTERED les lignes sont triées par la clef d'index

La clef d'un index est la liste des colonnes pour lesquelles la recherche est prévue.

La stratégie du moteur SQL Server (Sybase à l'origine) est optimisée pour l'utilisation massive des index clustered. Microsoft recommande donc que :

- Chaque table ait une clef primaire
- Chaque table ait un index clustered

Mais attention, le choix de l'index clustered est délicat

## 4 / Les index

75

ROWID, contient 3 informations :

- N° de fichier dans la base (file\_id)
- N° de page dans le fichier
- N° de slot de ligne dans la page

Une page est souvent indiquée de la façon suivante :

<file\_id>:<page\_id>

Exemple :

1:113

1:113 signifie fichier 1 page 113

## 4 / Les index

76

Une clef d'index clustered, devrait...

- être unique (si ce n'est pas le cas, SQL Server rajoute un « UNIQUIFIER »)
- être NOT NULL (si ce n'est pas le cas, SQL Server rajoute un « UNIQUIFIER »)
- être la plus petite possible (puisque'elle servira pour tous les index non clustered)
- ne pas pouvoir être modifiée (c'est le cas si donnée asémantique)

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

La clef primaire est par nature UNIQUE et NOT NULL, c'est pourquoi par défaut SQL Server y implante un index clustered.

La clef clustered ne devrait jamais dépasser la longueur du mot du processeur (64 bits soit 8 octets pour les OS modernes x64)

La modification de la valeur d'une clef clustered oblige à modifier tous les index qui référence cette ligne !

Si en plus :

- la clef est composée d'une seule colonne
- le type de données est non littérale (à cause des collations)
- les valeurs de cette clef sont monotones

C'est le top !

**Donc, un autoincrément est parfait !**

## 4 / Les index

77

### Création d'un index, syntaxe :

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX nom_index
  ON <objet> ( <liste_colonne_clef> )
  [ INCLUDE ( <liste_colonne_incluses> ]
  [ WHERE <predicat_filtirage> ]
  [ WITH ( <liste_options> ) ]
  [ ON { nom_schéma_partition ( colonne_partitionnée )
        | nom_groupe_fichier
        | default } ]

<liste_colonne_clef> ::=
  colonne1 [ ASC | DESC ] [ , colonne2 [ ASC | DESC ] [ ... ] ]

<liste_colonne_incluses> ::=
  colonne1 [ , colonne2 [ ... ] ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les noms des index doivent être uniques dans le schéma SQL dans lequel repose la table

Une même colonne ne peut apparaître à la fois dans la clef et dans la liste de colonnes de la clause INCLUDE

- UNIQUE : équivalente à la contrainte UNIQUE
- CLUSTERED : l'index est clustered (il ne peut y en avoir qu'un, puisque c'est la table organisée sous forme d'index)
- NONCLUSTERED : valeur par défaut

En l'absence de spécification de la clause ON, l'index est créé dans le groupe de fichier dans lequel se trouve la table. Si l'on spécifie DEFAULT, alors le groupe de fichier par défaut est choisi (qui peut être différent de celui de la table);

Pour la clef d'index

- ASC signifie ASCENDING (tri ascendant (par défaut)
- DESC signifie DESCENDING (tri descendant à définir)

DESC est un choix intéressant pour les données temporelles lorsqu'elle sont « noyées » dans une clef multicolonne

## 4 / Les index

78

Index couvrant (clause INCLUDE) :

La plupart du temps, la lecture seule de l'index ne suffit pas à « couvrir » la requête.

- Il faut faire une recherche dans l'index (seek)
- Puis une recherche (seek, lookup) ou balayage (scan) dans la table

La clause INCLUDE permet d'éviter cette double lecture.

On peut voir l'index CLUSTERED comme un super index incluant toutes les colonnes de la table !

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Exemple :

```
SELECT nom, prenom, date_naissance
FROM   personne
WHERE  date_naissance >= '2000-01-01'
```

Si présence d'un index composé de la manière suivante :

```
CREATE INDEX X_prs_dn
ON   personne (date_naissance)
INCLUDE (nom, prenom);
```

Alors il n'y aura qu'un seul accès, celui de l'index pour renvoyer les résultats attendus.

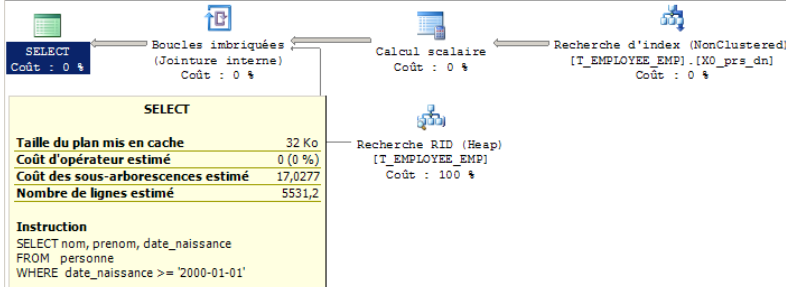
**ATTENTION** : par nature un index CLUSTERED ne peut pas avoir de clause INCLUDE car il possède déjà toutes les colonnes de la table !

# 4 / Les index

79

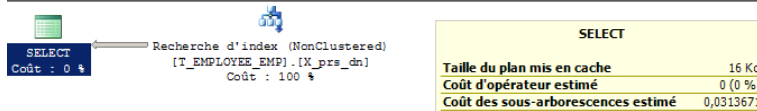
## Index couvrant (clause INCLUDE), plans de requête :

Requête 1 : coût de requête (relatif au lot) : 100%  
SELECT nom, prenom, date\_naissance FROM personne WHERE date\_naissance >= '2000-01-01'



Sans  
INCLUDE

Requête 1 : coût de requête (relatif au lot) : 100%  
SELECT nom, prenom, date\_naissance FROM personne WHERE date\_naissance >= '2000-01-01'



Avec  
INCLUDE

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Exemple :

```
SELECT nom, prenom, date_naissance  
FROM personne  
WHERE date_naissance >= '2000-01-01'
```

La première requête utilise l'index suivant :

```
CREATE INDEX X1_prs_dn  
ON personne (date_naissance);
```

La seconde utilise l'index suivant :

```
CREATE INDEX X_prs_dn  
ON personne (date_naissance)  
INCLUDE (nom, prenom);
```

Le gain est de 543 fois ! (table de 1 253 000 lignes, 2 183 lignes retournées)

**NOTA** : obtenez le plan de requête estimé via le menu « Requête » / « Afficher le plan d'exécution estimé » ou par l'icône correspondante.

## 4 / Les index

80

### Index filtrés (clause WHERE)

Il est parfois inutile d'indexer toutes les valeurs, soit :

- parce que certaines valeurs sont sur-représentées
- parce que certaines valeurs sont rarement cherchées
- pour un cas particulier de traitement
- pour simuler une contrainte d'unicité normative

SQL Server n'est pas conforme à la norme sur les index uniques :

- Une seule « valeur » NULL permise !

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Exemple de valeurs sur représentées : les groupes sanguins

Exemple de valeurs rarement cherchés : les factures datant de plus de 10 ans

Exemple de cas particulier : étude statistique à mener sur le cancer la prostate en région parisienne pour des personnes nées dans les années 60

Exemple de contrainte unique : immatriculation de véhicule, numéro de sécurité sociale  
Pour ce dernier cas :

```
CREATE UNIQUE INDEX X_PAT_NUMSECU  
ON T_PATIENT_PAT (PAT_NUMSECU)  
WHERE PAT_NUMSECU IS NOT NULL;
```

NOTA : les possibilité du prédicat WHERE de la commande CREATE INDEX sont limitées



## 4 / Les index

81

### Index multicolonne :

La clef d'index peut contenir jusqu'à 16 colonnes, mais ne doit pas dépasser 900 octets.

Quelle est l'efficacité d'un tel index ?

|      | ADR_NOM    | ADR_PRENOM | ADR_DATE_NAISSANCE |
|------|------------|------------|--------------------|
| 171  | ADAM       | Bernard    | 1928-06-20         |
| 172  | ADAM       | JACQUELINE | 1938-12-08         |
| 173  | ADAM       | LOUISE     | 1944-01-22         |
| 174  | ADAM       | ODETTE     | NULL               |
| 175  | ADAM       | PIERRE     | NULL               |
| 176  | ADAM       | RAYMONDE   | 1930-01-11         |
| 177  | ADAM       | RENE       | 1934-07-05         |
| 178  | ADAM       | SYLVIE     | 1935-05-05         |
| 5... | LECONTE    | BERNARD    | NULL               |
| 5... | LECONTE    | CHRISTIANE | 1930-09-05         |
| 5... | LECONTE    | LOUISE     | 1935-08-19         |
| 5... | LECONTE    | RENEE      | 1929-01-17         |
| 5... | LECONTE    | ROGER      | 1929-11-06         |
| 5... | LECONTE    | SYLVIE     | 1954-03-21         |
| 9... | ZIMMERMANN | BERNARD    | 1963-01-07         |
| 9... | ZIMMERMANN | CHRISTIANE | 1956-01-27         |
| 9... | ZIMMERMANN | LOUISE     | 1939-10-09         |
| 9... | ZIMMERMANN | SYLVIE     | 1957-02-16         |

En fait les informations d'un index multicolonne sont « vectorisées ». Chaque colonne suivante précise la colonne précédente.

## 4 / Les index

|      | ADR_NOM    | ADR_PRENOM | ADR_DATE_NAISSANCE |
|------|------------|------------|--------------------|
| 171  | ADAM       | Bernard    | 1928-06-20         |
| 172  | ADAM       | JACQUELINE | 1938-12-08         |
| 173  | ADAM       | LOUISE     | 1944-01-22         |
| 174  | ADAM       | ODETTE     | NULL               |
| 175  | ADAM       | PIERRE     | NULL               |
| 176  | ADAM       | RAYMONDE   | 1930-01-11         |
| 177  | ADAM       | RENE       | 1934-07-05         |
| 178  | ADAM       | RENE       | 1935-05-05         |
| 5... | LECONTE    | BERNARD    | NULL               |
| 5... | LECONTE    | CHRISTIANE | 1930-09-05         |
| 5... | LECONTE    | LOUISE     | 1935-08-19         |
| 5... | LECONTE    | RENEE      | 1929-01-17         |
| 5... | LECONTE    | RENEE      | 1929-11-06         |
| 5... | LECONTE    | SYLVIE     | 1954-03-21         |
| 9... | ZIMMERMANN | BERNARD    | 1930-01-07         |
| 9... | ZIMMERMANN | CHRISTIANE | 1936-01-27         |
| 9... | ZIMMERMANN | LOUISE     | 1939-10-09         |
| 9... | ZIMMERMANN | SYLVIE     | 1942-02-16         |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les statistiques des index multi colonnes ne portent que sur la première colonne.  
 Vous devez faire en sorte que ce soit la plus discriminante, dans la mesure du possible.

## 4 / Les index

83

### Options physiques utilisées lors de la création des index

- FILLFACTOR = *fillfactor*
- PAD\_INDEX = { ON | **OFF** }
- SORT\_IN\_TEMPDB = { ON | **OFF** }
- DROP\_EXISTING = { ON | **OFF** }
- ONLINE = { ON | **OFF** }
- STATISTICS\_NORECOMPUTE = { ON | **OFF** }
- MAXDOP = *max\_degree\_of\_parallelism*
- DATA\_COMPRESSION = { **NONE** | ROW | PAGE }

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les options de création d'index doivent être spécifiés dans une clause WITH

FILLFACTOR : facteur de remplissage des pages d'index (inverse du PCTFREE d'Oracle)  
Permet de laisser un peu de « mou » dans les pages feuilles de l'index afin d'éviter trop de fragmentation

PAD\_INDEX : à ON, report du FILLFACTOR au niveau des pages non feuille (pages de navigation)

SORT\_IN\_TEMPDB : le tri pour construire l'index est exclusivement exécuté dans la base tempdb et non en mémoire

DROP\_EXISTING : sert à recréer un index existant (il doit avoir les mêmes caractéristiques logiques)

ONLINE : (édition Entreprise seulement) évite de bloquer la table pendant la création de l'index

STATISTICS\_NORECOMPUTE : les statistiques créées avec l'index ne seront plus jamais réévaluées

MAXDOP : limite le parallélisme lors de la création de l'index

DATA\_COMPRESSION : active la compression de niveau ligne ou page (cette dernière inclut la compression au niveau ligne)

## 4 / Les index

84

Options physiques utilisées lors de l'usage des index

- ALLOW\_ROW\_LOCKS = { **ON** | OFF }
- ALLOW\_PAGE\_LOCKS = { **ON** | OFF }
- IGNORE\_DUP\_KEY = { ON | **OFF** }

```
CREATE INDEX X_PAT_NOM_PRENOM
ON T_PATIENT_PAT (PAT_NOM, PAT_PRENOM)
WITH (FILLFACTOR = 95,
      ONLINE = ON);
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

ALLOW\_ROW\_LOCKS : autorise ou non, l'usage de verrous de ligne

ALLOW\_PAGE\_LOCKS : autorise ou non, l'usage de verrous de page

IGNORE\_DUP\_KEY : à ON, rejette les lignes en doublons dans un index unique sans annuler la transaction

## 4 / Les index « columnstore »

85

Index « columnstore » : indexation verticale des informations obtenues à l'aide d'un algorithme de hachage

- Rend la table « READ ONLY »
- Réservé à l'OLAP

Syntaxe :

```
CREATE COLUMNSTORE INDEX nom_index
    ON nom_table ( <liste_colonne_clef> )
    [ WITH ( <liste_options> ) ]
    [ ON <destination> ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

La limitation « READ ONLY » de la table devrait être levée avec la version 2016.

Un index columnstore ne peut pas contenir plus de 1024 colonnes et on ne peut pas indexer un type LOB

La liste des colonnes clef d'un index columnstore n'accepte pas le sens de tri (ASC / DESC)

Les options sont les suivantes :

- DROP\_EXISTING = { ON | OFF }
- MAXDOP = max\_degree\_of\_parallelism }

La destination est, soit :

- Un nom de groupe de fichiers
- Le mot clef DEFAULT pour designer le groupe de fichiers par défaut de la base
- Un schéma de partitionnement

En sus des index columnstore, SQL Server propose des index en hash pour les tables « In Memory »

## 4 / Les index « columnstore »

86

### Index ROW...

| PAGE (B-Tree+)<br>(index relationnel) |         |
|---------------------------------------|---------|
| nom                                   | prénom  |
| DUPONT                                | Alain   |
| DUPONT                                | Marcel  |
| DUPONT                                | Zoé     |
| MARTIN                                | Alain   |
| MARTIN                                | Yves    |
| WAGNER                                | Bernard |
| WAGNER                                | Zoé     |

Indexation de lignes  
« vectorisées »



Indexation de colonnes  
indépendantes

### Index COLUMN...

| PAGE<br>(columnstore)<br>(index OLAP) | PAGE<br>(columnstore)<br>(index OLAP) |
|---------------------------------------|---------------------------------------|
| nom                                   | prénom                                |
| DUPONT                                | Alain                                 |
| DUPONT                                | Alain                                 |
| DUPONT                                | Bernard                               |
| MARTIN                                | Marcel                                |
| MARTIN                                | Yves                                  |
| WAGNER                                | Zoé                                   |
| WAGNER                                | Zoé                                   |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

L'indexation des colonnes est compressée par une technique de hachage associée à un dictionnaire.

Avantage : compacité de l'index (compression), performances pour des recherches multiples

Inconvénient : très coûteux en mise à jour (ce qui explique le « READ ONLY »)

## 4 / Vue indexée

87

Ordinairement une vue ne contient pas de données.

- seule les tables contiennent les données.
- une vue est une représentation logique des données des tables

Une vue indexée est une vue sur laquelle on « matérialise » les données par la création d'un index de vue.

Les données de cette vue sont synchrones par rapport aux données des tables qui la compose

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les vues indexées comportent de nombreuses limitation car il doit y avoir déterminisme et surjection de la vue par rapport aux tables qui la compose, autrement dit : **tout ligne de la vue possède au moins une ligne dans la table réponse formée par la requête**

Ceci induit des jointures interne et interdit :

- Les jointures externes
- Les sous requêtes
- Les requêtes ensemblistes (UNION, INTERSECT EXCEPT)
- Les opérations d'agrégations autres que COUNT\_BIG et SUM
- L'utilisation de fonction table multi instruction
- Les fonctions « OPEN »
- L'opérateur DISTINCT
- APPLY, PIVOT/UNPIVOT, OFFSET, CONTAINS, FREETEXT...

## 4 / Vue indexée

88

Étape 1, créer la vue avec la directive SCHEMABINDING :

```
CREATE VIEW nom_vue  
WITH SCHEMABINDING  
AS
```

```
    SELECT ...
```

Étape 2, créer un index UNIQUE CLUSTERED pour matérialiser la vue

```
CREATE UNIQUE CLUSTERED INDEX nom_index  
    ON nom_vue (liste_colonne);
```

Étape 3 (facultative), créez autant d'index « secondaires » que vous voulez !

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

**ATTENTION** : les tables indiquée dans la requête de construction de la vue doivent être exprimées avec leur schéma de manière explicite.

SCHEMABINDING indique aux tables participant à la vue que tout changement de structure y est proscrit.

En édition Enterprise, l'optimiseur fait la substitution mathématique de toute requête à la vue s'il est possible de gagner par ce biais.

Dans les autres éditions, cette substitution n'est pas automatique. Il faut la forcer en ajoutant explicitement le tag (indicateur ou « hint ») de table « NOEXPAND » à la spécification de la vue dans la requête finale.



## 4 / Les index, diagnostic

89

Il est possible de diagnostiquer les index :

- Connaître ceux à poser (1)
- Connaître l'usage de chaque index (1)
- Voir la fragmentation des index

**(1) ATTENTION : pour ces deux diagnostics, il s'agit d'information mémoire, volatile, acquises depuis le dernier démarrage de l'instance.**

*Ayez au moins un recul de plus de 31 jours !*

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Pour connaître la date heure du dernier démarrage de l'instance, exécutez la requête suivante :

```
SELECT sqlserver_start_time FROM sys.dm_os_sys_info.
```

NOTA : obtenez la liste des index d'une base par la vue sys.indexes et les colonnes indexées par la vue sys.index\_columns à joindre avec le vue sys.columns

## 4 / Les index, diagnostic

90

### Connaître ceux à poser

| DMV                                 | Description                            |
|-------------------------------------|--|
| sys.dm_db_missing_index_details     | Liste des index estimés manquants      |
| sys.dm_db_missing_index_group_stats | Statistiques sur les index manquants   |
| sys.dm_db_missing_index_groups      | Vue de jointure pour les 2 précédentes |
| sys.dm_db_missing_index_columns()   | Détails sur les colonnes               |

```
SELECT mid.*, user_seeks, user_scans, avg_user_impact
FROM sys.dm_db_missing_index_details AS mid
JOIN sys.dm_db_missing_index_groups AS g
ON mid.index_handle = g.index_handle
JOIN sys.dm_db_missing_index_group_stats AS s
ON g.index_group_handle = s.group_handle
ORDER BY user_seeks * avg_user_impact DESC;
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Il ne faut jamais créer tous les index demandés car beaucoup se recourent

Deux approches possibles :

- Par mutualisation (index couvrants, redondants à la clause INCLUDE près...)
- Par estimation d'usage (avg\_user\_impact, user\_seeks...)

Ne jamais recréer la table « à l'envers » avec une demande INCLUDE reprenant toutes les autres colonnes de la table .

La requête donne dans l'ordre les index les plus intéressants à rajouter par estimation d'utilité.

## 4 / Les index, diagnostic

91

Ceux à supprimer : sys.dm\_db\_index\_usage\_stats

À joindre avec les objets de la base

```
SELECT s.name AS TABLE_SCHEMA, o.name AS TABLE_NAME,  
       i.name AS INDEX_NAME, ius.*  
FROM   sys.dm_db_index_usage_stats AS ius  
       JOIN sys.objects AS o  
           ON ius.object_id = o.object_id  
       JOIN sys.schemas AS s  
           ON o.schema_id = s.schema_id  
       JOIN sys.indexes AS i  
           ON ius.object_id = i.object_id  
           AND ius.index_id = i.index_id  
WHERE  ius.database_id = DB_ID()  
ORDER BY user_scans, user_lookups, user_seeks, user_updates DESC
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Cette requête ordonne les index des plus pénalisant aux plus profitables.

Un index est pénalisant s'il ne sert aucunement en recherche mais doit être mis à jour.

La DMV suivante sys.dm\_db\_index\_operational\_stats, donne l'utilisation physiques des index (IO), contrairement à sys.dm\_db\_index\_usage\_stats qui donne l'utilisation logique.

## 4 / Les index, diagnostic

92

Estimer la fragmentation :

`sys.dm_db_index_physical_stats(...)`

La fragmentation des index pénalise :

- très peu les recherches (seek)
- un peu plus les balayage (scan)
- beaucoup la volumétrie

Elle doit être réparée régulièrement

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Exemple de requête d'analyse de fragmentation :

```
SELECT s.name AS TABLE_SCHEMA, o.name AS TABLE_NAME,
       i.name AS INDEX_NAME, ips.*
FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, NULL) AS
ips
JOIN sys.objects AS o
  ON ips.object_id = o.object_id
JOIN sys.schemas AS s
  ON o.schema_id = s.schema_id
JOIN sys.indexes AS i
  ON ips.object_id = i.object_id
AND ips.index_id = i.index_id
WHERE ips.database_id = DB_ID()
AND page_count > 64;
```

NOTA : un index ne comportant que peu de pages sera toujours vu comme fragmenté alors qu'il ne l'est pas

## 4 / Les index, diagnostic

93

### Défragmentation des index

Deux méthodes :

- REBUILD : reconstruit l'index (aucune fragmentation résiduelle), mais bloque la table (sauf ONLINE si ed. Enterprise) et **recalcule les statistiques**.
- REORGANIZE : réarrange l'index (fragmentation résiduelle), pas de blocage

Il est aussi possible de reconstruire une table fragmentée par REBUILD

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

### Syntaxes :

```
ALTER INDEX { nom_index | ALL }  
ON nom_table REBUILD  
[ WITH ( <rebuild_options> )
```

```
ALTER INDEX { nom_index | ALL }  
ON nom_table REBUILD  
REORGANIZE [ WITH ( LOB_COMPACTION = { ON | OFF } ) ]
```

```
ALTER TABLE nom_table REBUILD  
[ WITH ( <rebuild_options> )
```

**NOTA** il est possible de defragmenter au niveau de la partition. Syntaxe spécifique.

Les options sont sensiblement les mêmes que lors de la creation des index.

**IMPORTANT** : une défragmentation d'index est une opération de maintenance à réaliser couramment (quotidiennement) sauf pour de très grosses bases.

## 4 / Index XML

94

Permet d'accélérer les recherches dans les colonnes de type XML.

- Mieux encore si le XML est typé par une collection de schémas XML.

Il faut d'abord créer un index XML « primaire »

```
CREATE PRIMARY XML INDEX nom_index
ON <object> ( nom_colonne_xml )
[ WITH ( <liste_options_index_xml> ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Une collection de schémas XML se crée à l'aide de l'instruction CREATE XML SCHEMA... et recense une collection de documents de type XML SCHEMA définissant des documents XML de diverses natures.

Une colonne XML d'une table peut être créée avec un typage du XML via la collection précédemment créée.

Exemple :

```
CREATE XML SCHEMA dbo.XSC_CONTRAT
AS
```

```
'<?xml version="1.0" encoding="UTF-16"?>
```

```
<xsd:schema targetNamespace="http://sqlspot.com/contrat/assurance/xsdef" xmlns="http://sqlspot.com/contrat/assurance/xsdef/AssuranceVie"
```

```
elementFormDefault="qualified" attributeFormDefault="unqualified"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
```

```
<xsd:complexType ...
```

## 4 / Index XML

95

L'index primaire à un rôle similaire au « clustered » index d'une table

Vous pouvez ensuite créer les index secondaires suivants :

```
CREATE XML INDEX nom_index
    ON <object> ( nom_colonne_xml )
    USING XML INDEX nom_index_xml_primaire
    FOR { VALUE | PATH | PROPERTY }
    [ WITH ( <liste_options_index_xml> ) ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

La liste des options disponibles pour un index XML est la suivante :

- PAD\_INDEX = { ON | OFF }
- FILLFACTOR = fillfactor
- SORT\_IN\_TEMPDB = { ON | OFF }
- IGNORE\_DUP\_KEY = OFF
- DROP\_EXISTING = { ON | OFF }
- ONLINE = OFF
- ALLOW\_ROW\_LOCKS = { ON | OFF }
- ALLOW\_PAGE\_LOCKS = { ON | OFF }
- MAXDOP = max\_degree\_of\_parallelism

## 4 / Index XML

96

Rôle des index secondaires :

- L'index XML FOR VALUE optimise l'accès aux valeurs (indexation des valeurs et chemins/nœuds)
- L'index XML FOR PATH optimise les accès (indexation des chemins/nœuds)
- L'index XML FOR PROPERTY optimise les accès croisés aux clefs de la table et aux valeurs (indexation de la clef primaire, des valeurs et chemins/nœuds)

Nous vous conseillons de faire des essais et de mesurer les gains obtenus sur les IO et accessoirement sur les temps de réponse pour les grosses tables)

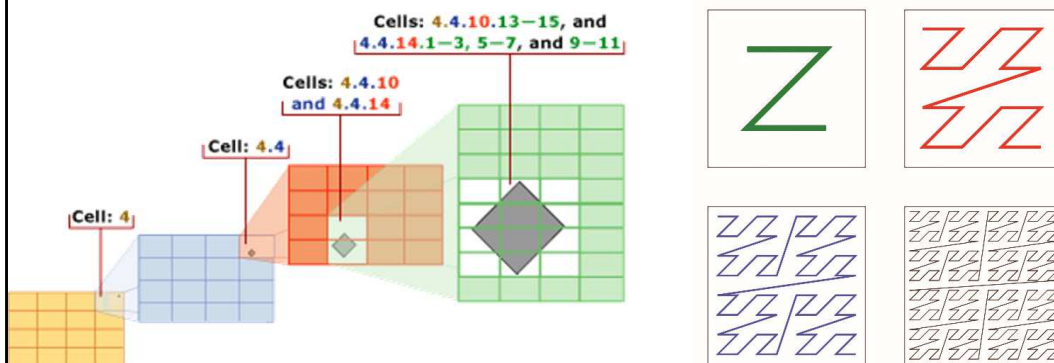


## 4 / Index spatiaux

97

Permet d'indexer les types geometry et geography.

Utilise un pavage (*tesselation* en anglais) multiniveau associé à une courbe de cheminement (de Lebesgues) en Z.



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Ces informations sont stockées dans un arbre équilibré de même nature que les index relationnels

Il y a 4 niveau de pavage avec des « mailles » de plus en plus serrées.

Pour les index géométrique, il faut limiter la zone d'indexation par une « bounding box ». En effet, la géométrie euclidienne fait que l'espace géométrique (ici le plan) est par nature infini ! Contrairement aux index spatiaux limités à la surface terrestre par nature finie !

## 4 / Index spatiaux

98

### Syntaxe :

```
CREATE SPATIAL INDEX nom_index
  ON nom_table ( nom_colonne_geo )
  [ USING GEOMETRY_[AUTO_]GRID ]
  WITH ( BOUNDING_BOX = ( xmin, ymin, xmax, ymax )
        [ GRIDS = ( { <liste_niveaux_grille> | <liste_niveaux> } ) ]
        [ CELLS_PER_OBJECT = n ]
        [ <options_index_spatial> ] )
  [ ON nom_groupe_fichier ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Si le pavage (tessellation) est automatique (GEOMETRY\_AUTO\_GRID), alors ne pas préciser la clause GRIDS. Dans le cas contraire (GEOMETRY\_GRID), la préciser.

Pour un index sur le type GEOGRAPHY, ne pas préciser la clause BOUNDING\_BOX.

Par nature le plan géométrique euclidien est infini. Indexer l'infini étant impossible, il faut définir un cadre (BOUNDING BOX).

Les options des index spatiaux sont les mêmes que ceux des index relationnels.

### Détails de la syntaxe :

```
<tessellation_automatique_geom> ::=
{ [ USING GEOMETRY_AUTO_GRID ]
  WITH ( BOUNDING_BOX = ( xmin, ymin, xmax, ymax )
        [ CELLS_PER_OBJECT ]
        [ <options_index_spatial> ] )
```

```
<tessellation_manuelle_geom> ::=
{ [ USING GEOMETRY_GRID ]
  WITH ( BOUNDING_BOX = ( xmin, ymin, xmax, ymax )
        [ GRIDS = ( { <liste_niveau_grille> | <liste_niveau> } ) ]
        [ CELLS_PER_OBJECT = n ]
        [ <options_index_spatial> ] )
```

```
<liste_niveau_grille> ::=
<liste_niveau_grille1>
[ , <liste_niveau_grille2>
[ , <liste_niveau_grille3>
[ , <liste_niveau_grille4> ] ] ]
```

```
<liste_niveau_grilleN> ::=
LEVEL_N = <taille_maille>
```

```
<liste_niveau> ::= <taille_maille1>, <taille_maille2>, <taille_maille3>, <taille_maille4>
```

```
<taille_maille> ::= { LOW | MEDIUM | HIGH }
```

La syntaxe complète est disponible à l'URL :

<https://msdn.microsoft.com/fr-fr/library/bb934196%28v=sql.120%29.aspx>

## 4 / Index spatiaux, exemple

99

```
CREATE SPATIAL INDEX XS_BATIMENT_PLANS
ON T_BATIMENT_BTM (BTM_PLAN_ETAGE)
USING GEOMETRY_GRID
WITH (BOUNDING_BOX = (xmin=0, ymin=0,
                      xmax=500, ymax=200),
      GRIDS = (LOW, LOW, MEDIUM, HIGH),
      CELLS_PER_OBJECT = 64,
      FILL_FACTOR = 75 );
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

### Détails de la syntaxe pour le type GEOGRAPHY :

```
<tesselation_automatique> ::=
{ [ USING GEOGRAPHY_AUTO_GRID ]
  [ WITH ( [ CELLS_PER_OBJECT ]
           [ <options_index_spatial> ] ) ] }
```

```
<tesselation_manuelle> ::=
{ [ USING GEOMETRY_GRID ]
  [ WITH ( [ GRIDS = ( { <liste_niveau_grille> | <liste_niveau> } ) ]
           [ CELLS_PER_OBJECT = n ]
           [ <options_index_spatial> ] ) ] }
```

```
<liste_niveau_grille> ::=
<liste_niveau_grille1>
[ , <liste_niveau_grille2>
  [ , <liste_niveau_grille3>
    [ , <liste_niveau_grille4> ] ] ]
```

```
<liste_niveau_grilleN> ::=
LEVEL_N = <taille_maille>
```

```
<liste_niveau> ::= <taille_maille1>, <taille_maille2>, <taille_maille3>, <taille_maille4>
```

```
<taille_maille> ::= { LOW | MEDIUM | HIGH }
```

Par nature l'espace géographique terrestre est fini. Il n'y a donc pas de cadre.

Pour la destination, groupe de fichier spécifique ou DEFAULT ou bien partition.

La syntaxe complète est disponible à l'URL :

<https://msdn.microsoft.com/fr-fr/library/bb934196%28v=sql.120%29.aspx>

## 4 / Index textuel

100

« Full Text Search » intégré au moteur SQL.

Création en deux phases :

Créer un catalogue d'indexation (stockage « logique »)

```
CREATE FULLTEXT CATALOG nom_catalogue
  [ ON FILEGROUP nom_groupe_fichier ]
  [ WITH ACCENT_SENSITIVITY = {ON|OFF} ]
  [ AS DEFAULT ]
  [ AUTHORIZATION nom_proprietaire ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

SQL Server permet d'indexer des documents électroniques de toutes natures lorsqu'ils sont conservés dans :

- une colonne de type BLOB (varbinary(max))
- un FILESTREAM
- une FILETABLE

Pour cela SQL Server utilise des DLL de type « ifilters » dont une trentaine est installée par défaut. La liste est accessible par la vue sys.fulltext\_document\_types.

Exemple de création de catalogue :

```
CREATE FULLTEXT CATALOG FTC_ENTREPRISE
  ON FILEGROUP FG_FULLTEXT
  WITH ACCENT_SENSITIVITY = ON
  AS DEFAULT;
GO
```

**ATTENTION** à la sensibilité aux accents, cela peut être important, par exemple pour chercher du **maïs** ou des **congrès** !

## 4 / Index textuel

101

2) Créer autant d'index que nécessaire :

```
CREATE FULLTEXT INDEX ON nom_table
  [ ( <liste_colonne_indexée> ) ]
  KEY INDEX nom_index_unique_non_null
  [ ON <destination> ]
  [ WITH [ ( ) <liste_options_ft> [ ) ] ] ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

### Détails de la syntaxe :

<liste\_colonne\_indexée> ::= <col\_indexée\_ft1> [, <col\_indexée\_ft2> [, ... ]]

<col\_indexée\_ftN> ::= *nom\_colonne*  
[ TYPE COLUMN *nom\_colonne\_type\_doc* ]  
[ LANGUAGE <langue> ]  
[ STATISTICAL\_SEMANTICS ]

La destination permet d'indiquer un catalogue ou un FILEGROUP ou les deux. Pour le groupe de fichier, précisez le mot clef FILEGROUP;

<liste\_options\_ft> ::= <options\_ft1> [, <options\_ft2> [, <options\_ft3> ]]

<liste\_options\_ftN> ::= CHANGE\_TRACKING [ = ] { MANUAL | AUTO | OFF [, NO POPULATION ] }  
| STOPLIST [ = ] { OFF | SYSTEM | *nom\_liste\_mots\_noirs* }  
| SEARCH PROPERTY LIST [ = ] *nom\_liste\_propriétés*

**Limitation** : un seul index de recherche textuelle n'est autorisé par table et ne doit pas dépasser 1024 colonnes.

## 4 / Index textuel

102

Pour la colonne indexée :

- TYPE COLUMN : permet d'indiquer une colonne précisant le type de document électronique contenu dans la colonne indexée
- LANGUAGE : permet de préciser la langue de la colonne indexée (LCID ou *name* de la vue `sys.fulltext_languages`)
- STATISTICAL\_SEMANTICS : crée un index spécifique à la recherche sémantique

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

TYPE COLUMN doit contenir le code d'extension des fichiers électroniques, tels que définis dans la vue `sys.fulltext_document_types`. Par exemples : `.doc`, `.html`, `.ppt`, `.rtf`, `.txt`, `.xls`, `.xml`...

Par défaut la langue est celle d'installation du serveur, sauf pour les document XML s'il y a une méta balise de langue (`xml:lang`)

STATISTICAL\_SEMANTICS (à partir de la version 2014) utilisé pour les recherches sémantiques.

## 4 / Index textuel

103

Alimentation de l'index sémantique :

- **CHANGE\_TRACKING** : spécifie comment les modifications sont intégrées à l'index textuel et comment il est créé
  
- **STOPLIST** : gestion de la liste des « mots noirs »
  
- **SEARCH PROPERTY LIST** : associe une liste de propriétés de recherche à l'index.

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

**CHANGE\_TRACKING** :

- **AUTO** : journalisée au file de l'eau puis indexée de manière asynchrone (valeur par défaut)
- **MANUAL** : journalisée au fil de l'eau et indexée manuellement en lançant la commande `ALTER FULLTEXT INDEX ...START UPDATE POPULATION` (peut être planifiée dans l'Agent SQL)
- **OFF** : non journalisée indexation effectuée dans la foulée
- **OFF NO POPULATION** : non journalisée indexation non effectuée. Il sera possible de remplir l'index par la commande `ALTER FULLTEXT INDEX` avec l'un des clauses `START FULL POPULATION` ou `START INCREMENTAL POPULATION`.

**STOPLIST** :

- **OFF** : aucune liste de mots noirs n'est associée aux recherches dans l'index textuel
- **SYSTEM** : la liste système par défaut de mots noirs, propre à la langue spécifiée, est utilisée
- **nom\_liste** : une liste particulière de mots noirs est utilisée

**NOTA** : SQL Server indexe tous les « mots » quelque soit leur nature (chaînes composées de lettres, chiffres...). Ce n'est qu'à la recherche que l'on peut ou pas, appliquer le filtrage des mots noirs. Vous pouvez créer vos propres listes de mots noirs à l'aide des commandes : `CREATE FULLTEXT STOPLIST ...` et `ALTER FULLTEXT STOPLIST ...`

Les listes de propriétés de recherches sont des listes de mots clés assurant des services de métadonnées dans les fichiers électroniques. Par exemple dans un document électronique Word, il existe des propriétés pour chaque document, comme l'auteur, la date de dernière modification, le nombre de pages...

## 4 / Index textuel

104

### Méthodes de recherches :

CONTAINS... cherche :

- | Fonction scalaire | Fonction table |
|-------------------|----------------|
| CONTAINS          | CONTAINSTABLE  |
| FREETEXT          | FREETEXTTABLE  |
- Un mot, une expression, un mot commençant par, un ensemble de mots...
  - Un mot à une certaine proximité d'un autre
  - La forme fléchié d'un mot (féminin, pluriel, conjugaison...)
  - Un synonyme (nécessite un thésaurus)
  - Une « expansion »
  - Une valeur de propriété
  - De multiples éléments enchainés par AND ou OR

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les synonymes et expansions, nécessite l'utilisation d'un thésaurus que l'utilisateur doit configurer en le remplissant de ses propres synonymes (spécifique à son univers sémantique) ou bien acheter. Voir :

<https://msdn.microsoft.com/fr-fr/library/ms142491%28v=sql.120%29.aspx>

L'emplacement des fichiers du thésaurus est obtenu par la requête suivante :

```
WITH T AS ( SELECT CAST(SERVERPROPERTY('instancedefaultdatapath') AS  
VARCHAR(2014)) AS REP)
```

```
SELECT SUBSTRING(REP, 1, LEN(REP) - 5) + 'FTDATA\'
```

```
FROM T;
```

Le fichier pour le français est tsfra.xml

Les valeurs de propriété ne concernant que les métadonnées des documents électroniques (Auteur, Titre, Volume, nombre de pages...)

Voir : [https://www.toadworld.com/platforms/sql-](https://www.toadworld.com/platforms/sql-server/b/weblog/archive/2013/09/26/using-property-lists-in-sql-2012-full-text-searches)

[server/b/weblog/archive/2013/09/26/using-property-lists-in-sql-2012-full-text-searches](https://www.toadworld.com/platforms/sql-server/b/weblog/archive/2013/09/26/using-property-lists-in-sql-2012-full-text-searches)

- CONTAINS : fonction scalaire avec recherches précises
- CONTAINSTABLE : fonction table avec recherche précises, affichage du score de recherches et possibilité de pondération des membres du prédicat de recherche
- FREETEXT : fonction scalaire avec recherches flous (à la « Google » )
- FREETEXTTABLE : fonction table avec recherche recherches flous (à la « Google » ), affichage du score de recherches et possibilité de pondération des membres du prédicat de recherche



## 4 / Recherche sémantique

105

Permet de comparer un texte de référence à un lot de texte

- score basé sur la rareté d'apparition des mots dans les textes du lot par rapport aux mots contenus dans la référence.
- nécessite la mise en place d'une base de statistique des mots (ces bases sont incluses sur le média d'installation de SQL Server)

Utilise les fonctions table :

- SEMANTICKEYPHRASETABLE
- SEMANTICSIMILARITYTABLE

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

La requête :

```
SELECT * FROM sys.fulltext_semantic_language_statistics_database;
```

Liste la liste des bases de références sémantiques installées dans l'instance.

Si elle n'est pas présente, lancez SemanticianguageDatabse.msi dans le média d'installation de SQL Server

Ceci installe les fichiers de la base dans « C:\Program Files\Microsoft Semantic Language Database ». Mais vous pouvez la déplacer ou bon vous semble

Il faut ensuite l'attacher avec :

```
CREATE DATABASE semanticsdb
```

```
ON ( FILENAME = 'C:\Program Files\Microsoft Semantic Language Database\semanticsdb.mdf' )
```

```
LOG ON ( FILENAME = 'C:\Program Files\Microsoft Semantic Language Database\semanticsdb_log.ldf' )
```

```
FOR ATTACH;
```

Le contenu de cette base est inaccessible !

## 4 / Les statistiques

106

Les statistiques permettent à l'optimiseur de choisir un plan de requête adapté en fonction des valeurs de filtre et de la distribution des données.

- Derrière chaque index se cachent des statistiques qui permettent de savoir si l'index vaut le coût d'être utilisé.
- L'accès aux données par un index n'est pas toujours la meilleure solution. Tout dépend de la volumétrie manipulée.
- En sus, en cas de pluralité de critères il faut aussi étudier des solutions en utilisant directement la table
  - ▣ Pour cela il faut des statistiques de colonne.

Les statistiques ne sont pas synchrones comme les index. Elles n'ont pas besoin d'être très précises car elles indiquent une tendance et non l'absolue réalité.

## 4 / Les statistiques

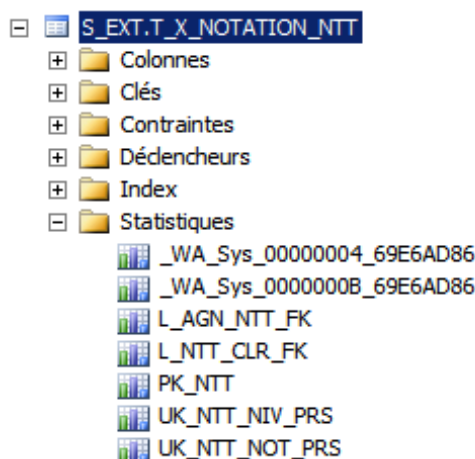
107

Elles sont visibles dans l'explorateur.

Leur contenu peut être vu :

- en demandant les propriétés dans l'IHM
- par la commande DBCC SHOW\_STATISTICS

Ce sont des histogrammes



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les statistiques dont le nom comme par `_WA_Sys_` sont générées automatiquement pour les colonnes de la table si la base est paramétrée avec l'option `AUTO_CREATE_STATISTICS` à ON.

Syntaxe :

```
DBCC SHOW_STATISTICS ( nom_table , stat )
```

```
[ WITH [ NO_INFOMSGS ] < option > ]
```

```
< option > :: =
```

```
STAT_HEADER | DENSITY_VECTOR | HISTOGRAM | STATS_STREAM
```

`STATS_STREAM` serait le contenu binaire de l'histogramme. Ceci n'est pas garanti. Cela peut servir à exporter une statistique. L'importation se faisant par `UPDATE STATISTICS ... WITH STAT_STREAM ...` (non garanti).

Des statistiques complémentaires non visibles sont collectées afin de permettre d'estimer des recherches LIKE avec joker. Celle-ci ne sont pas documentées et font l'objet d'un brevet logiciel.

## 4 / Les statistiques

108

En tête les métadonnées de l'index

Le vecteur de densité indique la sélectivité globale

L'histogramme indique la sélectivité pour une valeur

| Name | Updated    | Rows              | Rows Sampled | Steps | Density | Average key length | String Index | Filter Expression | Unfiltered Rows |         |
|------|------------|-------------------|--------------|-------|---------|--------------------|--------------|-------------------|-----------------|---------|
| 1    | _WA_Sys... | déc 5 2015 1:14PM | 1297080      | 68304 | 194     | 0,8680606          | 32           | YES               | NULL            | 1297080 |

| All density | Average Length | Columns |         |
|-------------|----------------|---------|---------|
| 1           | 4.497434E-06   | 32      | EMP_NOM |

**DENSITY VECTOR**      **HEADER**

| RANGE_HI_KEY | RANGE_ROWS | EQ_ROWS  | DISTINCT_RANGE_ROWS | AVG_RANGE_ROWS |          |
|--------------|------------|----------|---------------------|----------------|----------|
| 1            | ABANA      | 0        | 1                   | 0              | 1        |
| 2            | AGOUX      | 6386,635 | 30,24925            | 1224           | 5,218303 |
| 3            | AMEAU      | 9275,826 | 74,55286            | 1746           | 5,31382  |
| 4            | ANDRARD    | 6500,682 | 53,38486            | 1141           | 5,698755 |
| 5            | ARARD      | 4542,874 | 113,5158            | 872            | 5,207914 |
| 6            | ARNARD     | 4580,89  | 94,5965             | 853            | 5,367815 |
| 7            | ARVIER     | 4124,702 | 53,38486            | 687            | 6,003441 |
| 8            | AUZARD     | 6842,823 | 74,55286            | 1318           | 5,19023  |
| 9            | BARON      | 4447,835 | 170,2737            | 816            | 5,46219  |

**HISTOGRAM**

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Résultat d'un DBCC SHOW\_STATISTICS sur les statistiques d'une colonne littérale contenant un nom de personne.

### Histogramme :

Le nombre d'entrée ne dépasse jamais 200? Description des colonnes :

- EQ\_ROWS indique le nombre de lignes ayant exactement la valeur de référence située dans RANGE\_HI\_KEY
- RANGE\_ROWS indique le nombre de lignes dont les valeurs sont situées entre les bornes de l'histogramme
- DISTINCT\_RANGE\_ROWS indique le nombre de valeurs distinctes trouvées entre les bornes de l'histogramme
- AVG\_RANGE\_ROWS donne la densité entre les bornes de l'histogramme (normalement RANGE\_ROWS / DISTINCT\_RANGE\_ROWS si l'échantillon est de 100%)

## 4 / Les statistiques

109

**Attention** : le calcul des, statistiques peut porter sur un échantillon (tir aléatoire de pages).

- Cela peut porter préjudice à la qualité des statistiques et entraîner des plans de requête peu judicieux.

Il est intéressant de forcer un recalcul en FULLSCAN de temps en temps

- On peut aussi les recalculer par partition ou de manière incrémentale

**La maintenance des statistiques est FONDAMENTALE !**

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les statistiques incrémentales sont recrées par partitions.

## 4 / Les statistiques, maintenance

110

### Connaître les statistiques obsolètes (à minima) :

```
SELECT s.name AS TABLE_SCHEMA, o.name AS TABLE_NAME,
       st.name AS STAT_NAME, p.*
FROM   sys.stats AS st
       CROSS APPLY sys.dm_db_stats_properties(
                   st.object_id,
                   st.stats_id) AS p
JOIN   sys.objects AS o
       ON st.object_id = o.object_id
JOIN   sys.schemas AS s
       ON o.schema_id = s.schema_id
WHERE  modification_counter > rows / 10.0
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Certaines statistiques sont vues comme NULL dans cette requête. On peut alors se demander s'il y a des lignes quand même dans la table !  
(voir les requêtes données en exemple)

## 4 / Les statistiques, maintenance

111

### Recalculer les statistiques :

- Globalement : `sp_update_stats`
- Particulièrement : `UPDATE STATISTICS ...`  
syntaxe simplifié :

```
UPDATE STATISTICS nom_table ( nom_stats )  
  [ WITH [ FULLSCAN  
        | SAMPLE nombre { PERCENT | ROWS } ]  
  [ [ , ] [ ALL | COLUMNS | INDEX ]  
  [ [ , ] NORECOMPUTE ] ]
```

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

La procédure `sp_updatestats` exécute la commande `UPDATE STATISTICS` sur toutes les tables utilisateur et les tables internes de la base de données courante.

Pour le recalcul par partition, voir la syntaxe détaillée de la commande `UPDATE STATISTICS`

## 4 / Les statistiques, maintenance

112

Mise à jour automatique des statistiques :

- ALTER DATABASE :
  - AUTO\_UPDATE\_STATISTICS
  - AUTO\_UPDATE\_STATISTICS\_ASYNC (ed. Enterprise)
  - AUTO\_CREATE\_STATISTICS
- sp\_autostats : affiche ou modifie l'option de mise à jour automatique des statistiques pour un objet particulier

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Le seuil de déclenchement du recalcul des statistiques autrefois de 20% (avec un minimum de 500 lignes) a changé pour une version dynamique à partir de la version 2008 R2 SP1.

- AUTO\_UPDATE\_STATISTICS met à jour automatiquement une statistique obsolète dès que cette statistique est estimée nécessaire, avant de calculer le plan de requête
- AUTO\_UPDATE\_STATISTICS\_ASYNC met à jour les statistiques obsolètes au fil de l'eau en utilisant un thread dédié. ATTENTION : ce thread pouvant empêcher l'arrêt d'une base de données il est recommandé de placer AUTO\_UPDATE\_STATISTICS\_ASYNC à OFF préalablement. Enfin, cette option nécessite que AUTO\_UPDATE\_STATISTICS soit à ON (les deux options doivent être conjointement mise à on).

Syntaxe de la procédure sp\_autostats :

```
EXEC sp_autostats [ @tblname = ] 'nom_table'  
    [ , [ @flagc = ] { 'ON' | 'OFF' }  
    [ , [ @indname = ] 'nom_stat' ]
```

Exemples :

```
EXEC sp_autostats 'ma_table'; --> affiche l'état de toutes les statistiques d'une table  
EXEC sp_autostats 'ma_table', 'OFF', 'ma_stat'; --> interdit la mise à jour automatique de  
la statistique "ma_stat" de la table "ma_table"
```



## 4 / Les plans de requête

113

Indique comment l'optimiseur va traiter la requête :

□ Avant exécution (plan estimé)



□ Après exécution (plan réalisé)



L'optimiseur utilise deux techniques :

- Optimisation sémantique : tient compte des contraintes (PK, UNIQUE, CHECK et FK) pour éliminer des opérations inutiles
- Optimisation statistiques : tient compte de la distribution des données afin de choisir les accès aux données et les algorithmes de traitement

Le plan de requête est un arbre algébrique de traitement.

Le plan réalisé peut différer du plan estimé pour différentes raisons. Lorsque le plan estimé est repris du cache, une contre analyse peut estimer ce plan obsolète et en calculer un nouveau. Par exemple suite à des modifications majeures de données dans les tables.

Pour information, un outil gratuit de l'éditeur Sentry (Plan Explorer) permet de réarranger les icônes et affiche des informations plus pertinentes que celui de Microsoft. Dans la version 2016, l'affichage du plan est dynamique (pour de longues requêtes on voit les opérations s'effectuer dans l'affichage du plan de requête)

## 4 / Les plans de requête

114

Les feuilles de l'arbre sont les accès aux données

La racine le résultat.

En passant la souris sur les opérations on obtient le détail

**Correspondances de hash**  
Utilisez chaque ligne à partir de l'entrée supérieure pour construire une table de hachage, et chaque ligne à partir de l'entrée inférieure pour analyser la table de hachage et générer toutes les lignes correspondantes.

| Opération physique                 | Correspondances de hash |
|------------------------------------|-------------------------|
| Opération logique                  | Agrégation              |
| Mode d'exécution estimé            | Row                     |
| Coût E/S estimé                    | 0                       |
| Coût d'opérateur estimé            | 0,062472 (9 %)          |
| Coût des sous-arborescences estimé | 0,707484                |
| Coût I/O estimé                    | 0,062472                |

Administration SQL Server - CH 04 - Agent SQL V3.0 10/07/2017

Il est aussi possible d'afficher la fenêtre propriété qui donne encore plus de détails sur chaque opération

## 4 / Les plans de requête

115

La fenêtre  
« propriété »  
donne des  
indications  
supplémentaires  
sur chaque étape  
du plan

| Propriétés   |  |
|--|--|
| SELECT   |  |
| Divers   |  |
| CardinalityEstimationModelVersion                          | 140                                      |
| Cause de la fin précoce de l'optimisation de l'instruction | Délai dépassé                            |
| CompileCPU   | 51                                       |
| CompileMemory  | 3088                                     |
| CompileTime  | 52                                       |
| Coût des sous-arborescences estimé                         | 0,395601                                 |
| Coût d'opérateur estimé                                    | 0 (0 %)                                  |
| Instruction  | SELECT * FROM INFORMATION_SCHEMA.COLUMNS |
| MemoryGrantInfo  |  |
| SerialDesiredMemory  | 2352                                     |
| SerialRequiredMemory                                       | 2048                                     |
| Niveau d'optimisation                                      | FULL                                     |
| Nombre de lignes estimé                                    | 494,799                                  |
| NonParallelPlanReason                                      | CouldNotGenerateValidParallelPlan        |
| OptimizerHardwareDependentProperties                       |  |
| EstimatedAvailableDegreeOfParallelism                      | 24                                       |
| EstimatedAvailableMemoryGrant                              | 279424                                   |
| EstimatedPagesCached                                       | 838274                                   |
| MaxCompileMemory   | 81226600                                 |
| OptimizerStatsUsage  |  |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Ici on peut voir que l'optimiseur a du trouver un plan de requête qui n'est pas forcément le meilleur (Cause de la fin précoce de l'optimisation...) mais avec un niveau d'optimisation « FULL ».

## 4 / Les plans de requête

116

Parfois des différences peuvent apparaître dans le plan réalisé.

Il faut regarder les métriques estimées et réalisées

Ici une estimation anormale de cardinalité pour une jointure...

| Merge Join  |                      |
|---|----------------------|
| Match rows from two suitably sorted input tables exploiting their sort order. |                      |
| Physical Operation  | Merge Join           |
| Logical Operation   | Right Anti Semi Join |
| Actual Execution Mode   | Row                  |
| Estimated Execution Mode  | Row                  |
| Actual Number of Rows   | 19707                |
| Actual Number of Batches  | 0                    |
| Estimated Operator Cost   | 0,0474667 (27%)      |
| Estimated I/O Cost  | 0                    |
| Estimated CPU Cost  | 0,0474594            |
| Estimated Subtree Cost  | 0,16881              |
| Estimated Number of Executions  | 1                    |
| Number of Executions  | 1                    |
| Estimated Number of Rows  | 1                    |
| Estimated Row Size  | 56 B                 |
| Actual Rebinds  | 0                    |
| Actual Rewinds  | 0                    |
| Many to Many  | False                |
| Node ID   | 2                    |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Ce cas est classique lorsqu'il y a redondance des jointures avec une jointure interne et une semi anti jointure externe. Exemple :SELECT ...

```
SELECT ...
```

```
FROM T1 JOIN T2 ON T1.K =T2K
```

```
WHERE T1.C NOT IN (SELECT C
```

```
FROM T1 AS T
```

```
WHERE A = @A
```











```
AND T.K = T1.K)
```

NOTEZ que cette anomalie à été corrigée dans les différents Service pack.

## 4 / Les plans de requête

117

### Icones importantes... les lectures :

| Lecture des données      |   |  |
|--------------------------|---|--|
| Table scan               |  | Balayage de table  |
| Table valued function    |  | Construction d'une table à partir d'une fonction         |
| Columnstore index scan   |  | Balayage d'index columnstore                             |
| Clustered index scan     |  | Balayage de table organisée sous forme d'index clustered |
| Non clustered index scan |  | Balayage d'index non clustered                           |
| Parameter table scan     |  | Lecture d'une table construite avec des paramètres       |
| Key lookup               |  | Recherche multilignes par clef                           |
| RID lookup               |  | Recherche multiligne par Row ID                          |
| Clustered index seek     |  | Recherche dans un index clustered                        |
| Non clustered index seek |  | Recherche dans un index non clustered                    |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017











Souvenez vous que CLUSTERED INDEX = TABLE !

Préférez les seeks, puis les lookups aux scans !

## 4 / Les plans de requête

118

### Icones importantes... jointures et autres opérations :

| Jointures             |   |                                       |
|-----------------------|---|---------------------------------------|
| Nested loop           |  | Boucle imbriquée                      |
| Hash match            |  | Rapprochement par hachage             |
| Merge join            |  | Jointure par tri fusion               |
| Concaténation         |  | Concaténation de résultat (Union all) |
| Opérations classiques |   |                                       |
| Sort                  |  | Tri                                   |
| Stream Aggregate      |  | Groupage et agrégat                   |
| Filter                |  | Filtre                                |
| Compute scalar        |  | Calcul scalaire                       |
| Top                   |  | Retourne les <i>n</i> premières ligne |
| UDX                   |  | Opération XQuery/XPath                |

Administration SQL Server - CH 04 - Agent SQL












V3.0 10/07/2017

Les opérations de tri et de groupage sont souvent pénalisantes

## 4 / Les plans de requête

119

### Icones importantes... parallélisme et tempdb :

| Parallélisme                |   |   |
|-----------------------------|---|---|
| (parallelism)               |  | Indique qu'une opération s'effectue en parallèle (en surimpression sur une autre icône) |
| Gather streams              |  | Rassemble plusieurs flux en un seul   |
| Repartition streams         |  | Ventilation de $n$ flux en $m$ flux   |
| Distribute stream           |  | Éclate un flux en plusieurs   |
| Bitmap                      |  | Filtrage bitmap   |
| Opérations utilisant tempdb |   |   |
| Spool                       |  | Enregistre un résultat intermédiaire dans la base tempdb                                |
| Table spool                 |  | Copie de lignes dans une table de tempdb  |
| Eager spool                 |  | Stocke des données transitoires dans un objet de la base tempdb                         |
| Lazy spool                  |  | Stocke des données transitoires dans un objet masqué de la base tempdb                  |
| Non clustered Index spool   |  | Stocke des données de recherche dans un index masqué de la base tempdb                  |
| Row count spool             |  | Compte le nombre de ligne et renvoi une table temporaire exempte de données             |

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Lorsque des opérations utilisent la tempdb cela signifie que certaines opérations n'ont pas pu être effectuées en mémoire compte tenue de la forte volumétrie

## 4 / Les plans de requête

120

Mesure des requêtes...

Il est possible d'obtenir des métriques d'IO et de temps très utiles pour l'affinage des requêtes :

- SET STATISTICS IO { ON | OFF }
- SET STATISTICS TIME { ON | OFF }

Commencez par régler le problème des IO... moins il y en a, mieux c'est.

Pour les très grosses requêtes, affinez avec le temps de calcul... moins, c'est mieux aussi !

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

STATISTICS IO :

- Lecture logique = en mémoire
- Lectures physiques ou anticipées = sur le disque
- Analyse = pénétration dans l'objet (arbre ou liste)

STATISTICS TIME :

- Temps d'analyse et de compilation : temps mis à intégrer, « comprendre » et trouver un plan de requête (peu important)
- Temps d'exécution : temps de réalisation de la requête
- Temps UC : durée globale d'utilisation des processeurs (cumulées si parallélisme)
- Temps écoulé : durée chrono de la requête



## 4 / Les plans de requête

121

Obtenir des scores de performance du serveur :

- Par requête : `sys.dm_exec_query_stats`
- Par procédure : `sys.dm_exec_procedure_stats`
- Par déclencheurs : `sys.dm_exec_trigger_stats`
- Par fonction : `sys.dm_exec_function_stats` (ed. 2016)

Complétez avec :

- `sys.dm_exec_query_plan`
- `sys.dm_exec_sql_text`

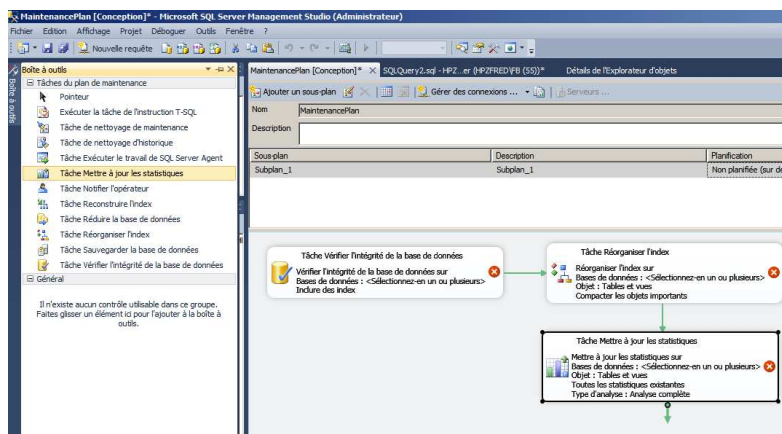
Pour obtenir la commande SQL et le plan d'exécution

Voir requêtes données en exemple.

## 4 / Le plan de maintenance

122

SSMS dispose d'un assistant de création de plan de maintenance basé sur des tâches préprogrammées de SSIS



Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Les plans de maintenance « graphiques » se situent dans l'entrée « Gestion / Plan de maintenance » de l'arborescence de l'explorateur. Les différents « paquetage » SSIS conduisent à différents travaux planifiés dans l'Agent SQL

Ce système peut être utilisé pour mettre en place la maintenance des petites bases. Travaux très grossiers.

À éviter pour des configurations moyennes à grosses (nombreuses bases ou volumétrie importantes)

Ne surtout pas utiliser pour les VLDB ou les bases à fonctionnement continu ! Soyez malin....

### ATTENTION :

évités absolument la tâche « Réduire la base de données » ! Cela effectue un « SHRINK » Commencez toujours par la maintenance des index puis celle des statistiques.

## 4 / Le plan de maintenance

123

Étapes essentielles d'un plan de maintenance :

1. Vérifier la base (DBCC CHECK...)
2. Défragmenter les index (ALTER INDEX ...)
3. Recalculer les statistiques (UPDATE STATISTICS ...)

**Dans cet ordre précis**

À faire si possible quotidiennement !

Pour les VLDB, options ONLINE et partiellement par rotation pour 2 et 3

Prévoyez vos sauvegardes avant et/ou après. Par exemple sauvegarde complète avant et différentielle après

## 4 / Le plan de maintenance

124

Étapes utiles d'un plan de maintenance :

1. Vérifier la présence de transaction longue et alerter
2. Vérifier le taux d'occupation des fichiers et prévenir en augmentant les enveloppes
3. Vérifier le remplissage des disques du serveur (par exemple 80%) et alerter, faire du ménage

Administration SQL Server - CH 04 - Agent SQL

V3.0 10/07/2017

Pour les transactions longues, utilisez DBCC OPENTRAN ou des DMV (sys.dm\_tran\_database\_transactions, sys.dm\_tran\_session\_transactions, sys.dm\_tran\_active\_transactions...)

Pour le taux d'occupation des fichiers, utilisez les DMV sys.database\_files et sys.filegroups associé à la fonction FILEPROPERTY pour obtenir le taux d'occupation

Pour le taux d'occupation des disques, utilisez la fonction table sys.dm\_os\_volume\_stats associée à la vue sys.master\_files

Voir les requêtes d'exemple...



Version 3.001  
Date 10/07/2017



5 Sauvegardes et restauration

Administration SQL Server - CH 05 - Sauvegardes et restauration V3.0 10/07/2017

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

# 5 – Sauvegardes et restauration

3



- ❑ Les différents type de sauvegardes
- ❑ Mode de récupération
- ❑ Stratégie de sauvegarde
- ❑ Paramètres de la sauvegarde
- ❑ Planification des sauvegardes
- ❑ Restauration des bases
- ❑ Restauration avec les journaux
- ❑ Paramètres de la restauration
- ❑ Restauration de la base master

## 5 / Sauvegarde

4

**Définition** : consiste à dupliquer les données contenues dans un système informatique **de manière à pouvoir le restaurer...**

*sauvegarder sans prévoir dès le départ  
la restauration est une hérésie !*

Deux conceptions de la sauvegarde

- Sauvegarde pour pallier à une erreur humaine
- Sauvegarde pour archivage (souvent légal)

Ne pas confondre sauvegarde et haute disponibilité.

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

À partir de la version 2016 de SQL Server, avec l'introduction des « temporal tables » il est possible de conserver automatiquement toutes les versions de toutes les lignes de la table (UPDATE et DELETE) avec un référentiel de temps. Des opérateurs particuliers permettent de revoir les données telles qu'elles se présentaient à n'importe quel point dans le temps.

La mise en place massive d'une telle technique conduira probablement à l'abandon des systèmes de sauvegardes tel que nous les connaissons actuellement.



## 5 / Sauvegarde

5

Quelques questions...

- Comment sauvegarder ?
- Qui effectue les sauvegardes ?
- Où sauvegarder ?
- Faut-il être paranoïaque ?
- Faut-il tester ses sauvegardes ?
- Faut-il chiffrer ses sauvegardes ?

BACKUP  
le DBA  
disque, bande  
non!  
oui!  
cela dépend

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Dans un SGBDR l'exigence est d'effectuer les sauvegardes sans interrompre le service des données. Les SGBDR moderne travaillant 24h sur 24h en général il ne faut pas arrêter le serveur ce qui aurait pour conséquence d'interdire les accès à la base et en outre de faire perdre tout ce qui a été mis en cache. C'est pourquoi la copie des fichiers d'une base qui nécessiterait l'arrêt du serveur, ne peut être considéré comme viable (sauvegarde à froid).

**ATTENTION** : la sauvegarde ne doit pas concerner que les bases de production ! Les bases systèmes master et msdb doivent être sauvegardées le plus régulièrement possible.

## 5 / Modes de sauvegarde

6

- Complète** : enregistre l'intégralité de la base de données
- Différentielle** : enregistre toutes les pages modifiées depuis la dernière complète
- Journal des transactions** : enregistre la portion active du journal des transactions

### Déclinaisons :

- Complète de fichiers ou groupes de fichiers
- Différentiel de fichiers ou groupes de fichiers

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

**ATTENTION** : bien qu'il soit possible d'effectuer des sauvegardes partielles, il n'est pas possible de restaurer partiellement (quelques éléments d'une base par exemple). Cela n'a aucun sens d'un point de vue logique car risquerait de compromettre l'intégrité de la base...

## 5 / Sauvegarde complète

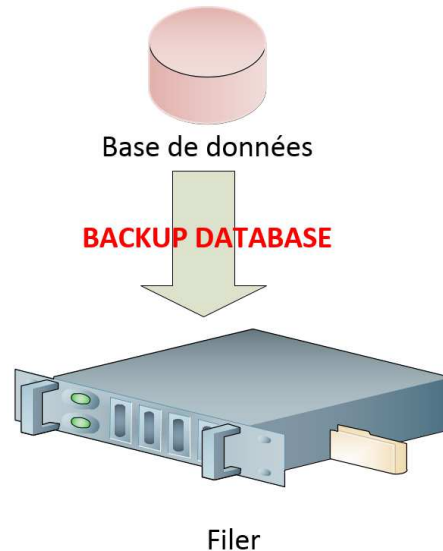
7

Enregistre l'intégralité de la base de données, transactions comprises.

À la restauration, la base contient les données à l'heure de fin de la sauvegarde complète.

Laisse le journal tel quel.

Initialise les sauvegardes différentielles et du journal de transaction.



Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Le mécanisme est le suivant :

- Met une information de démarrage de la sauvegarde dans le journal
- Créé un fichier de réception d'une taille estimée devoir contenir l'intégralité de la sauvegarde
- Enregistre les premières pages techniques du fichier primaire (qui comporte la description du stockage de la base)
- Enregistre les autres pages non vides dans un ordre déterminé par un algorithme en logique flou (augmente la taille du fichier de sauvegarde si l'estimation de départ était incorrecte)
- Enregistre les transactions effectuées depuis le début de la sauvegarde
- Met une information de fin de sauvegarde dans le journal
- Diminue au besoin la taille du fichier de sauvegarde si l'enveloppe était trop grande.

## 5 / Sauvegarde différentielle

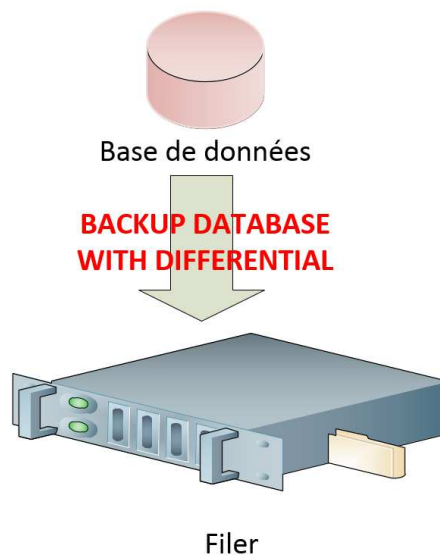
8

Enregistre les pages modifiées depuis la dernière sauvegarde complète et les transactions qui ont lieu pendant la sauvegarde.

À la restauration, la base contient les données à l'heure de fin de la sauvegarde différentielle.

Laisse le journal tel quel.

Initialise les sauvegardes du journal de transaction.



Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Une différentielle part toujours de la dernière sauvegarde complète. Des sauvegardes différentielle successives ne sont pas incrémentales, elles cumulent toutes les modifications. Chacune enregistre l'intégralité des pages mises à jour depuis la dernière sauvegarde complète. Par conséquent, en cas de sauvegarde différentielle successive, la dernière sauvegarde différentielle contient toutes les données à restaurer.

## 5 / Sauvegarde transactionnelle

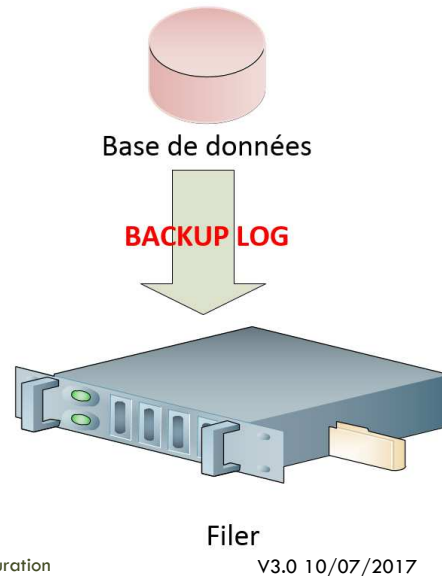
9

Enregistre les transactions qui ont lieu depuis la dernière sauvegarde.

À la restauration, la base contient les données à l'heure de fin de la sauvegarde transactionnelle.

Purge le journal de transaction.

Initialise de nouvelles sauvegardes du journal de transaction.



Administration SQL Server - CH 05 - Sauvegardes et restauration

La sauvegarde transactionnelle à un double rôle :

Permettre de rejouer les dernières transaction de la base (génération des « redo logs »)

Purger le journal de transactions

Les sauvegardes transactionnelles sont incrémentales. Chacune ne contient que les transactions effectuées entre la fin de la dernière sauvegarde quelle qu'elle soit (complète, différentielle ou transactionnelle).

Des sauvegardes transactionnelles successives doivent être restaurées successivement et dans l'ordre chronologique pour assurer une restauration intégrale de la base.

## 5 / Destination de la sauvegarde

10

- Sur un « device » : objet logique du serveur pointant vers une destination physique
- Sur une unité à bande (obsolète)
- Vers un fichier :
  - Local
  - Distant (chemin UNC)
- Vers Azure

### **ATTENTION :**

1. le jeu de sauvegarde peut résider sur plusieurs unités
2. Le fichier de destination peut « empiler » les sauvegardes

Pour créer un « device », utilisez la procédure `sp_addumpdevice`

## 5 / Mode de récupération

11

### « Recovery Model »

- Détermine ce que vous pouvez restaurer à partir de vos sauvegardes :
  - ▣ Complet (Full)
  - ▣ Journalisé en bloc (Bulk Logged)
  - ▣ Simple

- Spécifie ce qui est journalisé

ALTER DATABASE ... SET RECOVERY ...

Tant qu'une base n'a pas encore été initialisée par une sauvegarde complète, le journal de transaction se comporte comme si la base était en mode « simple », quelque soit le mode de récupération !

## 5 / Mode de récupération

12

### Complet (Full) :

- mode par défaut (sauf changement dans base model);
- Permet une récupération totale ou à un point dans le temps (PITR) de la base de donnée.

### Avantages :

- Aucune perte de donnée;
- Restauration sur un point temporel possible.

### Inconvénients :

- Taille du journal de transactions importante.
- La taille du journal augmente

Mode utilisé la plupart du temps pour les bases OLTP

PITR : Point In Time Recovery



## 5 / Mode de récupération

13

Bulk logged (journalisé en bloc) :

- mode souvent transitoire;
- Permet une récupération totale de la base de donnée.

Avantages :

- Aucune perte de donnée;
- Taille du journal de transactions minimisée.

Inconvénients :

- Restauration sur un point temporel impossible.
- La taille du journal augmente

## 5 / Mode de récupération

14

Bulk logged (journalisé en bloc), opération à journalisation minimale :

- Chargement de fichier via :
  - bcp.exe
  - BULK INSERT
- CREATE/ALTER INDEX ...
- SELECT ... INTO ...
- INSERT ... SELECT ...
- WRITETEXT, UPDATETEXT ... (obsolètes)

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Le mode de journalisation pouvant être changé dynamiquement, il est souvent intéressant de passer en mode BULK LOGGED lors des phases de maintenance des index.

Pour les fichiers, le journal conserve le chemin du fichier. En cas de restauration du journal de transaction, si le fichier a disparu, une erreur est générée et la base ne peut pas être restaurée.

## 5 / Mode de récupération

15

Simple :

- Journalisation minimale (comme dans le mode BULK LOGGED)
- Le journal est purgé après chaque CHECKPOINT

Avantages

- La taille du journal est minimale et maîtrisée
- Amélioration des performances de requêtes

Inconvénients

- Impossible de restaurer à partir du journal de transactions

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

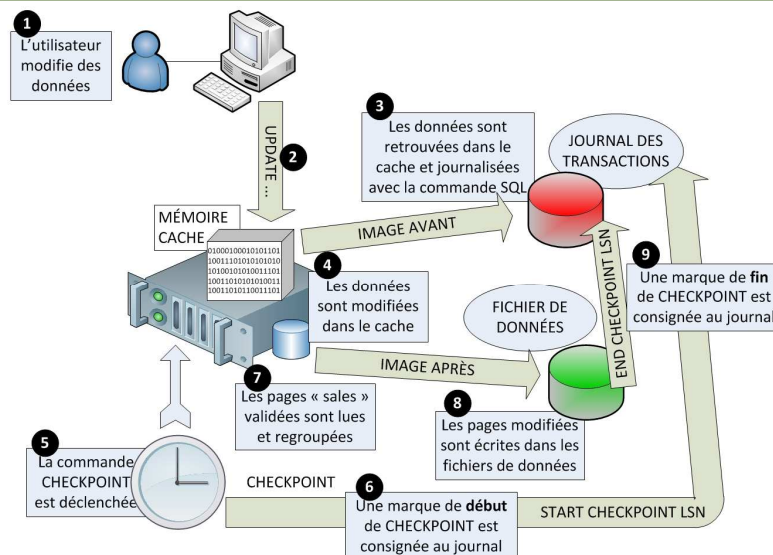
La commande CHECKPOINT intervient régulièrement (à défaut, environ toutes les minutes) pour aller écrire de manière asynchrone les pages « sales » (pages mise à jour en cache, mais pas encore écrites sur disque) en mémoire dans les fichiers de données.

La commande CHECKPOINT est inscrite au journal.

**ATTENTION** : les sauvegardes du journal de transactions sont impossible dans ce mode vu qu'il est purge en permanence !

## 5 / Checkpoint

16



Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Lors de l'envoi de la commande CHECKPOINT le système parcourt la mémoire à la recherche des pages « sales », donc contenant les nouvelles données à écrire, les regroupe par contiguïtés d'emplacements physiques sur les plateaux du disque et les écrits dans les fichiers de données, ceci afin d'optimiser la durée du processus :

- si une page mémoire a été modifiée plusieurs fois dans l'intervalle entre deux checkpoints, une seule écriture physique sera effectuée ;
- le regroupement par contiguïtés des emplacements physiques sur les plateaux du disque permet de minimiser le trajet de la tête de lecture.

Une fois que les pages sales ont été définitivement répercutées dans les fichiers de données, les transactions inscrites au journal et concernant ces pages sont marquées comme terminées et la place qu'elles occupent dans le journal est libérable :

Effectivement si le mode de journalisation est SIMPLE

Lors de la sauvegarde transactionnelle pour les autres modes (FULL et BULK LOGGED)

Le réglage du délai entre deux CHECKPOINT peut être fait base par base (ALTER DATABASE ... SET TARGET\_RECOVERY\_TIME en secondes), ou globalement au niveau du serveur (EXEC sp\_configure 'recovery interval' en minutes). Le réglage par défaut est optimal pour la majorité des configurations.

**NOTA** : On peut « visionner » le travail du checkpoint en combinant les indicateurs de trace 3502 et 3605, voire 3504. Ceci ajoute des informations au journal d'événement de SQL Server pour les CHECKPOINT.

## 5 / Mode de récupération...

17

... et journal de transaction

En mode « SIMPLE » :

- Le journal est purgé après chaque CHECKPOINT
- La taille du journal est maîtrisée, sauf en cas de transaction longue

En mode « FULL » et « BULK LOGGED »

- Le journal va croître indéfiniment, jusqu'à ce qu'une sauvegarde transactionnelle ait lieu qui le purge

**ATTENTION** : purge <> réduction !

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Une purge libère de la place dans le journal, mais le fichier du journal ne diminue pas. La seule commande qui permet de diminuer un fichier de bases de données dans SQL Server est la commande DBCC SHRINK...

Pour savoir si une transaction ancienne perdure, utilisez DBCC OPENTRAN ou une requête portant sur les vues systèmes sys.dm\_tran\_active\_transactions et sys.dm\_tran\_session\_transactions

## 5 / Mode de récupération

18

Vérifier le mode de récupération de vos bases :

```
SELECT database_id, name, recovery_model,
       recovery_model_desc
FROM   sys.databases
```

Pour une base :

```
SELECT
DATABASEPROPERTYEX('maBase', 'RECOVERY');
```

Pour changer le mode :

```
ALTER DATABASE nom_base
SET RECOVERY { FULL | BULK_LOGGED | SIMPLE }
```

Les bases master, tempdb et msdb sont en mode de récupération simple.  
Inchangeable pour master en tempdb.

## 5 / Stratégie de sauvegarde

19

Élaborer une stratégie de sauvegarde passe par de multiples questions :

- Quel est votre plan de récupération sur erreur ?
- Quelle est l'importance stratégique de vos données ?
- Quelle est la fréquence de mise à jour des données ?
- Quelle quantité/durée de mises à jour pouvez-vous sacrifier ?
- A quel point du passé, avec quelle précision souhaitez-vous pouvoir restaurer ?

## 5 / Stratégie de sauvegarde

20

Deux mesures de temps sont importantes :

Combien d'heures de production acceptez vous de perdre ?

- FREQUENCE des sauvegardes

Combien de temps va durer la restauration ?

- STRATÉGIE et type de sauvegardes

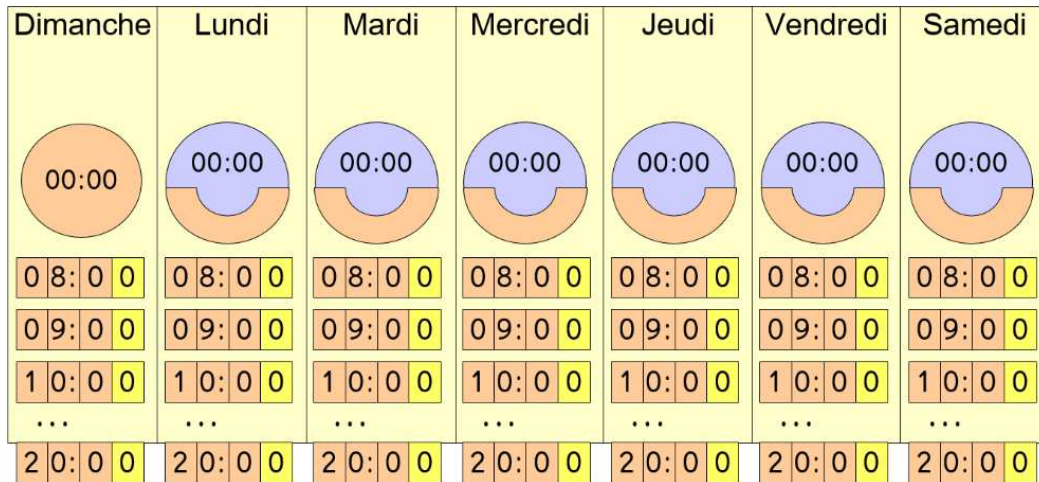
Vos commandes SQL vont en dépendre...



# 5 / Stratégie de sauvegarde

21

Exemple de planification des sauvegardes :



Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Une FULL le dimanche à 0h

Une différentielle du lundi au samedi à 0h

Les journaux de transaction chaque jour, toutes les heures entre 8h et 20h

Deux modes de récupération permettent de mettre en place cette stratégie :

- Mode FULL
- Mode BULK LOGGED

Mais seul le mode FULL permet une restauration à un point dans le temps.

## 5 / Commande de sauvegarde

22

Complète, syntaxe :

```
BACKUP DATABASE { nom_base | @variable_nom_base }  
  TO <unité_sauvegarde> [, ...n]  
  [WITH <options> ]
```

Différentielle, syntaxe :

```
BACKUP DATABASE { nom_base | @variable_nom_base }  
  TO <unité_sauvegarde> [, ...n]  
  WITH DIFFERENTIAL [ , <autres_options> ]
```

Transactionnelle, syntaxe

```
BACKUP LOG { nom_base | @variable_nom_base }  
  TO <unité_sauvegarde> [, ...n]  
  [WITH <options> ]
```

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Les options diffèrent légèrement entre les sauvegardes DATABASE et les sauvegardes LOG

La sauvegarde ne prend en compte que les pages écrites. Les pages vides sont ignorées, mais bien restituée au moment de la restauration.

Pour envoyer une sauvegarde vers un disque, précisez :

- TO DISK = 'MonCheminVersMonRépertoire\monFichierDeSauvegarde.extension'

Pour envoyer une sauvegarde vers une bande, précisez : (obsolète)

- TO TAPE = 'MonCheminVersMonRépertoire\monFichierDeSauvegarde.extension'

Pour envoyer une sauvegarde dans le cloud Azure :

TO URL = 'mon\_url\_azure'

CONSEIL : Prévoyez le chiffrement de la sauvegarde vers Azure !

Pour envoyer une sauvegarde vers un *device*, précisez :

- TO *nom\_device*

N'oubliez pas que, pour un répertoire distant, il faut un chemin de type UNC (\\MonServeur\MonPartage\...)

SQL Server est autorisé à faire des sauvegardes locales. Pour les sauvegardes distante il faut que le compte de service de SQL Server soit munis des droits systèmes pour aller écrire sur la ressource distante.

Par convention, les extensions de fichiers pour les sauvegardes de SQL Server sont BAK pour les complètes et différentielles et TRN pour les journaux de transactions. Mais elle ne sont pas reconnues par le système Windows.

Options spécifiques aux bandes :

{ REWIND | NOREWIND }, { UNLOAD | NOUNLOAD }

## 5 / Commande de sauvegarde

23

Quelques options importantes :

FORMAT / NOFORMAT : formate l'unité de destination

INIT / NOINIT : vide l'unité de destination

CHECKSUM / NO\_CHECKSUM : vérifie la somme de contrôle des pages au moment de les écrire

COMPRESSION : comprime les pages avant de les écrire

STATS : indique le pourcentage de traitement pendant l'opération de sauvegarde

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Souvenez vous que pour SQL Server est fichier doit être considéré comme un disque...

Par défaut NOFORMAT, NOINIT :

- Si le fichier n'existe pas, il est créé
- S'il existe, la sauvegarde est empilée dans le fichier
- Si INIT est précisé, toutes les sauvegardes présentes sont supprimées
- Si FORMAT est précisé, le fichier de destination de la sauvegarde est recréé et toutes les sauvegardes qui y résidaient sont perdues

Par défaut NO\_CHECKSUM

## 5 / Commande de sauvegarde

24

Comme dans un même fichier différentes sauvegardes peuvent coexister, se pose le problème de la conservation...

EXPIREDATE = { 'date' | @variable\_date }

ou

RETAIN\_DAYS = { jours | @variable\_jour }

L'option SKIP / NOSKIP : désactive le contrôle de rétention. Les sauvegardes sont préservées même si elles sont obsolètes (délai EXPIRE.../RETAIN... dépassé).

Souvenez-vous que pour SQL Server un fichier doit être considéré comme un disque...

Par défaut :

- Pas d'EXPIREDATE ou RETAIN\_DAYS, ce qui signifie que toutes les sauvegardes à l'intérieur du fichier sont conservées
- NOSKIP

## 5 / Commande de sauvegarde

25

Documenter sa sauvegarde :

- DESCRIPTION : décrit le jeu de sauvegarde (255 car.)
- NAME : nom du jeu de sauvegarde (128 car.)
- MEDIADescription : décrit le média de destination (255 car.) support de la sauvegarde.
- MEDIANAME : nom du média de destination (128 car;°
- Ces informations de métadonnées permettent de repérer facilement un jeu de sauvegarde dans un support.

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Toutes les sauvegardes effectuées dans l'instance sont tracées, et les données disponibles dans la base msdb :

- dbo.backupset : jeux de sauvegardes
- dbo.backupmediaset : media (support) de sauvegardes
- dbo.backupmediafamily : repartition des jeux de sauvegardes dans les médias
- dbo.backupfile : fichiers constitutif des bases sauvegardés par les jeux de sauvegardes
- dbo.backupfilegroup : groupes de fichiers constitutif des bases sauvegardés par les jeux de sauvegardes

## 5 / Commande de sauvegarde

26

### Gérer les erreurs :

STOP\_ON\_ERROR / CONTINUE\_AFTER\_ERROR :

Détermine si une opération de sauvegarde s'arrête ou continue après avoir rencontré une erreur de CHECKSUM

### Chiffrement :

```
ENCRYPTION (ALGORITHM =  
{ AES_128 | AES_192 | AES_256 | TRIPLE_DES_3KEY } ,  
encryptor_options )  
<encryptor_options> ::=  
    SERVER CERTIFICATE = Encryptor_Name  
    | SERVER ASYMMETRIC KEY = Encryptor_Name
```

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Par défaut : STOP\_ON\_ERROR.

Le chiffrement des sauvegardes est déconseillé pour des raisons de performances. Il est néanmoins indispensable si vous envoyez vos sauvegardes vers le cloud Azure !

Les options PASSWORD et MEDIAPASSWORD ont été supprimées depuis la version 2012 d'autant plus que le niveau de protection offert était plutôt inefficace !

## 5 / Commande de sauvegarde

27

### Performances :

**BLOCKSIZE** : Indique, en octets, la taille physique du bloc d'écriture.

**BUFFERCOUNT** : spécifie le nombre total de tampons d'E/S à utiliser pour l'opération de sauvegarde.

**MAXTRANSFERSIZE** : spécifie la taille de la plus grande unité de transfert à utiliser entre SQL Server et le support de sauvegarde (en octets);

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

**ATTENTION** : uniquement pour des cas particuliers !

Par défaut : BLOCKSIZE est choisit automatiquement. Les tailles possibles sont 512, 1024, 2048, 4096, 8192, 16384, 32768 et 65536 (64 Ko) octets. Par défaut 65536 pour les périphériques à bandes, 512 si l'analyse automatique n'est pas disponible.

**BUFFERCOUNT** Vous pouvez spécifier n'importe quel entier positif mais un nombre élevé de tampons peut provoquer des erreurs liées à une insuffisance de mémoire.

**MAXTRANSFERSIZE** valeurs multiples de 65536 octets (64 Ko), dans la limite de 4194304 octets (4 Mo).

L'espace total utilisé en mémoire est déterminé par :  $\text{buffercount} * \text{maxtransfersize}$ .

SQL Server dimensionne automatiquement ces paramètres au plus juste. Vous pouvez les tracer à l'aide du flag 3213.

Voir : <https://blogs.msdn.microsoft.com/psssql/2008/02/06/how-it-works-how-does-sql-server-backup-and-restore-select-transfer-sizes/>

## 5 / Commande de sauvegarde

28

### Urgences :

**COPY\_ONLY** : effectue une sauvegarde sans inscription dans la plan de sauvegarde;

Avec un BACKUP LOG :

- **NORECOVERY** : Effectue une sauvegarde de la fin du journal et laisse la base de données en état de restauration (RESTORING)  
ou
- **STANDBY** : Effectue une sauvegarde de la fin du journal et laisse la base de données en lecture seule et en état STANDBY
- **NO\_TRUNCATE** : évite la purge du journal.

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Toutes les sauvegardes effectuées dans SQL Server se séquences les unes aux autres afin de permettre de restaurer en enchainant les différents fichiers lors de la restauration. Ces enchainements sont basés sur le séquençement des LSN (Log Segment Number) du journal de transaction. Chaque sauvegarde initialise donc le journal de transaction. **COPY\_ONLY** évite ce marquage.

**NORECOVERY** : utile lors du basculement vers une base de données de secours ou de l'exécution d'une sauvegarde de la fin du journal avant une opération RESTORE.

**STANDBY** = *nom\_fichier\_standby* : La clause STANDBY écrit les données en attente (annulation avec option de restauration ultérieure). Le mode d'attente nécessite un fichier d'annulation, spécifié par *nom\_fichier\_standby*, dont l'emplacement figure dans le journal de la base de données. Si le fichier spécifié existe déjà, le Moteur de base de données l'écrase ; sinon, le Moteur de base de données le crée. Le fichier d'annulation devient partie intégrante de la base de données.

Ce fichier contient les modifications annulées, qui doivent être restaurées si des opérations RESTORE LOG sont effectuées ultérieurement. Vous devez disposer d'un espace disque suffisant pour que le fichier d'annulation puisse contenir toutes les pages distinctes de la base de données qui ont été modifiées par suite du rejet des transactions non validées.

**NO\_TRUNCATE** : tente la sauvegarde, quel que soit l'état de la base de données. Les métadonnées d'une telle sauvegarde peuvent être incomplètes. Cette option permet de sauvegarder le journal lorsque la base de données est endommagée. Équivalent à **COPY\_ONLY** et **CONTINUE\_AFTER\_ERROR**. Cette commande peut être lancée même si la base n'est pas en disponible.



## 5 / Commande de sauvegarde

29

### Sauvegardes de fichiers ou groupes de fichiers

Il est possible de sauvegarder certains fichiers ou groupe de fichier.

La syntaxe est alors la suivante :

```
BACKUP DATABASE { nom_base | @variable_nom_base }
  <fichiers_ou_groupes> [ , ... ]
  TO <destinations>
  [ WITH <options> ]
< fichiers_ou_groupes >::=
  { FILE = { nom_logique_fichier
             | @var_ nom_logique_fichier }
    | FILEGROUP = { nom_logique_groupe
                   | @var_ nom_logique_groupe } }
```

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Exemple :

```
BACKUP DATABASE DB_COMPTA
  FILEGROUP = 'COMPTA_ARCHIVE_2014',
  FILEGROUP = 'COMPTA_ARCHIVE_2015'
TO DISK = '\\FILER_SQL_DATA\COMPTA\ARCHIVES\COMPTA_14_15.BAK'
```

Lors de la restauration il est possible de commencer à disposer des données avant que la sauvegarde ne soit terminée, dès qu'un groupe de fichier a été restauré, et à condition d'avoir commencé par le fichier ou groupe de fichier contenant le fichier primaire. Limité à la version Enterprise.

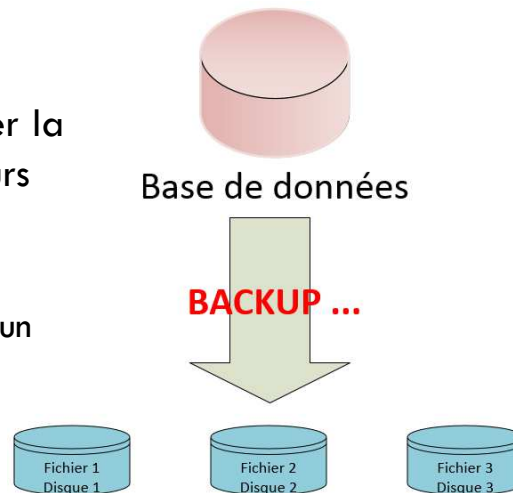
À la place de FILE ou FILEGROUP on peut préciser READ\_WRITE\_FILEGROUPS. Dans ce cas, tous les groupes de fichiers ouverts en écriture seront sauvegardés, à l'exception de ceux en lecture seule.

## 5 / Commande de sauvegarde

30

### Jeu de sauvegarde multifichiers

- Il est possible d'envoyer la sauvegarde sur plusieurs fichiers en parallèle
  - ▣ Permet d'accélérer les écritures (équivalent à un RAID 0)



Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Syntaxe :

```
BACKUP DATABASE { nom_base | @variable_nom_base }  
TO <destination1> [ , <destination2> [ , ... ] ]  
[ WITH <options> ]
```

Exemple :

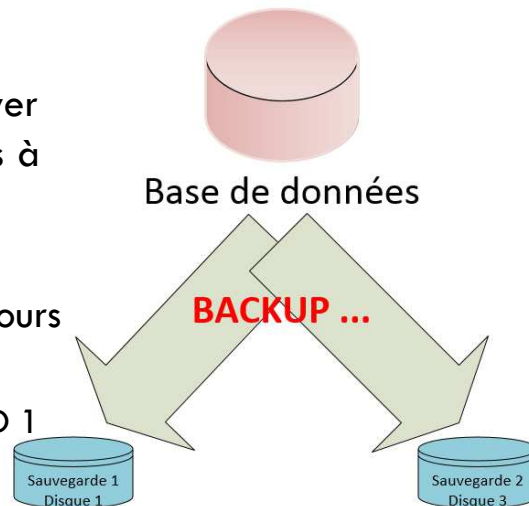
```
BACKUP DATABASE DB_COMPTA  
TO DISK = '\\FILER1_SQL_DATA\COMPTA\ARCHIVES\COMPTA_F1_14_15.BAK',  
DISK = '\\FILER2_SQL_DATA\COMPTA\ARCHIVES\COMPTA_F2_14_15.BAK'
```

## 5 / Commande de sauvegarde

31

### Jeu de sauvegarde multifamilles

- Il est possible d'envoyer plusieurs sauvegardes à plusieurs destination simultanément
- Permet d'avoir un secours de la sauvegarde
  - ▣ Correspond à un RAID 1



Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Syntaxe :

```
BACKUP DATABASE { nom_base | @variable_nom_base }  
  TO <destination1>  
  MIRROR TO <destination2>  
  [ MIRROR TO <destination3> ]  
  [ MIRROR TO <destination4> ]  
  [ WITH <options> ]
```

3 destinations en sus de la principale au maximum.

Exemple :

```
BACKUP DATABASE Mabase  
  TO DISK = 'S:\SAUVE\SQL\COMPTA.BAK'  
  MIRROR TO DISK = '\\FILER1\SQL\COMPTA.BAK'
```

# 5 / Commande de sauvegarde

32

## Sauvegardes compressées (WITH COMPRESSION)

```
BACKUP DATABASE [DB_PORTFOLIO_MANAGER]
TO DISK = 'G:\SAVE\SQL\DB_PORTFOLIO_MANAGER.BAK'
GO

```

Messages

1976 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_PRIM' dans le fichier 1.  
411792 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_DATA\_01' dans le fichier 1.  
336608 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_INDX\_01' dans le fichier 1.  
333128 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_INDX\_02' dans le fichier 1.  
8 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_GDOC\_01' dans le fichier 1.  
549416 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_STRM\_01' dans le fichier 1.  
7 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_LOG\_01' dans le fichier 1.  
BACKUP DATABASE a traité avec succès 1632926 pages en 459.718 secondes (27.758 Mo/s).

0% -

Exécution de requête réussie.

```
Query4.sql - HP2_R (HP2FRED\FB (63)) * X
BACKUP DATABASE [DB_PORTFOLIO_MANAGER]
TO DISK = 'G:\SAVE\SQL\DB_PORTFOLIO_MANAGER_C.BAK'
WITH COMPRESSION;
GO

```

Messages

1976 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_PRIM' dans le fichier 1.  
411792 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_DATA\_01' dans le fichier 1.  
336608 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_INDX\_01' dans le fichier 1.  
333128 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_INDX\_02' dans le fichier 1.  
8 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_GDOC\_01' dans le fichier 1.  
549416 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_STRM\_01' dans le fichier 1.  
2 pages traitées pour la base de données 'DB\_PORTFOLIO\_MANAGER', fichier 'F\_PM\_LOG\_01' dans le fichier 1.  
BACKUP DATABASE a traité avec succès 1632921 pages en 162.128 secondes (78.685 Mo/s).

## 5 / Commande de sauvegarde

33

### Sauvegardes compressées (WITH COMPRESSION)

- Gains importants sur le temps de sauvegarde
- Gains importants sur le temps de restauration
- Gains importants sur le volume
  - Sauf si données chiffrées ou TDE



The screenshot shows a file explorer window with a table overlay. The table has four columns: 'Nom', 'Modifié le', 'Type', and 'Taille'. It lists two backup files: 'DB\_PORTFOLIO\_MANAGER.BAK' (13 068 447 Ko) and 'DB\_PORTFOLIO\_MANAGER\_C.BAK' (2 829 055 Ko).

| Nom                        | Modifié le       | Type        | Taille        |
|----------------------------|------------------|-------------|---------------|
| DB_PORTFOLIO_MANAGER.BAK   | 07/03/2016 19:03 | Fichier BAK | 13 068 447 Ko |
| DB_PORTFOLIO_MANAGER_C.BAK | 07/03/2016 19:16 | Fichier BAK | 2 829 055 Ko  |

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

TDE (Transparent Database Encryption) propose de chiffrer les fichiers de la base (données et journal).

## 5 / Lire une sauvegarde

34

### **Il est possible de lire des métadonnées des fichiers de sauvegardes :**

**RESTORE LABELONLY** : donne des informations sur le média de sauvegarde (date de création, logiciel l'ayant créé..)

**RESTORE HEADERONLY** : liste les fichiers de sauvegarde contenu dans le media de sauvegarde

**RESTORE FILELISTONLY** : liste les fichiers restaurable pour l'un des sauvegardes contenues dans le média (si plusieurs fichiers présent utilisez l'option FILE = n pour atteindre la sauvegarde de position n dans le média)

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Syntaxes :

```
RESTORE LABELONLY
FROM <source_sauvegarde>
[ WITH <options> ]
```

```
RESTORE HEADERONLY
FROM <source_sauvegarde>
[ WITH <options> ]
```

```
RESTORE FILELISTONLY
FROM <source_sauvegarde>
[ WITH <options> ]
```

<options>

```
{ FILE = { backup_set_file_number | @backup_set_file_number }
| PASSWORD = { password | @password_variable }
| MEDIANAME = { media_name | @media_name_variable }
| MEDIAPASSWORD = { mediapassword | @mediapassword_variable }
| { CHECKSUM | NO_CHECKSUM }
| { STOP_ON_ERROR | CONTINUE_AFTER_ERROR }
}
```

NOTA : les option FILE et PASSWORD ne sont pas disponible pour RESTORE LABELONLY

## 5 / Vérifier une sauvegarde

35

### RESTORE VERIFYONLY : contrôle la sauvegarde

Vérifie que la sauvegarde est :

- Lisible (page, somme de contrôle)
- Complète
- Optionnellement restaurable sur la destination choisie
- Permet d'enregistrer les métadonnées de la sauvegardes dans les tables de msdb (par exemple sur un serveur tiers) – option LOADHISTORY

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Syntaxes :

```
RESTORE VERIFYONLY
FROM <source_sauvegarde> [ ,...n ]
[ WITH <options> ]
```

<options>

```
{ LOADHISTORY
| MOVE 'logical_file_name_in_backup' TO 'operating_system_file_name' [ ,...n ]
| FILE = { backup_set_file_number | @backup_set_file_number }
| PASSWORD = { password | @password_variable }
| MEDIANAME = { media_name | @media_name_variable }
| MEDIAPASSWORD = { mediapassword | @mediapassword_variable }
| { CHECKSUM | NO_CHECKSUM }
| { STOP_ON_ERROR | CONTINUE_AFTER_ERROR }
| STATS [= percentage ] }
```

L'option LOADHISTORY permet de charger les métadonnées de cette sauvegarde dans les tables d'historique de sauvegarde (base msdd) dans l'instance sur laquelle on lance cette commande. L'option MOVE permet de savoir si la destination des restaurations dispose d'une place suffisante.

Exemple vérification de la dernière sauvegarde d'une base nommée DB\_COMPTA :

```
DECLARE @LAST_BACKUP_ID INT;
SELECT @LAST_BACKUP_ID = position
FROM msdb.dbo.backupset AS s
WHERE database_name = N'DB_COMPTA'
AND backup_set_id = (SELECT MAX(backup_set_id)
FROM msdb.dbo.backupset
WHERE database_name = s.database_name);
RESTORE VERIFYONLY
FROM DISK = N'G:\SAVE\SQL\DB_COMPTA.BAK'
WITH FILE = @LAST_BACKUP_ID;
```

## 5 / Restauration

36

### Quizz !

- En mode de récupération simple, pouvez-vous utiliser une sauvegarde de journal de transactions ?
- Avec une sauvegarde complète, pouvez-vous restaurer à l'état de la base à un instant précis dans le passé ?
- Devez-vous rejouer chaque sauvegarde différentielle effectuée depuis la dernière sauvegarde complète ?
- A votre avis, que faire si un développeur a supprimé une table par erreur il y a une heure ?
  - Pouvez-vous récupérer cette erreur avec chaque mode de récupération ?



## 5 / Restauration

37

Syntaxe :

```
RESTORE { DATABASE | LOG }
        { nom_base | @var_nombase }
  [ FROM <source>
  [ WITH { [ RECOVERY | NORECOVERY ]
          [ , <options> ] } ]
```

**RECOVERY** (par défaut) : la base est immédiatement mise à disposition des utilisateurs une fois la sauvegarde terminée

**NORECOVERY** : la sauvegarde est pendante, elle attend d'autres fichiers de restauration pour être complétée

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Lorsque vous avez plusieurs jeux de sauvegardes à restaurer (par exemple, une complète, une différentielle et des journaux de transaction), il faut interdire aux utilisateurs d'exploiter la base immédiatement après que l'on ait passé le premier fichier de restauration. Pour cela il faut mettre l'option **NORECOVERY**, sauf sur l'ultime fichier.

L'option **RECOVERY** est par défaut.

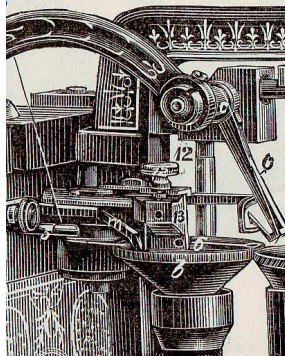
Vous pouvez changer le nom de la base au moment de la restauration. Il suffit tout simplement de donner le nom de base que vous voulez.

## 5 / Récupération

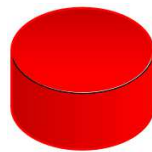
38

**RECOVERY** : consiste à rejouer les dernières transactions du journal, passé le dernier checkpoint, qui n'ont pas encore été répercutées dans les fichiers de données.

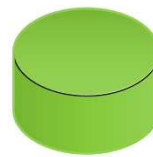
### Moteur SQL Server



Base en  
« récupération »



Base en ligne



Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Un exemple :

Restauration de la dernière sauvegarde complète en mode d'attente (NORECOVERY) :

```
RESTORE DATABASE DB_COMPTA  
FROM DISK = '\\FILER\SAVE\SQL\DB_COMPTA.BAK'  
WITH NORECOVERY;
```

Restauration de la sauvegarde transactionnelle de 8h en mode d'attente (NORECOVERY) :

```
RESTORE LOG DB_COMPTA  
FROM DISK = '\\FILER\SAVE\SQL\DB_COMPTA_0800.TRN'  
WITH NORECOVERY;
```

Restauration de la sauvegarde transactionnelle de 9h avec récupération (RECOVERY) :

```
RESTORE LOG DB_COMPTA  
FROM DISK = '\\FILER\SAVE\SQL\DB_COMPTA_0900.TRN'  
WITH RECOVERY;
```

À ce moment seront rejouées toutes les transactions finales dont les données n'ont pas pu être reportées dans les fichiers de données.

## 5 / Options de restauration

39

Beaucoup d'options de sauvegarde sont disponible pour la restauration :

- PASSWORD
- MEDIANAME
- MEDIAPASSWORD
- BLOCKSIZE
- BUFFERCOUNT
- MAXTRANSFERSIZE
- CHECKSUM / NO\_CHECKSUM
- STOP\_ON\_ERROR / CONTINUE\_AFTER\_ERROR
- STATS

Voir la description de ces options dans la partie sauvegarde

## 5 / Options de restauration

40

**RESTART** : relance une restauration précédemment interrompue.

**RESTRICTED\_USER** : Restreint l'accès à la base de données une fois restaurée aux membres des rôles db\_owner, db\_creator ou sysadmin.

**KEEP\_REPLICATION** : conserve les paramètres de réplication lors de la restauration des journaux de transaction

**KEEP\_CDC** : conserve les paramètres de CDC lors de la restauration des journaux de transactions

**ENABLE\_BROKER | ERROR\_BROKER\_CONVERSATIONS |**

**NEW\_BROKER** : active ou désactive la remise des messages de Service Broker

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

CDC (Change Data Capture) : outil de trace du changement (INSERT, UPDATE, DELETE avec timing) afin d'alimenter un DataWarehouse en mode différentiel

Service Broker est un middleware orienté message destiné à transmettre des données d'un serveur à l'autre (SODA : Service Oriented Database Architecture)

Autres options particulières

**CREDENTIAL** : uniquement pour Azure (chiffrement par accréditation)

## 5 / Restauration avec déplacement

41

Lors de la restauration, les fichiers de la base sont recréés et placés à leur endroit d'origine.

Sur un autre serveur ou bien lorsque l'on désire restaurer la base en parallèle à celle en production, il faut « déplacer » les fichiers.

Ceci se fait avec l'option MOVE :

```
MOVE 'nom_fichier_logique'  
TO 'nouvel_emplacement_physique'
```

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Exemple :

```
RESTORE DATABASE DB_COMPTA  
FROM DISK = '\\FILER\SAVE\SQL\DB_COMPTA.BAK'  
WITH MOVE 'DB_COMPTA_DATA' TO 'C:\MesBases\compatibilité_data.mdf',  
MOVE 'DB_COMPTA_TRAN' TO 'C:\MesBases\compatibilité_JT.ldf';
```

## 5 / Restauration à un point de temps

42

Il est 11:12

Un développeur de votre équipe vient vous voir ...

... il est livide

Il vous explique qu'il a malencontreusement supprimé une table  
il y a vingt minutes ...

... son doigt a glissé sur la souris

La question est alors la suivante :

- Comment faire pour récupérer les données de la table et vous assurer de ne perdre aucune autre modification faite dans la base avant et ... depuis ?

## 5 / Restauration à un point de temps

43

C'est la restauration à un point dans le temps... (PITR)

Différentes possibilités :

- STOPAT : la restauration s'arrête à la date et l'heure spécifiée
- STOPATMARK : la restauration s'arrête à la marque transactionnelle
- STOPBEFOREMARK : la restauration s'arrête avant la marque transactionnelle

Marque transactionnelle : marque ou LSN

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Les transactions peuvent être marquées (BEGIN TRANSACTION WITH MARK N'Ma marque');)

Rappel : LSN = Log Segment Number. Pour lire le journal de transaction : SELECT \* FROM sys.fn\_dblog(NULL, NULL);

Nécessite le mode de récupération FULL.

Syntaxes :

RESTORE DATABASE ...

```
WITH { STOPAT      = { 'datetime' | @datetime_var }  
      | STOPATMARK = 'lsn:lsn_number' [ AFTER 'datetime']  
      | STOPBEFOREMARK = 'lsn:lsn_number' [ AFTER 'datetime'] }
```

RSTORE LOG ...

```
WITH { STOPAT      = { 'datetime' | @datetime_var }  
      | STOPATMARK  = { 'mark_name' | 'lsn:lsn_number' } [ AFTER 'datetime']  
      | STOPBEFOREMARK = { 'mark_name' | 'lsn:lsn_number' } [ AFTER 'datetime']  
}
```

## 5 / Restauration avec écrasement

44

Lorsque l'on veut restaurer une base déjà présente et l'écraser, il suffit de rajouter l'option REPLACE.

Ceci détruit la base existante (même nom) à condition qu'aucun utilisateur n'y soit connecté.

Pour « dégager » les utilisateurs d'une base :

```
USE ma_base;  
ALTER DATABASE ma_base  
    SET SINGLE_USER WITH ROLLBACK IMMEDIATE;  
USE master;
```

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Exemple :

```
USE DB_COMPTA;  
ALTER DATABASE DB_COMPTA  
    SET SINGLE_USER WITH ROLLBACK IMMEDIATE;  
USE master;
```

```
RESTORE DATABASE DB_COMPTA  
FROM DISK = '\\FILER\SAVE\SQL\DB_COMPTA.BAK'  
WITH REPLACE
```



## 5 / Restauration fragmentaire

45

Lorsque qu'une stratégie de sauvegarde par fichier et groupe de fichiers a été mis en place, il est possible de faire apparaître la base alors que la restauration est en cours.

- Option PARTIAL :
  - Commencer par le groupe de fichier PRIMARY
  - Continuez dans l'ordre des fichiers ou groupe de fichier les plus important.
- Dès que PRIMARY est en ligne, la base commence à apparaître partiellement
- Elle reste en lecture jusqu'à complète restauration

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

La version Enterprise permet de faire apparaître la base au cours de la restauration. Pour les autres éditions, la base est hors ligne.

Pour de plus amples informations :

<https://msdn.microsoft.com/fr-fr/library/ms177425%28v=sql.120%29.aspx>

## 5 / Sauvegarde après défaillance

46

Lorsque qu'une base de données connaît une défaillance majeure, il est intéressant de tenter une sauvegarde de la fin du journal de transaction (« tail log backup »), même si la base semble inaccessible.

Ceci capture les transactions des derniers instant de vie de la base, à condition que le fichier du journal ne soit pas endommagé

```
BACKUP LOG ...  
WITH [ NORECOVERY, ] NO_TRUNCATE
```

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

**NORECOVERY** : met la base de données dans l'état de restauration à la fin de l'opération de sauvegarde

**NO\_TRUNCATE** (=> **COPY\_ONLY** et **CONTINUE\_AFTER\_ERROR**) ne réinitialise pas le journal et continue la sauvegarde même si des erreurs surviennent.

Vous pourrez alors restaurer l'intégralité de la base avec ses derniers instant de vie et ainsi de ne perdre aucune donnée !

Réussite potentielle si la base est endommagée ou hors connexion.

En mode **BULK LOGGED**, si le journal contient des modifications journalisées en bloc, la sauvegarde ne réussit que si tous les fichiers de données sont présents et intacts (pour récupérer les extensions).

Peut contenir des métadonnées de sauvegarde incomplètes (mais le journal capturé est complet et exploitable), parce que **CONTINUE\_AFTER\_ERROR** est utilisé. Vérifier avec **RESTORE HEADERONLY** et lisez la colonne **HasIncompleteMetadata**. Si = 1, alors il manque certaines informations sur les groupes de fichiers. Dans ce cas, la réussite de la restauration est compromise.

## 5 / Restauration de *master*

47

Plusieurs cas :

- Master endommagé, mais SQL Server fonctionnel : restaurez master
- Master endommagé, et SQL Server non fonctionnel : récupérez des fichiers propres de master, placez les à l'endroit de ceux endommagés puis restaurez master
- Master endommagé et SQL Server inaccessible : reconstruire la base de données master à l'aide de `setup.exe`

Administration SQL Server - CH 05 - Sauvegardes et restauration

V3.0 10/07/2017

Restauration de master d'après le média d'installation :

```
start /wait <CD ou DVD>\setup.exe /qn INSTANCENAME=<NomInstance>  
REINSTALL=ALL REINSTALLMODE = AMUS REBUILDDATABASE=1  
SAPWD=<MotDePasseSA>
```

Pour l'instance par défaut, spécifiez : MSSQLSERVER

La base resource est écrasée, il faut donc réappliquer les service pack

- /qn = supprime les boîtes de dialogues et les erreurs (inscrites dans le log)
- /qb = affiche les boîtes de dialogue de base et les erreurs

REINSTALLMODE = AMUS

- A - Force la réinstallation de tous les fichiers
- M - Ecrit les clés de registre de machine locale
- U - Ecrit les clés de registre de l'utilisateur
- S - Réinstalle les raccourcis

## 5 / Restauration de *master*

48

Pour restaurer master :

- Démarrer SQL Server en mode mono-utilisateur en modifiant les paramètres de lancement
  - `sqlservr.exe -c -m`
    - -c = ne pas démarrer en service
    - -m = mono-utilisateur
- Restaurer master
- Supprimez les paramètres de lancement
- Redémarrer le serveur

Tentez avant tout de restaurer master directement !

## 5 / Planification des sauvegardes

49

Utilisez l'Agent SQL : tout sera tracé dans l'instance...

N'oubliez pas de vous entraîner à la restauration

Si la sauvegarde est facilement automatisable, la restauration est impossible à mécaniser et les pièges sont très nombreux !


Passez du temps à maquetter vos restaurations dans un environnement aussi proche que possible de la production !



# SQL SERVER 2012/2017 ADMINISTRATION

Frédéric Brouard (SQLpro) [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
Richard Gaillard - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)

Version 3.001  
Date 10/07/2017



**6** Surveillance

Administration SQL Server - CH 06 - Surveillance V3.0 10/07/2017

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

## 6 – Surveillance & performances

3



- ❑ L'historisation des données
- ❑ Les journaux d'événement
- ❑ L'analyseur de performances
- ❑ Les DMV et proc. stockées d'exécution
- ❑ Le profiler SQL
- ❑ Les événements étendus
- ❑ Verrouillage et blocages
- ❑ Le collecteur de données
- ❑ Le gouverneur de ressources



## 6 / Historisation des données

4

### Historisation fonctionnelle :

- Tables temporelles

### Historisation pour la BI :

- Change Tracking : note les lignes qui ont évoluées
- Change Data Capture : copie les lignes qui ont évoluées

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Tout ce qui concerne la BI est hors du champs de ce cours.

CHANGE TRACKING indique quelles lignes ont évoluées, mais ne permet pas de savoir ni le nombre de fois, ni d'avoir les valeurs intermédiaire. La capture des informations pour alimenter un DataWarehouse se fait en effectuant une requête qui croise les métadonnées de l'évolution avec les tables de la base de production. C'est une solution légère (avantage) dont l'inconvénient est de devoir lire les lignes des tables de production (donc pose de verrous)

À lire : <https://docs.microsoft.com/fr-fr/sql/relational-databases/track-changes/about-change-tracking-sql-server?view=sql-server-2017>

CHANGE DATA CAPTURE copie toutes les modifications dans des tables à part dans un schéma SQL spécifique. Des fonctions table particulière permettent de croiser les métadonnées données et les données ayant évoluées sans jamais accéder aux tables de production. La solution est donc plus lourde (inconvénient) mais ne pose pas de verrou sur les tables de production (avantage)

À lire : <https://docs.microsoft.com/fr-fr/sql/relational-databases/track-changes/about-change-data-capture-sql-server?view=sql-server-2017>

## 6 / Historisation des données

5

Tables temporelles, concepts :

- ❑ **Table temporalisée** : une table de production dont on rajoute 2 colonnes techniques (début et fin de validité) qui peuvent être « hidden » et qui sont alimentées automatiquement à chaque INSERT, UPDATE.
- ❑ **Table d'historisation** : table technique contenant l'historique des évolutions de chacune des lignes de la table à laquelle elle est associée, avec la période de validité (début et fin).

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Norme SQL :

ISO/IEC TR 19075-2:2015 : Information technology – Database language – SQL Technical Report

Part 2: SQL Support for Time-Related Information

La date de début de validité contient la dateheure UTC à laquelle l'INSERT ou l'UPDATE a eut lieu

La date de fin de validité contient toujours la dateheure de la fin des temps SQL (9999-12-31 23:59:59.9999999)

La propriété « hidden » ne fait pas apparaître ces colonnes dans le cas du SELECT \*

La table d'historisation peut être située logiquement dans un schéma SQL différent et stockée dans un groupe de fichier spécifique.

## 6 / Historisation des données

6

- ❑ **La table temporalisée** : contient les données **actuelles**.
- ❑ **La table d'historisation** : contient les données **passées**.

**NOTA** : le terme « table temporelle » parle du concept et rassemble les deux tables.

L'intervalle de temps formé par les horodatages de début et des fin est fermé à droite et ouvert à gauche

[ dh\_debut, dh\_fin [

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Pour cet intervalle de temps on parle de période de validité transactionnelle. Le but de cet forme d'intervalle ( [... [ ) est nécessaire pour la continuité temporelle sans chevauchement des intervalles.

La table d'historisation voit ses données compressés. En fait c'est un mixte entre vue et table, le stockage logique se limitant aux valeurs modifiées, colonnes composant la clef et horodatages de début et fin.

La table d'historisation peut être renommée, changer de schéma, déplacé au niveau du stockage, agrémentées d'index et même reconstruite. Elle possède un index clustered qui peut même être transformé en index columnstore. On peut enfin la placer dans le cloud Azure (stretch table).

## 6 / Historisation des données

7

### Mise à jour et historisation :

| Commande | Table temporalisée  | Table d'historisation  |
|----------|---|--|
| INSERT   | dh_début = UTC INSERT, dh_fin = 9999-12-31 23:59:59.9999999 |  |
| UPDATE   | dh_début = UTC INSERT, dh_fin = 9999-12-31 23:59:59.9999999 | ajout d'une nouvelle ligne avec les anciennes valeur<br>début = valeur originale du début<br>fin = date heure UTC UPDATE |
| DELETE   | <i>La ligne est supprimée</i>                               | ajout d'une nouvelle ligne avec les anciennes valeur<br>début = valeur originale du début<br>fin = date heure UTC DELETE |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Le MERGE fonctionne suivante la commande finale INSERT, UPDATE ou DELETE.

Le BULK INSERT est un INSERT

Le TRUNCATE TABLE n'est pas supporté.

## 6 / Historisation des données

8

### Interrogation temporelle :

| Opérateur           | Paramètre   | Description  |
|---------------------|---|--|
| <b>AS OF ...</b>    | <date_heure>                                      | données telles qu'elles étaient à ce moment                                  |
| <b>FROM ...</b>     | <debut_date_heure><br><b>TO</b> <fin_date_heure>  | Période située en tout ou partie dans l'intervalle : [ ... ] (fermé)         |
| <b>BETWEEN</b>      | <debut_date_heure><br><b>AND</b> <fin_date_heure> | Période située en tout ou partie dans l'intervalle [ ... [ (ouvert à droite) |
| <b>CONTAINED IN</b> | ( <debut_date_heure><br>, <fin_date_heure> )      | Période située en totalité dans l'intervalle [ ... ] (fermé)                 |
| <b>ALL</b>          |   | Toutes les données   |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Ces opérateurs normalisés sont à utiliser dans la clause FOR SYSTEM\_TIME qui est ajouté à la suite du nom de la table historisée dans la clause FROM

## 6 / Historisation des données

9

Syntaxes pour la table historisée :

```
nom_col_début DATETIME2[(n)]
    GENERATED ALWAYS AS ROW START [HIDDEN] NOT NULL
    DEFAULT SYSUTCDATETIME(),
nom_col_fin DATETIME2 [(n)]
    GENERATED ALWAYS AS ROW END [HIDDEN] NOT NULL
    DEFAULT '9999-12-31T23:59:59.9999999',
PERIOD FOR SYSTEM_TIME (nom_col_début, nom_col_fin)
WITH (SYSTEM_VERSIONING = ON)
```

HIDDEN : (facultatif) pour ne pas exposer la colonne

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

N étant la précision en centième de seconde [1..7]. Par défaut = 7.

Pour attribuer la propriété HIDDEN après coup, utilisez la commande :  
ALTER TABLE ... ALTER COLUMN ... ADD HIDDEN

SYSTEM\_VERSIONING = ON indique que la chronodatation est active.

Pour ajouter la période de validité à une table existante :

```
ALTER TABLE MaTable
    ADD DHT_debut DATETIME2 GENERATED ALWAYS AS ROW START
    HIDDEN NOT NULL DEFAULT SYSUTCDATETIME(),
    DHT_fin DATETIME2 GENERATED ALWAYS AS ROW END
    HIDDEN NOT NULL DEFAULT SYSUTCDATETIME(),
    PERIOD FOR SYSTEM_TIME (DHT_debut, DHT_fin);
```

## 6 / Historisation des données

10

Syntaxes pour la table d'historisation :

Il faut modifier la table source (historisée) de la sorte :

Lors de la création :

```
WITH (SYSTEM_VERSIONING = ON  
      (HISTORY_TABLE = MaTable_Historique));
```

Après coup :

```
ALTER TABLE MaTable  
      SET (SYSTEM_VERSIONING = ON  
          (HISTORY_TABLE = MaTable_Historique));
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

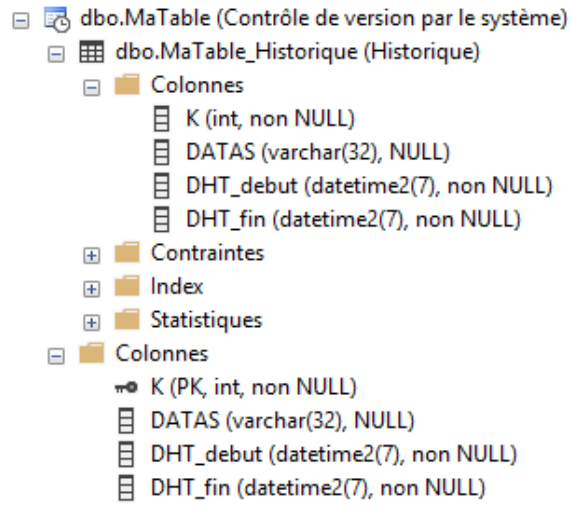
On peut aussi récupérer une table ayant déjà historisé les données de manière fonctionnelle.

```
ALTER TABLE MaTable  
      SET (SYSTEM_VERSIONING = ON (HISTORY_TABLE = MaTable_Historique));
```

## 6 / Historisation des données

11

Au niveau  
de l'IHM SSMS :



Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

En version 2016, la rétention étant « infinie », vous devez la purger manuellement, par exemple avec un batch de l'Agent SQL :

```
BEGIN TRANSACTION
BEGIN TRY
ALTER TABLE MaTable SET (SYSTEM_VERSIONING = OFF);
DELETE FROM MaTable WHERE DHT_fin <= DATEADD(day, -366,
SYSUTCDATETIME());
ALTER TABLE MaTable
    SET (SYSTEM_VERSIONING = ON (HISTORY_TABLE =
dbo.MaTable_Historique, DATA_CONSISTENCY_CHECK = ON));
COMMIT;
END TRY
BEGIN CATCH
    IF XACT_STATE() <> 0
        ROLLBACK;
END CATCH
```

En version 2017, la purge peut être automatisée :

```
ALTER DATABASE CURRENT SET TEMPORAL_HISTORY_RETENTION ON;
GO
ALTER TABLE MaTable SET (SYSTEM_VERSIONING = ON
(HISTORY_RETENTION_PERIOD = 12 MONTH));
```



## 6 / Les outils de monitoring

12

### Outils intégrés :

- Sécurité : C3, C2, audit de connexion, audit de base de données
- Changement des données : CHANGE TRACKING, CHANGE DATA CAPTURE, déclencheurs DML, DDL et LOGON, Notifications, Réplication de données
- Exécution des requêtes : Profiler SQL, Extended Events, Database Tuning Advisor (Assistant Paramétrage du moteur de base de données), DMV, Data Collector (Collecteur de Données), Journaux SQL et Agent
- Fonctionnement : Performance Monitor (Windows), Extended Events, Journaux systèmes
- Tests : Profiler SQL, Distributed replay

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

C3 (Comon Criterai Compliance) : via sp\_configure paramètre 'common criteria compliance enabled'

C2 (US Department of Defense) : via sp\_configure paramètre 'c2 audit mode' (obsolète)

Audit de connexion : xp\_instance\_regwrite N'HKEY\_LOCAL\_MACHINE', N'Software\Microsoft\MSSQLServer\MSSQLServer', paramètre 'AuditLevel' de type REG\_DWORD valeur 0, 1, 2 ou 3

Audit de base de données, voir la commande CREATE DATABASE AUDIT

Database Tuning Advisor est aussi appelé DTA

DMV : Data management View (dans le schéma sys, vue commençant par dm\_...)

Distributed replay est disponible sur le média d'installation

## 6 / Les outils de monitoring

13

Outils externes gratuits :

- SQLDiag, PSSDiag (Microsoft)
- PAL (performance Analysis of Logs)
- RML (Replay Markup Language)
- SQL Nexus Tool
- Kankuru

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Touts ces outils sont gratuits et accessible sur les sites suivants :

- [microsoft.com](http://microsoft.com)
- [codeplex.com](http://codeplex.com)
- [github.com](http://github.com)

Ou sur les sites éditeurs ([kankuru.fr](http://kankuru.fr))

## 6 / Monitoring de performances

14

Ceux dont nous allons parler :

- Les journaux SQL, Agent et Windows
- Le moniteur de performances
- Métadonnées d'exécution avec les DMV et les procédures systèmes
- Le profileur SQL, l'assistant DTA
- Les événements étendus
- Les déclencheurs DDL et LOGON
- Verrouillage, attentes et étreintes fatales
- Le collecteur de bases de données
- Le gouverneur de ressources

## 6 / Les journaux d'événement

15

Accessible depuis SSMS :

- SQL Server
- Agent SQL
- Messagerie de bases de données
- Windows NT
  - Application
  - Security
  - ...

Administration SQL Server - CH 06 - Surveillance & performances

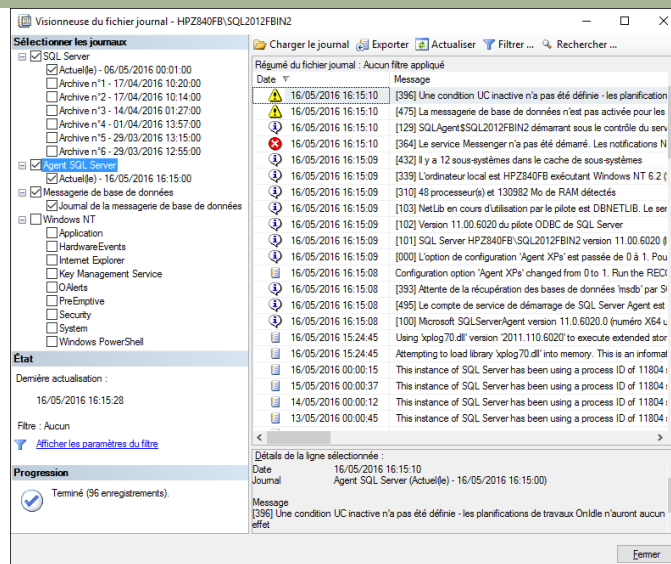
V3.0 10/07/2017

Les journaux sont appelées couramment « log » en anglais.

SQL Server écrit des événements dans les journaux Application et Sécurité

## 6 / Les journaux d'événement

16



Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Les journaux auxquels vous pouvez accéder dépendent des permissions accordées à votre connexion.

Pour les journaux SQL Server, vous pouvez augmenter le nombre de rotation (un nouveau journal est créé à chaque arrêt du service ou forçage via `sp_cycle_errorlog` / `sp_cycle_agent_errorlog`) à l'aide de la clef de registre :

“NumErrorLogs” (REG\_DWORD) située à

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQL.X\MSSQLServer\

Par défaut elle n'existe pas, donc la créer.

## 6 / Les journaux d'événement

17

En dehors de l'IHM, vous pouvez rechercher des informations par requête via la procédure :

`xp_readerrorlog`

Les 7 paramètres sont :

- ordinal (n° du fichier à partir de 0, fichier actuel)
- Type (1 ou NULL error log, 2 SQL Agent log)
- Motif 1 (chaîne de caractères UNICODE à trouver)
- Motif 2 (chaîne UNICODE à trouver en complément)
- DateDebut
- DateFin
- Ordre (« asc » ou « desc » pour trier les données relativement au temps)

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Exemples :

```
EXEC master.dbo.xp_readerrorlog 0, 1, N'backup', NULL, N'2017-01-02', N'2017-02-02', N'desc'
```

```
EXEC master.dbo.xp_readerrorlog 0, 1, N'backup', N'failed', NULL, NULL, N'asc'
```

## 6 / Les journaux d'événement

18

Exemple :

```
EXEC xp_readerrorlog 0, 1, N'taking longer than ';
```

Ceci recherche dans le fichier d'événement actuel les attentes (n générale de plus de 15 secondes) pour l'écriture des fichiers.

Vous pouvez aussi recherches les « Login failed », « Logon failed »

## 6 / L'analyseur de performances

19

### Perfmon.exe

Prélève régulièrement des métriques (compteurs) au niveau :

- Systèmes (os Windows)
- Et pour les applications qui en publie (chaque instance/service SQL Server en publie plus d'une centaines

Possibilité de tracer ces informations dans un fichier.

- Corrélation possible avec le Profiler SQL

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Performance Monitor pour son nom anglais.

Il existe des outils complémentaire au moniteur de performance :

- Netdiag.exe pour l'aspect réseau
- Dcdiag.exe pour des problèmes liés aux contrôleurs de domaines

Vous pouvez commencer par lancer un rapport global perfmon avant toute action afin d'avoir l'état générale de la machine :

perfmon /report

Enregistrez une session de métriques des compteurs sous forme de fichier. Activez le profiler SQL en même temps. Vous pourrez alors corréler les deux.



## 6 / L'analyseur de performances

20

| Groupe (* instance SQL)      | Compteur   |
|------------------------------|--|
| Physical disk / Logical Disk | Avg. Disk Queue Length, % Disk Time                                  |
| Process                      | %Processor Time (SQL Server)   |
| System                       | % Total Processor Time, Processor Queue Length                       |
| * Access Methods             | Full Scans / sec., Index Searches / sec., Page Splits / Sec          |
| * Buffer Manager             | Buffer cache hit ratio, Page life expectancy, Checkpoint Pages / Sec |
| * Databases                  | Active transactions, Log cache hit ratio, Log growths                |
| * General Statistics         | Processes Block  |
| * Locks                      | Lock Waits / Sec (_Total)  |
| * SQL Statistics             | Batch Requests/Sec, SQL Compilations/Sec, SQL Re-Compilations/Sec    |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Pour le monitoring des disques, lire :

<https://blogs.msdn.microsoft.com/askjay/2011/07/08/troubleshooting-slow-disk-io-in-sql-server/>

Pour le monitoring système autour de SQL Server, lire :

<https://www.sqlskills.com/blogs/jonathan/the-accidental-dba-day-21-of-30-essential-perfmon-counters/>

Quest fournit une liste des compteurs les plus intéressants pour SQL Server sous forme de poster en pdf :

[http://helsinki.sqlpass.org/Portals/139/Files/SQL\\_post\\_29x21\\_2010\\_PerfmonFinal.pdf?ver=2013-12-01-010942-587](http://helsinki.sqlpass.org/Portals/139/Files/SQL_post_29x21_2010_PerfmonFinal.pdf?ver=2013-12-01-010942-587)

## 6 / DMV et procédures d'exécution

21

### Data Management View : surveiller en « live »

| Catégorie de DMV    | Utilisation                             |
|---------------------|---|
| .dm_os_ (46)        | Activité système de SQL Server          |
| .dm_db_ (43)        | Métriques à travers les bases           |
| .dm_exec_ (40)      | Activité logiques en cours              |
| .dm_tran_ (16)      |   |
| .dm_fts_ (13)       | Activités de l'indexation « full text » |
| .dm_xe_ (9)         | Fonctionnement de « Extended Events »   |
| .dm_io_ (5)         | E/S fichier et mémoire                  |
| .dm_server_ (4)     | Objets de niveau « serveur » (instance) |
| .dm_filestream_ (3) | Activité du FILESTREAM                  |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**ATTENTION** : la structure des DMV est constamment remise à jour de version en version. De nouvelles DMV peuvent apparaître dans les « Service Pack ». Vérifiez bien l'adéquation DMV / Structure de DMV à votre version / service pack.

En savoir plus sur les DMV :

<http://www.sqlservergeeks.com/consolidated-dmv-scripts-part-1/> (série d'articles)

SQL Server DMVs in Action (livre Manning 2011)

Performance Tuning with SQL Server Dynamic Management Views (livre Red Gate 2010)

**NOTA** : les DMV ne peuvent être lues qu'en mode d'isolation READ COMMITTED (sauf exception). Minimisez l'impact en production en évitant le SELECT \* !

Autres DMV sur hadr (haute disponibilité via AlwaysOn), xtp (in memory), broker (service broker), cryptographic (EKM), logpool (buffer pool), repl (réplication de données), ...

## 6 / DMV et procédures d'exécution

22

### Activités logiques en cours d'exécution (dm\_exec) :

| DMV                      | Description  |
|--------------------------|--|
| .dm_exec_connections     | Connexions en cours dans le serveur                    |
| .dm_exec_sessions        | Sessions actives dans le serveur                       |
| .dm_exec_requests        | Requêtes en cours de traitement dans le serveur        |
| .dm_exec_sql_text        | (OF) Texte d'une requête via son identifiant           |
| .dm_exec_query_plan      | (OF) Plan d'exécution relatif à un identifiant de plan |
| .dm_exec_query_stats     | Statistiques d'exécution des requêtes                  |
| .dm_exec_procedure_stats | Statistiques d'exécution des procédures                |
| .dm_exec_trigger_stats   | Statistiques d'exécution des déclencheurs              |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**RAPPEL** : Toutes ces DMV sont dans le schéma sys.

OF : online function

La liste n'est pas exhaustive

## 6 / DMV et procédures d'exécution

23

Activités logiques en cours d'exécution, exemples :

```
SELECT *
FROM sys.dm_exec_connections
      CROSS APPLY
           sys.dm_exec_sql_text(most_recent_sql_handle)

SELECT TOP 10 *
FROM   sys.dm_exec_query_stats AS qs
      CROSS APPLY sys.dm_exec_query_plan(plan_handle)
      CROSS APPLY sys.dm_exec_sql_text(sql_handle)
ORDER BY total_worker_time DESC;
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Notez l'utilisation des fonctions table en ligne "sys.dm\_exec\_query\_plan" et "sys.dm\_exec\_sql\_text" via l'opérateur APPLY.

La première requête liste les connexions en cours et montre la dernière requête exécutée ou en cours d'exécution.

Le seconde requête présente les 10 requêtes les plus consommatrices en temps de travail avec les métriques et le plan de requête associé.

## 6 / DMV et procédures d'exécution

24

### Activité de l'OS SQL Server :

| Table_name                  | Objet   |
|-----------------------------|---|
| .dm_os_buffer_descriptors   | Pages de données (tables, index) en cache             |
| .dm_os_latch_stats          | Attentes de verrous interne par nature                |
| .dm_os_memory_*             | Fonctionnement de la RAM allouée à l'instance         |
| .dm_os_nodes                | Fonctionnement des nœuds CPU                          |
| .dm_os_performance_counters | Compteurs de perf. Spécifiques à l'instance (perfmon) |
| .dm_os_process_memory       | Quantité de RAM allouée à l'instance par zone         |
| .dm_os_schedulers           | « Planificateur » par type et cœur (avec nœud numa)   |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**RAPPEL** : Toutes ces DMV sont dans le schéma sys. La liste n'est pas exhaustive

La vue sys.dm\_os\_latch\_stats présente des données cumulées depuis le démarrage du serveur ou la remise à zéro effectuée manuellement.

Vous pouvez remettre à zéro la vues via la commande : DBCC SQLPERF ('sys.dm\_os\_latch\_stats', CLEAR);

## 6 / DMV et procédures d'exécution

25

### Activité de l'OS SQL Server :

| DMV                  | Description                                    |
|----------------------|--|
| .dm_os_sys_info      | Information de l'instance                      |
| .dm_os_threads       | Threads utilisés par l'instance                |
| .dm_os_tasks         | Tâches actives dans l'instance                 |
| .dm_os_volume_stats  | (OF) occupation physique du stockage (disques) |
| .dm_os_wait_stats    | Statistiques cumulées des attentes par nature  |
| .dm_os_waiting_tasks | Tâches actives en attente de ressources        |
| .dm_os_windows_info  | Information sur l'OS Windows                   |
| .dm_os_workers       | Processus en cours dans l'instance             |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**RAPPEL** : Toutes ces DMV sont dans le schéma sys. La liste n'est pas exhaustive.

OF : Online Function (fonction « table en ligne »)

La vue `sys.dm_os_wait_stats` présente des données cumulées depuis le démarrage du serveur ou la remise à zéro effectuée manuellement.

Vous pouvez remettre à zéro la vues via la commande : `DBCC SQLPERF ('sys.dm_os_wait_stats', CLEAR);`

## 6 / DMV et procédures d'exécution

26

### Activités OS en cours, exemples :

```
SELECT COUNT(*)
FROM sys.dm_io_pending_io_requests
WHERE io_type = 'disk' and io_pending_ms_ticks > 8;
```

```
SELECT DISTINCT COALESCE(DB_NAME(database_id),
'mssqlsystemresource') AS DATABASE_NAME,
        numa_node, COUNT(*) OVER() AS NOMBRE,
        COUNT(*) OVER(PARTITION BY database_id) AS NOMBRE_PAR_BASE,
        COUNT(*) OVER(PARTITION BY numa_node) AS NOMBRE_PAR_NOEUD
FROM sys.dm_os_buffer_descriptors AS bf
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

La première requête affiche le nombre d'accès disque attendant plus de 250 ms pour les transactions actives

La seconde requête affiche la répartition du cache des données par base et noeud numa

## 6 / DMV et procédures d'exécution

27

### Activité transactionnelle :

| DMV  | Description   |
|--|---|
| .dm_tran_active_snapshot_database_transactions | Transactions actives ayant des lignes versionnées (mode d'isolation snapshot) |
| .dm_tran_active_transactions                   | Transactions actives  |
| .dm_tran_current_snapshot                      | Transactions actives en mode d'isolation snapshot                             |
| .dm_tran_current_transaction                   | État des transactions en cours  |
| .dm_tran_database_transactions                 | Transaction en cours relatives aux bases de données                           |
| .dm_tran_locks                                 | Verrous posés par une transaction en cours                                    |
| .dm_tran_session_transactions                  | Transaction en cours relatives aux sessions                                   |
| .dm_tran_transactions_snapshot                 | N° de séquence des versions de lignes des transactions en mode snapshot       |
| .dm_tran_version_store                         | Enregistrements relatifs aux lignes versionnées (snapshot)                    |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**RAPPEL** : Toutes ces DMV sont dans le schéma sys. La liste n'est pas exhaustive.



## 6 / DMV et procédures d'exécution

28

### Activité transactionnelle :

```
SELECT DATEDIFF(second, transaction_begin_time, GETDATE())
       AS SECOND_DURATION_TRANSACTION, *
FROM   sys.dm_exec_requests AS r
       LEFT OUTER JOIN sys.dm_tran_session_transactions AS st
         ON r.session_id = st.session_id
       LEFT OUTER JOIN sys.dm_tran_active_transactions AS at
         ON st.transaction_id = at.transaction_id
       CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) AS q
       INNER JOIN sys.dm_exec_sessions AS s
         ON r.session_id = s.session_id
       INNER JOIN sys.databases AS db
         ON s.database_id = db.database_id
WHERE  DATEDIFF(second, transaction_begin_time, GETDATE()) > 30
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Cette requête affiche des informations sur les transactions durant plus de 30 secondes.

## 6 / DMV et procédures d'exécution

29

### Activité des bases de données :

| DMV                               | Description   |
|-----------------------------------|---|
| .dm_db_database_page_ allocations | (OF) allocation des pages de la base dans les index |
| .dm_db_log_space_usage            | Espace alloué dans le journal de transaction        |
| .dm_db_file_space_usage           | Espace alloué dans les fichier de données           |
| .dm_db_partition_stats            | Distribution dans les partitions                    |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**RAPPEL** : Toutes ces DMV sont dans le schéma sys. La liste n'est pas exhaustive.

OF : Online Function (fonction « table en ligne »)

On peut y ajouter les DMV relatives aux index pour leur utilisation :

- sys.dm\_db\_index\_operational\_stats
  - sys.dm\_db\_index\_physical\_stats
  - sys.dm\_db\_index\_usage\_stats
  - sys.dm\_db\_fts\_index\_physical\_stats
- ... comme pour les index manquants :
- sys.dm\_db\_missing\_index\_columns
  - sys.dm\_db\_missing\_index\_details
  - sys.dm\_db\_missing\_index\_group\_stats
  - sys.dm\_db\_missing\_index\_groups
- ... et celle consacrées au mirroring :
- sys.dm\_db\_mirroring\_auto\_page\_repair
  - sys.dm\_db\_mirroring\_connections
  - sys.dm\_db\_mirroring\_past\_actions

## 6 / DMV et procédures d'exécution

30

### Activité des bases de données :

| DMV de version / édition                              | Description                                     |
|---|---|
| .dm_db_objects_disabled_on_compatibility_level_change | (OF) Objets désactivés si changement de version |
| .dm_db_persisted_sku_features                         | Fonctionnalités non portables entre éditions    |

| DMV spécifique tempdb      | Description                                       |
|----------------------------|---|
| .dm_db_session_space_usage | Allocation et désallocation des pages par session |
| .dm_db_task_space_usage    | Allocation et désallocation des pages par tâche   |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**RAPPEL** : Toutes ces DMV sont dans le schéma sys. La liste n'est pas exhaustive.

OF : Online Function (fonction « table en ligne »)

## 6 / DMV et procédures d'exécution

31

### Activité des Entrées / Sorties (IO) :

| DMV                          | Description  |
|------------------------------|--|
| .dm_io_virtual_file_stats    | (OF) Statistiques d'IO cumulées sur les fichiers       |
| .dm_io_pending_io_requests   | Requêtes en cours en attente d'IO                      |
| .dm_io_backup_tapes          | Liste des destinations des sauvegardes « device »      |
| .dm_io_cluster_shared_drives | Liste les disques partagés dans le « cluster » Windows |

**RAPPEL** : Toutes ces DMV sont dans le schéma sys. La liste n'est pas exhaustive.

OF : Online Function (fonction « table en ligne »)

## 6 / DMV et procédures d'exécution

32

### Procédures utiles :

- `sp_who`, `sp_who2` : activité des sessions en cours
- `sp_lock` : indique les verrous en cours (**obsolète**)

Vous pouvez construire votre propre procédures pour lister les verrous...

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**sp\_who** prend éventuellement un seul paramètre qui peut être un nom de connexion (entre apostrophe), un identifiant de session (numérique) ou le mot clef ACTIVE (entre apostrophe) pour filtrage des résultats.

Le résultat est une table dont la colonne « blk » indique si différent de 0, l'id de la session bloquante.

**NOTA** : l'identifiant de session (session\_id) est dans la colonne spid.

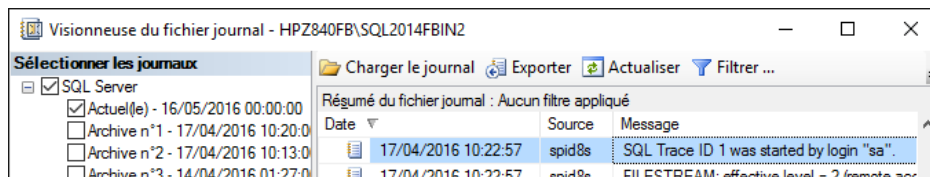
**sp\_who2** fournit des indications complémentaire au niveau du temps CPU consommé, des ES disque, du démarrage de la session et du programme qui a lancé la requête.

## 6 / Le profiler SQL

33

Considéré comme obsolète !

*Mais Microsoft l'utilise toujours pour les diagnostics internes...*



Le remplaçant est « eXtend Events »

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**eXtended Events** (événement étendus) est plus riche mais bien plus complexe et parfois très couteux. Bref, le profiler à encore de beaux jours devant lui !

## 6 / Le profiler SQL

34

Permet de tracer les opérations dans le moteur (requêtes et divers événements) :

- Pas seulement le moteur OLTP, mais aussi SSAS (XML A, MDX, DMX) et SSIS.
- Possibilité de créer des modèles (templates)
- Possibilité de filtrer les événements
- Peut enregistrer les données tracées dans une table ou un jeu de fichier (préférez les fichiers)
- Graphique (IHM) ou Transact-SQL
  - Ne jamais utiliser en IHM sur un serveur

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Une trace « fichier » peut être réinjectée sous forme de table, soit via l'IHM, soit via la fonction table `sys.fn_trace_gettable`

Les principales procédures stockées pour la manipulation du profiler sont :

- `sp_trace_create` : création de la définition de trace et du stockage
- `sp_trace_setevent` : ajout d'un attribut à tracer sur un événement particulier
- `sp_trace_setfilter` : ajout d'un filtre sur attribut d'un événement
- `sp_trace_setstatus` : démarrage, arrêt, suppression

Autres routines :

- `fn_trace_geteventinfo`
- `fn_trace_getfilterinfo`
- `fn_trace_getinfo`
- `sp_trace_generateevent`
- `sp_trace_getdata`

## 6 / Le profiler SQL

35

- Certains événements sont tracés sous forme XML, interprétable dans SSMS :
  - Plan de requêtes
  - Graphes d'interblocage (deadlock)
- Vous pouvez corréler les résultats du profiler avec les compteurs du moniteur de performances
- Le template « replay » permet de rejouer une trace pour étalonnage

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

PAL (Performance Analysis of Logs) est un outil de mise en place et analyse automatique des traces du profiler (à télécharger depuis Codeplex)

RML (Replay markup language) est un utilitaire pour rejouer et comparer une trace (à télécharger depuis Codeplex)



## 6 / Le profiler SQL

36

Mise en place :

- donner un nom
- choisir un template
- prévoir l'enregistrement en fichier ou table

Propriétés de la trace

Général | Sélection des événements

Nom de la trace : Ma session de profiler

Nom du fournisseur de trace : HPZ840FB\SQL2014FBIN2

Type de fournisseur de trace : Microsoft SQL Server "2014"

Utiliser le modèle : Standard (par défaut)

Enregistrer dans le fichier :

- Standard (par défaut)
- TSQL
- TSQL\_Duration
- TSQL\_Grouped
- TSQL\_Locks
- TSQL\_Replay
- TSQL\_SPs
- Tuning

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Traditionnellement les fichiers sont d'extension .trc

Il est possible de faire un « rollover » de fichiers

## 6 / Le profiler SQL

37

- Certains événements sont tracés sous forme XML, interprétable dans SSMS, notamment, les plans de requêtes.
- Template intéressant pour l'optimisation :

| Nom           | Description   |
|---------------|---|
| TSQL          | Toutes les commandes SQL                              |
| TSQL_Duration | " " regroupées et classées par durée                  |
| TSQL_Grouped  | " " regroupées et classées par client                 |
| TSQL_SPs      | Capture l'exécution de toutes les procédures stockées |

Vous pouvez utiliser le « template » vide ou partir d'un autre template puis l'enrichir.

## 6 / Le profiler SQL

38

- Les événements capturés sont enrichis de différentes données

| Events   | TextData                            | ApplicationName                     | NTUserName                          | LoginName                           | CPU                                 |
|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| <b>Security Audit</b>                                  |                                     |                                     |                                     |                                     |                                     |
| <input checked="" type="checkbox"/> Audit Login        | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |                                     |
| <input checked="" type="checkbox"/> Audit Logout       |                                     | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| <b>Sessions</b>  |                                     |                                     |                                     |                                     |                                     |
| <input checked="" type="checkbox"/> ExistingConnection | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |                                     |
| <b>Stored Procedures</b>                               |                                     |                                     |                                     |                                     |                                     |
| <input checked="" type="checkbox"/> RPC:Completed      | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| <b>TSQL</b>  |                                     |                                     |                                     |                                     |                                     |
| <input checked="" type="checkbox"/> SQL:BatchCompleted | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| <input checked="" type="checkbox"/> SQL:BatchStarting  | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |                                     |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Choisissez les événements et les données collectées en cochant les cases appropriées

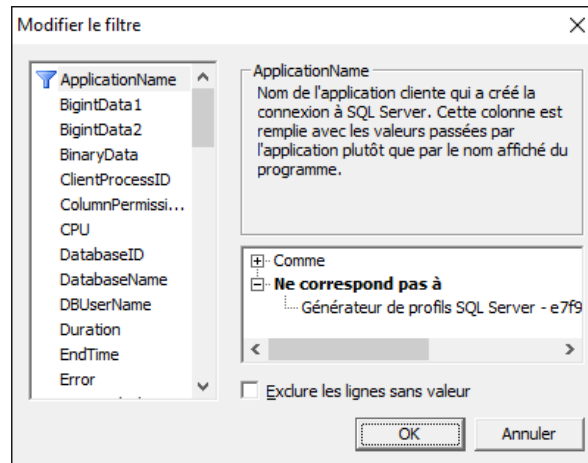
## 6 / Le profiler SQL

39

Vous pouvez filtrer les données des événements.

Par exemple pour :

- limiter à un « client »
- Limiter à une durée minimale



Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Un filtre systématique est mis en place afin d'empêcher l'auditeur (profiler SQL) de s'auditer !

## 6 / Le profiler SQL

40

Une fois collectées les données peuvent être mise en table via l'IHM ou la fonction table :

```
sys.fn_trace_gettable(...)
```

Ceci permet d'analyser le résultat par des requêtes SQL :

| Colonne           | Description                     |
|-------------------|---------------------------------|
| TextData          | La requête SQL ou procédure     |
| StartTime.EndTime | Début/fin horaire               |
| Duration          | Durée globale                   |
| CPU               | Durée CPU                       |
| RowCounts         | Nombre de lignes traitées       |
| Reads/Writes      | Nombre d'IO en lecture/écriture |

## 6 / Les événements étendus

41

### Événements Étendus

- Extended Events (XE)

Avantages :

- interface unique de diagnostic de ce qui affecte SQL Server,
  - interne (SQL, stockage)
  - externe (Windows)
- impact négligeable sur les performances

Inconvénients :

- Complexe à mettre en œuvre
- Nécessite de bien maîtriser le SQL !

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**ATTENTION** : pour SQL Server 2008 votre instance doit au moins être au niveau SP1

## 6 / Les événements étendus

42

### **Package d'XE...**

conteneur d'objets d'événements étendus

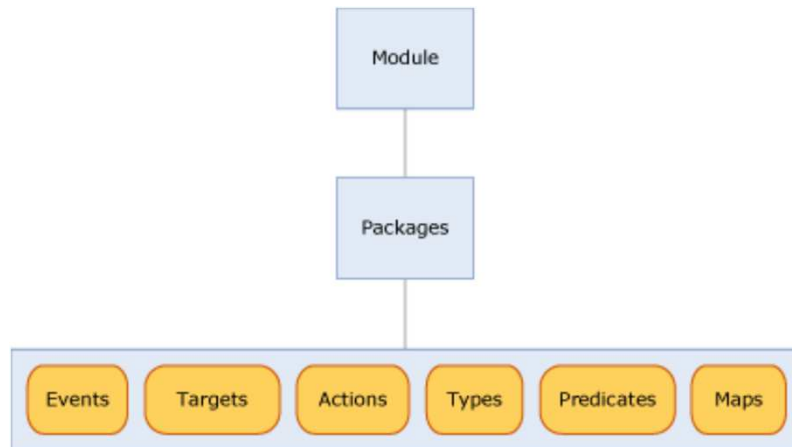
Plusieurs packages, dont :

- package0 – Événements étendus systèmes (par défaut)
- sqlserver – Objets relatifs à SQL Server.
- sqlos - SQL Server Operating System (SOS).
- SecAudit - Utilisé par SQL Server Audit. Inaccessible

## 6 / Les événements étendus

43

### Objets contenus dans un package :





## 6 / Les événements étendus

44

### Events :

- **Les événements**, (ce que l'on cherche à capturer) sont des actions déclenchées par l'exécution de code (requêtes utilisateur ou système),
- Exemple :
- une requête commence ou se termine,
- l'acquisition, de la libération, ou du time-out d'acquisition d'un verrou,
- un split de page,
- l'exécution d'un point de contrôle (CHECKPOINT),
- la fin de la restauration d'une base de données,
- la libération de pages par le gestionnaire des tampons...

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Par exemple l'événement « `xml_deadlock_report` » disponible dans le package `sqlserver` (Extended events for Microsoft SQL Server) permet de tracer sous forme XML les verrous mortels.

## 6 / Les événements étendus

45

### Connaître tous les événements de tous les packages :

```
SELECT P.name AS package_name, P.description,  
       P.capabilities_desc, O.name AS event_name  
FROM   sys.dm_xe_objects AS O  
       INNER JOIN sys.dm_xe_packages AS P  
             ON O.package_guid = P.guid  
WHERE  O.object_type = 'event'
```

### Et leurs attributs associés :

```
SELECT P.name AS package_name, O.name AS event_name,  
       OC.column_id, OC.name AS column_name, OC.column_type,  
       OC.column_value, OC.description AS column_description  
FROM   sys.dm_xe_objects AS O  
       INNER JOIN sys.dm_xe_object_columns AS OC  
             ON O.name = OC.object_name  
       INNER JOIN sys.dm_xe_packages AS P  
             ON O.package_guid = P.guid  
WHERE  O.object_type = 'event'
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Par exemple l'attribut « `source_database_id` » de l'événement `sp_statement_starting` du package `sqlserver` permet de connaître la base de données contextuelle d'exécution de l'ordre SQL

## 6 / Les événements étendus

46

### Targets :

- **Les cibles**, qui consomment les événements, et permettent de définir le réceptacle de la capture (fichiers, anneau de mémoire, journaux d'événements, .);

### Liste des cibles :

```
SELECT P.name AS package_name, O.name AS target_name,
       O.capabilities_desc AS target_capabilities,
       O.description AS target_description
FROM   sys.dm_xe_objects AS O
       INNER JOIN sys.dm_xe_packages AS P
              ON O.package_guid = P.guid
WHERE  O.object_type = 'target'
ORDER BY package_name, target_name
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

### Types de cibles :

- depuis un fichier binaire, spécifique aux événements étendus;
- directement dans la mémoire en anneau (« ring buffer ») de *SQL Server*;
- depuis le journal d'événements de *Windows (Event Tracing for Windows)*;
- en comptant le nombre d'événements capturés par une session, pour caractériser une charge de travail par exemple;
- chercher une correspondance entre deux événements d'une même session.

## 6 / Les événements étendus

47

### Les cibles ont des caractéristiques

Liste des cibles et leurs caractéristiques :

```
SELECT P.name AS package_name, O.name AS target_name,
       OC.column_id, OC.name AS column_name,
       OC.column_type, OC.capabilities_desc,
       OC.column_value, OC.description AS
column_description
FROM   sys.dm_xe_objects AS O
       INNER JOIN sys.dm_xe_object_columns AS OC
              ON O.name = OC.object_name
       INNER JOIN sys.dm_xe_packages AS P
              ON O.package_guid = P.guid
WHERE  O.object_type = 'target'
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Par exemple la cible « *asynchronous\_file\_target* » comprend 5 caractéristiques :

- filename l'emplacement et le nom du fichier de journalisation
- max\_file\_size la taille maximale des fichiers
- max\_rollover\_files le nombre maximal de fichier si itération
- Increment pas de croissance en Mo du fichier
- Metadatafile l'emplacement et le nom du fichier de métadonnées

**NOTA** : La colonne *capabilities\_desc* précise si l'attribut de la cible doit être obligatoirement spécifié (valeur *mandatory*) lors de l'utilisation de la cible dans la spécification d'une session.

## 6 / Les événements étendus

48

### **Actions :**

**Les actions**, caractéristiques des événements d'un package. Peuvent être capturées, ou permettre d'indiquer ce qui doit être fait lorsque les événements sont capturés.

Les actions sont :

- soit communes à tous les événements d'un package,
- soit spécifique à un événement particulier

Les actions sont les informations qui sont entreprises et dépendent des événements

## 6 / Les événements étendus

49

Liste des actions :

```
SELECT P.name AS package_name,
       O.name AS action_name,
       O.capabilities_desc AS
action_capabilities,
       O.description AS action_description
FROM   sys.dm_xe_objects AS O
       INNER JOIN sys.dm_xe_packages AS P
              ON O.package_guid = P.guid
WHERE  O.object_type = 'action'
ORDER BY package_name, action_name
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Par exemple dans le package *sqlserver* on trouvera entre autres les actions suivantes :

- client\_app\_name
- client\_hostname
- database\_context
- database\_id
- is\_system
- nt\_username
- sql\_text
- username
- ...

## 6 / Les événements étendus

50

### Predicates :

**Les prédicats**, permettent de filtrer les événements, tout comme dans SQL Profiler, avant que l'événement soit consommé par la cible;

### On peut par exemple :

- Limiter à un seuil (>, <...);
- Limiter le nombre d'occurrences d'un événement  
(`package0.counter = n`)
- Ne capturer qu'un échantillon  
(`package0.divides_by_uint64(package0.counter, n)`)
- Limiter la consommation de ressources de certaines actions, comme un dump mémoire ou la capture de la pile d'appel

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Deux façons d'utiliser ces prédicats :

- Par comparaison à une valeur (attribut), exemple : `sqlserver.database_id = 12`, prédicat dit « source »
  - limite à la base d'id 12
- Par application d'une fonction, exemple : `package0.less_than_equal_uint64 (package0.count , 300)` , prédicat dit « compare »
  - limite si le compteur descend en dessous de 300

## 6 / Les événements étendus

51

### Predicates...

#### Liste des prédicats :

```
SELECT P.name AS package_name,  
       P.capabilities_desc AS package_capabilities,  
       O.name AS pred_source_name, O.object_type  
FROM   sys.dm_xe_objects AS O  
       INNER JOIN sys.dm_xe_packages AS P  
               ON O.package_guid = P.guid  
WHERE  O.object_type LIKE 'pred?_%' ESCAPE '?'
```



## 6 / Les événements étendus

52

### Types

**Les types** représentent tous les types de données utilisés par l'ensemble des événements qu'il est possible de capturer avec les XE. Tout champ exposé par un événement est typé.

#### Liste des types :

```
SELECT P.name AS package_name, O.name AS type_name,
       O.type_size
FROM   sys.dm_xe_objects AS O
       INNER JOIN sys.dm_xe_packages AS P
              ON O.package_guid = P.guid
WHERE  O.object_type = 'type';
```

## 6 / Les événements étendus

53

### Maps :

Les « **maps** » sont des tables de correspondance pour les valeurs internes du moteur de base de données, afin de les rendre intelligibles.

### Liste des maps :

```
SELECT P.name, O.name AS event_name, O.object_type
FROM   sys.dm_xe_objects AS O
       INNER JOIN sys.dm_xe_packages AS P
              ON O.package_guid = P.guid
WHERE  O.object_type = 'map';
```

## 6 / Les événements étendus

54

### Maps :

On peut obtenir les différences valeurs « clés » des maps à l'aide de la requête suivante :

```
SELECT P.name AS package_name, O.name AS map_name,
       MV.map_key, MV.map_value
FROM   sys.dm_xe_objects AS O
       INNER JOIN sys.dm_xe_packages AS P
           ON O.package_guid = P.guid
       INNER JOIN sys.dm_xe_map_values AS MV
           ON MV.object_package_guid =
              O.package_guid AND MV.name = O.name
WHERE  O.object_type = 'map'
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Voici par exemple le contenu de la map « lock\_mode »

|    |       |
|----|-------|
| 0  | NL    |
| 1  | SCH_S |
| 2  | SCH_M |
| 3  | S     |
| 4  | U     |
| 5  | X     |
| 6  | IS    |
| 7  | IU    |
| 8  | IX    |
| 9  | SIU   |
| 10 | SIX   |
| 11 | UIX   |
| 12 | BU    |
| 13 | RS_S  |
| 14 | RS_U  |
| 15 | RI_NL |
| 16 | RI_S  |
| 17 | RI_U  |
| 18 | RI_X  |
| 19 | RX_S  |
| 20 | RX_U  |
| 21 | RX_X  |
| 21 | RX_X  |

## 6 / Les événements étendus

55

### Mise en œuvre

Créer une session d'événement étendu, syntaxe :

```
CREATE EVENT SESSION nom_session_événement  
ON SERVER  
{ <définition_événement> [ ,...n]  
  [ <définition_cible> [ ,...n ]  
  [ WITH ( <option_de_session> [ ,...n ] )  
}
```

## 6 / Les événements étendus

56

### Mise en œuvre

**<définition\_événement> ::**

```
{ ADD EVENT [GUID_me].nom_pack.nom_événement
  [ ( { [ SET { attribut_événement = <value> [ ,...n ] } ]
      [ ACTION ( { [GUID_me].nom_pack.nom_action [
, ...n ] } ) ]
      [ WHERE <prédicat> ]
    } ) ]
}
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

GUID\_me est le GUID marqueur du module d'événement  
nom\_pack est le nom du package

## 6 / Les événements étendus

57

### Mise en œuvre

**<definition-cible> ::=**

```
{  
    ADD TARGET [GUID_me].nom_pack.nom_cible  
    [ ( SET { nom_parametre_cible = <valeur> [ ,...n]  
    } ) ]  
}
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**MAX\_MEMORY** : Spécifie la quantité de mémoire maximale à allouer à la session pour la mise en mémoire tampon d'événement. À défaut 4 Mo.

**EVENT\_RETENTION\_MODE** : Spécifie le mode de rétention des événements à utiliser pour gérer la perte d'événements.

- **ALLOW\_SINGLE\_EVENT\_LOSS** : il est possible de perdre un événement de la session.
- **ALLOW\_MULTIPLE\_EVENT\_LOSS** : Il est possible de perdre des mémoires tampons d'événements saturées de la session contenant plusieurs événements.
- **NO\_EVENT\_LOSS** : aucune perte d'événements n'est autorisée.

**MAX\_DISPATCH\_LATENCY** : durée pendant laquelle les événements seront mis en mémoire tampon avant d'être distribués aux cibles de la session d'événements. À défaut 30 secondes.

**MAX\_EVENT\_SIZE** : taille maximale autorisée pour les événements.

**MEMORY\_PARTITION\_MODE** : emplacement où les mémoires tampons d'événements sont créées.

**TRACK\_CAUSALITY** : spécifie si la causalité est suivie ou non. Si cette option est activée, la causalité permet à des événements associés de différentes connexions au serveur d'être corrélés.

**STARTUP\_STATE** : spécifie si cette session d'événements doit être lancée automatiquement au démarrage de SQL Server.

## 6 / Les événements étendus

58

### Mise en œuvre

**<option\_de\_session>::=**

```
{ [ MAX_MEMORY = taille [ KB | MB ] ]  
  [ [ , ] EVENT_RETENTION_MODE = { ALLOW_SINGLE_EVENT_LOSS  
    | ALLOW_MULTIPLE_EVENT_LOSS  
    | NO_EVENT_LOSS } ]  
  
  [ [ , ] MAX_DISPATCH_LATENCY = { secondes SECONDS | INFINITE } ]  
  [ [ , ] MAX_EVENT_SIZE = taille [ KB | MB ] ]  
  [ [ , ] MEMORY_PARTITION_MODE = { NONE | PER_NODE | PER_CPU } ]  
  [ [ , ] TRACK_CAUSALITY = { ON | OFF } ]  
  [ [ , ] STARTUP_STATE = { ON | OFF } ] }
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

## 6 / Les événements étendus

59

### Mise en œuvre, filtrage des événements (Predicates) :

```
<prédicat> ::=  
{  
  [ NOT ] <prd_facteur> | { ( <prédicat> ) }  
  [ { AND | OR } [ NOT ] { <prd_facteur> | ( <prédicat>  
) } ]  
  [ ,...n ]  
}  
  
<prd_facteur> ::=  
{  
  <prédicat_feuille> | ( <prédicat> )  
}
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

NOTA : Le prédicat est peut être une structure arborescente (récursive). Seul le niveau feuille permet un filtrage effectif !



## 6 / Les événements étendus

60

### Mise en œuvre, filtrage des événements (Predicates) :

**<prédicat\_feuille> ::=**

```
{  
  <déclaration_prédicat>  
  { = | < | > | != | > | > = | < | < = } <valeur>  
  | [GUID_me].nom_pack.nom_comparateur  
  ( <déclaration_prédicat>, <valeur> )  
}
```

**<déclaration\_prédicat> ::=**

```
{  
  nom_champ_événement | ( [GUID_me].nom_pack.nom_source )  
}
```

## 6 / Les événements étendus

61

### Un exemple :

```
CREATE EVENT SESSION login_failure_audit ON SERVER
ADD EVENT sqlserver.error_reported
( ACTION (sqlserver.sql_text,
          sqlserver.client_app_name,
          sqlserver.client_hostname,
          sqlserver.database_id,
          sqlserver.session_nt_username)
  WHERE error = 4060
        OR error = 4064
        OR error = 15113...)
ADD TARGET package0.asynchronous_file_target
(SET FILENAME = 'C:\XE\login_failure_audit.xel',
  METADATAFILE = 'C:\XE\login_failure_audit.xem')
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

**Dans cet exemple, nous traquons les échecs d'authentification.**

Les n° d'erreur correspondent aux différentes erreurs possible de la table sys.messages.

Nous suivons toutes ces erreurs :

4060, 4064, 15113, 17197, 17892, 18056, 18311, 18315, 18321, 18322, 18323, 18324, 18325, 18326, 18327, 18331, 18332, 18333, 18339, 18343, 18345, 18346, 18347, 18348, 18350, 18357, 18401, 18451, 18452, 18456, 18458, 18459, 18460, 18461, 18462, 18463, 18464, 18465, 18466, 18467, 18468, 18470, 18471, 18486, 18487, 18488

Récupérées par :

```
SELECT DISTINCT message_id
FROM sys.messages
WHERE "text" LIKE '%fail%' AND "text" LIKE '%login%'
```

## 6 / Les événements étendus

62

**Dans cet exemple, nous traquons les échecs d'authentification.**

Pour choisir l'événement étendu à scruter, nous avons rechercher les événement mentionnant le terme « error »

```
SELECT name, description
FROM sys.dm_xe_objects
WHERE object_type = 'event'
      AND name LIKE '%error%';
```

Nous avons sélectionné « error\_reported »

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

| <b>Name</b>                | <b>description</b>         |
|----------------------------|----------------------------|
| error_ring_buffer_recorded | Error ring buffer recorded |
| errorlog_written           | Errorlog written           |
| error_reported             | Error has been reported    |
| cdc_error                  | CDC Error Information      |

## 6 / Les événements étendus

63

**Dans cet exemple, nous traquons les échecs d'authentification.**

Par la requête suivante :

```
SELECT name, type_name, column_type
FROM sys.dm_xe_object_columns
WHERE object_name = 'error_reported';
```

Nous obtenons les attributs exposés pour cet événement.  
Ils correspondent aux données habituelles des messages d'erreur plus les métadonnées de l'événement.

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

| Name         | type_name   | column_type |
|--------------|-------------|-------------|
| ID           | uint16      | readonly    |
| UUID         | guid_ptr    | readonly    |
| VERSION      | uint8       | readonly    |
| CHANNEL      | etw_channel | readonly    |
| KEYWORD      | keyword_map | readonly    |
| error        | int32       | data        |
| severity     | int32       | data        |
| state        | int32       | data        |
| user_defined | boolean     | data        |
| message      | unicode_    | string data |

## 6 / Les événements étendus

64

**Dans cet exemple, nous traquons les échecs d'authentification.**

Nous avons choisit d'enregistrer les actions suivantes :

|                               |                         |
|-------------------------------|-------------------------|
| sqlserver.sql_text            | requête SQL             |
| sqlserver.client_app_name     | nom application cliente |
| sqlserver.client_hostname     | nom hôte                |
| sqlserver.database_id         | id base contextuelle    |
| sqlserver.session_nt_username | compte NT               |

## 6 / Les événements étendus

65

**Dans cet exemple, nous traquons les échecs d'authentification.**

Et comme cible de l'enregistrement de ces événements un fichier en mode asynchrone situé à (FILENAME) :

- C:\XE\login\_faire\_audit.xel

Et dont la structure est décrite par (METADATAFILE) :

- C:\XE\login\_faire\_audit.xem

## 6 / Les événements étendus

66

**Dans cet exemple, nous traquons les échecs d'authentification.**

Il faut maintenant démarrer la session :

```
ALTER EVENT SESSION  
login_failure_audit  
ON SERVER  
STATE = START;  
GO
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Pour supprimer une session, utiliser DROP EVENT SESSION.

## 6 / Les événements étendus

67

**Dans cet exemple, nous traquons les échecs d'authentification.**

Après quelque temps, nous pouvons dépouiller les données. Pour cela il faut utiliser la fonction table :

□ `sys.fn_xe_file_target_read_file`

```
SELECT CAST(event_data AS xml) AS data
FROM sys.fn_xe_file_target_read_file
      ('C:\XE\login_failure_audit*.xel',
      'C:\XE\login_failure_audit*.xem',
      NULL,
      NULL);
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017



## 6 / Les événements étendus

68

### Métadonnées décrivant les objets XE

| Vue                      | Contenu  |
|--------------------------|--|
| sys.dm_xe_packages       | liste de tous les packages permettant l'utilisation des événements étendus           |
| sys.dm_xe_objects        | retourne tous les packages, événements, cibles, actions, prédicats, maps et types    |
| sys.dm_xe_object_columns | caractéristiques des packages, événements, cibles, actions, prédicats, maps et types |
| sys.dm_xe_map_values     | valeurs utilisées par SQL Server en interne par le moteur de bases de données        |

## 6 / Les événements étendus

69

### Métadonnées d'exploitation des sessions XE

| Vue                              | Description  |
|----------------------------------|--|
| sys.dm_xe_sessions               | caractéristiques des sessions d'événements étendus en cours d'exécution  |
| sys.dm_xe_session_events         | liste d'événements étendus, d'actions et de cibles qu'une session en cours d'exécution utilise                       |
| sys.dm_xe_session_event_actions  | liste des caractéristiques actions des événements capturés par une session d'événements étendus en cours d'exécution |
| sys.dm_xe_session_targets        | caractéristiques des cibles d'une session d'événements étendus en cours d'exécution                                  |
| sys.dm_xe_session_object_columns | liste des caractéristiques des événements capturés par une session en cours d'exécution                              |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Il existe d'autres vues de métadonnées des événements étendus :

**sys.server\_event\_sessions** Retourne la liste de toutes les sessions XE enregistrées dans l'instance, qu'elles soient démarrées ou non

**sys.server\_event\_session\_actions** Retourne tous les actions associées aux événements capturés par toutes les sessions enregistrées dans l'instance

**sys.server\_event\_session\_events** Retourne tous les événements capturés par une session enregistrée dans l'instance, avec éventuellement ses prédicats de filtrage

**sys.server\_event\_session\_fields** Retourne les caractéristiques d'une session enregistrée dans l'instance, comme les fichiers de la cible, ou la mémoire maximale configurée lors de la spécification de la session

**sys.server\_event\_session\_targets** Retourne la liste des cibles de toutes les sessions enregistrées dans l'instance

**sys.dm\_os\_dispatcher\_pools** retourne les caractéristiques du distributeur d'événements étendus pour les sessions en cours d'exécution.

## 6 / Les événements étendus

70

La session « **system\_health** » activée par défaut sur SQL Server collecte les événements dans un tampon mémoire circulaire (*ring buffer*). On peut en voir le contenu :

```
SELECT DISTINCT S.name AS session_name, SEAM.event_name,
A.action_list
FROM sys.dm_xe_sessions AS S
INNER JOIN sys.dm_xe_session_event_actions AS SEAM
ON S.address = SEAM.event_session_address
CROSS APPLY (SELECT PA.name + '.' + SEA.action_name + ', '
FROM sys.dm_xe_session_event_actions AS SEA
INNER JOIN sys.dm_xe_packages AS PA
ON SEA.action_package_guid = PA.guid
WHERE S.address = SEA.event_session_address
AND SEAM.event_name = SEA.event_name
FOR XML PATH ('')) AS A (action_list)
WHERE S.name = 'system_health'
ORDER BY SEAM.event_name;
```

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

## 6 / Les événements étendus

71

La liste des cibles de la session « *system\_health* » peut être obtenue par :

```
SELECT S.name AS session_name,  
       PT.name + '.' + ST.target_name AS target_name,  
       CAST(ST.target_data AS xml) AS target_data  
FROM   sys.dm_xe_session_targets AS ST  
       INNER JOIN sys.dm_xe_sessions AS S  
               ON ST.event_session_address = S.address  
       INNER JOIN sys.dm_xe_packages AS PT  
               ON ST.target_package_guid = PT.guid;
```

La colonne `target_data` retourne un document XML avec un contenu différent en fonction de la cible utilisée par la session concernée.

## 6 / Verrouillages et blocages

72

SQL Server utilise un mode de verrouillage pessimiste par défaut

- ▣ Niveau d'isolation READ COMMITTED
- ▣ Une écriture bloque la lecture et l'écriture d'autres processus
- ▣ Le verrouillage peut entraîner des
  - ▣ Attentes (wait)
  - ▣ Blocages
  - ▣ Verrous mortels (deadlocks)

## 6 / Verrouillages et blocages

73

- Il existe un mode interne de verrouillage plus léger
  - *Latches*
  - Utilisé pour verrouiller les pages d'index
- Les verrous assurent la consistance logique, les latches la consistance physique

Le verrouillage optimiste est assuré par le niveau d'isolation SNAPSHOT, de manière

- explicite (SET TRANSACTION ISOLATION LEVEL SCNAPSHOT)
- Ou implicite (réglage par base, remplace le READ COMMITTED)

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Pour assurer le verrouillage optimiste le DBA doit paramétrer la base :

ALTER DATABASE nom\_base SET

- ALLOW\_SNAPSHOT\_ISOLATION { ON | OFF } autorise l'utilisation explicite du niveau d'isolation SNAPSHOT
- READ\_COMMITTED\_SNAPSHOT { ON | OFF } remplace toute lecture consistante par une lecture SNAPSHOT

L'utilisation de ces deux modes a pour conséquence :

- De rajouter à chaque ligne de chaque table un « tag » de versionnement constitué de 14 octets dès qu'une modification a lieu
- De stocker les différentes versions en cours d'utilisation dans la base tempdb

## 6 / Verrouillages et blocages

74

### Types de verrous (granularité) :

| ID  | Description                        | Nature    |
|-----|------------------------------------|-----------|
| RID | Ligne (table en en heap – ROWID)   | Logique   |
| KEY | Ligne (clef, table en cluster)     | Logique   |
| PAG | Page                               | Logique   |
| HBT | Ensemble (Partition, table, index) | Logique   |
| TAB | Table                              | logique   |
| EXT | Extension                          | Physique  |
| FIL | Fichier                            | Physique  |
| AU  | Unité d'allocation                 | Physique  |
| MD  | Métadonnées                        | Technique |
| APP | Application                        | Externe   |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

HBT ou HOBT est l'acronyme de Heap Or Balanced Tree

Pour les retrouver :

```
select * from spt_values where type = 'LR'
```

## 6 / Verrouillages et blocages

75

Principaux mode de verrouillage (nature) :

| ID    | Description  |
|-------|--|
| S     | Shared (partagé)                                   |
| U     | Update (modification)                              |
| X     | eXclusive (mise à jour)                            |
| IS    | Intent Shared (partagé à un niveau inférieur)      |
| IU    | Intent Update (modification à un niveau inférieur) |
| IX    | Intent eXclusive (exclusif à un niveau inférieur)  |
| Sch-S | Schema Shared (partagé de métadonnées)             |
| Sch-M | Schema modify (modification de métadonnées)        |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

La liste n'est pas exhaustive. Pour les retrouver :  
`select * from spt_values where type = 'L'`



## 6 / Verrouillages et blocages

76

Cinématique :



**Platon**

```
UPDATE Verite
SET EstCertain = 0
WHERE Source =
'Reflexion'
```

**X - KEY**

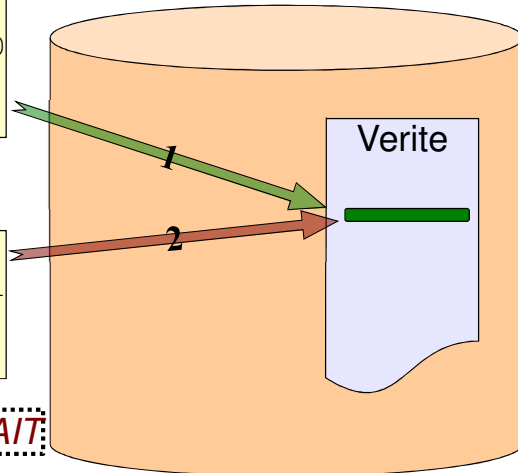


**Protagoras**

```
UPDATE Verite
SET EstCertain = 1
WHERE Source =
'Persuasion'
```



**S - KEY : WAIT**

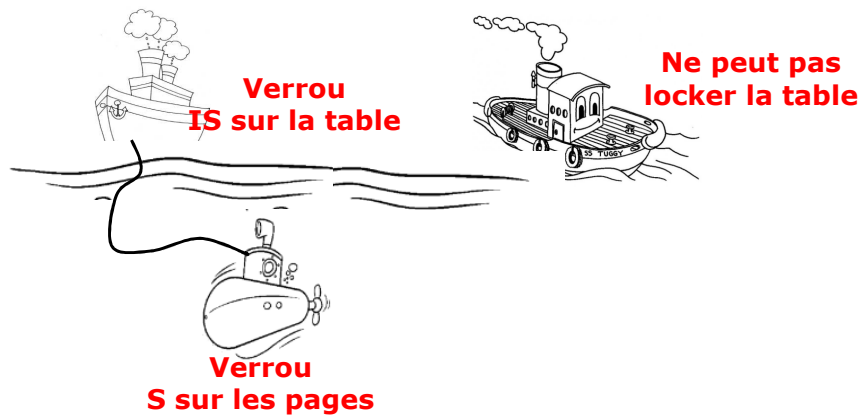


Platon a verrouillé en mode exclusif et Protagoras doit attendre la libération des verrous.

## 6 / Verrouillages et blocages

77

Les verrous « *intent* » servant à informer les niveau supérieurs...



Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Et par conséquent leur interdire de poser des verrous incompatibles plus « grossiers » !

## 6 / Verrouillages et blocages

78

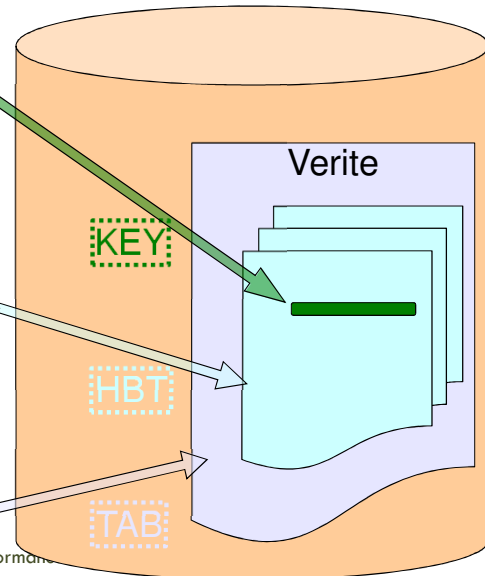
Escalade :



```
UPDATE Verite
SET EstCertain = 0
WHERE Source =
'Reflexion'
```

```
UPDATE Verite
SET EstCertain = 0
WHERE Source IN
('Reflexion',
'Persuasion')
```

```
UPDATE Verite
SET EstCertain = 0
```



Administration SQL Server - CH 06 - Surveillance & performance

SQL Server part du niveau PAGE ou ROW (RID ou KEY) pour monter au niveau PARTITION et ensuite TABLE

## 6 / Verrouillages et blocages

79

Matrice de  
compatibilité des  
verrous :

Ce mode est il  
possible ?

Mode  
posé

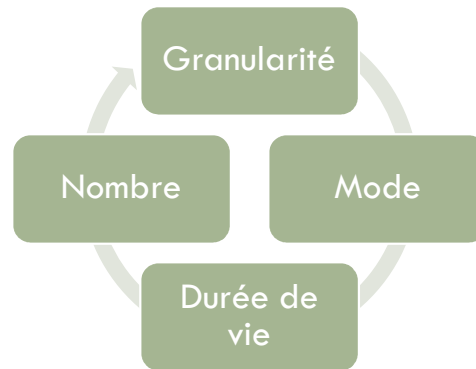
|    | S   | IS  | U   | X   | IX  |
|----|-----|-----|-----|-----|-----|
| S  | Oui | Oui | Oui | Non | Non |
| U  | Oui | Oui | Non | Non | Non |
| X  | Non | Non | Non | Non | Non |
| IS | Oui | Oui | Oui | Non | Oui |
| IX | Non | Oui | Non | Non | Oui |

## 6 / Verrouillages et blocages

80

4 points à considérer :

Plus la granularité est importante, plus le mode tend vers exclusif, plus la durée de vie du verrou augmente, et plus il y aura de la contention, donc de moins en moins de concurrence !



## 6 / Verrouillages et blocages

81

### Auditer le verrouillage...

- Verrous en cours :
  - DMV :  
sys.dm\_tran\_locks  
sys.dm\_os\_waiting\_tasks
- Attentes trop longues :  
« blocked process report »
- Statistiques de consommation de temps (pas uniquement verrous) :
  - sys.dm\_os\_wait\_stats  
À filtrer

**KILL pour tuer une session (force un ROLLBACK et déconnecte l'utilisateur).**

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Blocked Process Report se met en œuvre au niveau délai dans sp\_configure et nécessite d'activer une trace de profiler.

Sys.dm\_os\_wait\_stats donne toute les attentes, mais de nombreuses sont parfaitement normales. Éliminez les attentes suivantes :

N'BROKER\_EVENTHANDLER', N'BROKER\_RECEIVE\_WAITFOR', N'BROKER\_TASK\_STOP',  
N'BROKER\_TO\_FLUSH', N'BROKER\_TRANSMITTER', N'CHECKPOINT\_QUEUE', N'CHKPT',  
N'CLR\_AUTO\_EVENT',  
N'CLR\_MANUAL\_EVENT', N'CLR\_SEMAPHORE', N'DBMIRROR\_DBM\_EVENT',  
N'DBMIRROR\_EVENTS\_QUEUE', N'DBMIRROR\_WORKER\_QUEUE', N'DBMIRRORING\_CMD',  
N'DIRTY\_PAGE\_POLL',  
N'DISPATCHER\_QUEUE\_SEMAPHORE', N'EXECSYNC',  
N'FSAGENT', N'FT\_IFTS\_SCHEDULER\_IDLE\_WAIT', N'FT\_IFTSHC\_MUTEX',  
N'HADR\_CLUSAPI\_CALL', N'HADR\_FILESTREAM\_IOMGR\_IOCOMPLETION',  
N'HADR\_LOGCAPTURE\_WAIT', N'HADR\_NOTIFICATION\_DEQUEUE', N'HADR\_TIMER\_TASK',  
N'HADR\_WORK\_QUEUE', N'KSOURCE\_WAKEUP', N'LAZYWRITER\_SLEEP', N'LOGMGR\_QUEUE',  
N'MEMORY\_ALLOCATION\_EXT', N'ONDEMAND\_TASK\_QUEUE',  
N'PREEMPTIVE\_XE\_GETTARGETSTATE',  
N'PWAIT\_ALL\_COMPONENTS\_INITIALIZED', N'PWAIT\_DIRECTLOGCONSUMER\_GETNEXT',  
N'QDS\_PERSIST\_TASK\_MAIN\_LOOP\_SLEEP', N'QDS\_ASYNC\_QUEUE',  
N'QDS\_CLEANUP\_STALE\_QUERIES\_TASK\_MAIN\_LOOP\_SLEEP', N'QDS\_SHUTDOWN\_QUEUE',  
N'REDO\_THREAD\_PENDING\_WORK',  
N'REQUEST\_FOR\_DEADLOCK\_SEARCH', N'RESOURCE\_QUEUE', N'SERVER\_IDLE\_CHECK',  
N'SLEEP\_BPOOL\_FLUSH', N'SLEEP\_DBSTARTUP', N'SLEEP\_DCOMSTARTUP',  
N'SLEEP\_MASTERDBREADY',  
N'SLEEP\_MASTERMDREADY', N'SLEEP\_MASTERUPGRADED', N'SLEEP\_MSDBSTARTUP',  
N'SLEEP\_SYSTEMTASK', N'SLEEP\_TASK', N'SLEEP\_TEMPDBSTARTUP', N'SNI\_HTTP\_ACCEPT',  
N'SP\_SERVER\_DIAGNOSTICS\_SLEEP', N'SQLTRACE\_BUFFER\_FLUSH',  
N'SQLTRACE\_INCREMENTAL\_FLUSH\_SLEEP', N'SQLTRACE\_WAIT\_ENTRIES',  
N'WAIT\_FOR\_RESULTS', N'WAITFOR',  
N'WAITFOR\_TASKSHUTDOWN', N'WAIT\_XTP\_RECOVERY', N'WAIT\_XTP\_HOST\_WAIT',  
N'WAIT\_XTP\_OFFLINE\_CKPT\_NEW\_LOG', N'WAIT\_XTP\_CKPT\_CLOSE', N'XE\_DISPATCHER\_JOIN',  
N'XE\_DISPATCHER\_WAIT', N'XE\_TIMER\_EVENT'

## 6 / Verrouillages et blocages

82

Des processus attendent la libération de verrous ...

- ... attendent la libération de verrous ...
- ... placés par des requêtes trop longues
- SET LOCK TIMEOUT
  - Paramètre l'abandon d'une trop longue attente
  - Réglé à 30 secondes par défaut dans les applications
  - Infini dans SSMS

Il faut détecter les chaines de blocage...

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

La requête suivante permet de trouver les « lead blockers » dans les chaines de blocages :

```
WITH
T_LEADS AS
(SELECT session_id AS LEAD_BLOCKER
FROM sys_dm_exec_requests AS r
WHERE blocking_session_id = 0
AND EXISTS(SELECT *
FROM sys_dm_exec_requests AS ri
WHERE r.session_id = ri.blocking_session_id)),
T_TREE AS
(SELECT L.LEAD_BLOCKER, blocking_session_id, r.session_id, 0 AS prof
FROM sys_dm_exec_requests AS r
JOIN T_LEADS AS L ON r.session_id = L.LEAD_BLOCKER
UNION ALL
SELECT LEAD_BLOCKER, r.blocking_session_id, r.session_id, prof + 1
FROM sys_dm_exec_requests AS r
JOIN T_TREE AS T
ON r.blocking_session_id = T.session_id)
SELECT LEAD_BLOCKER, COUNT(*) AS BLOCKED_SESSIONS, MAX(prof) AS
BLOCKED_DEEP
FROM T_TREE
WHERE blocking_session_id > 0
GROUP BY LEAD_BLOCKER;
```

C'est à eux qu'il faut s'intéresser !

## 6 / Verrous mortels

83

Étreinte fatales, interblocages, *deadlocks*

- Deux transactions attendent pour la libération des mêmes verrous
- Situation sans issue
- SQL Server sait les reconnaître, et annule (*rollback*) une des transactions, et lui envoie un message d'erreur
  - Il choisit une victime
  - En général la moins coûteuse à annuler
  - Priorité modifiable sur la session par :  
SET DEADLOCK\_PRIORITY

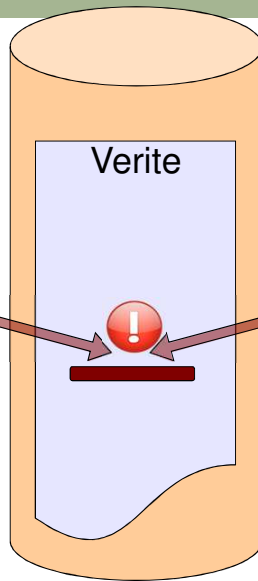


# 6 / Verrous mortels

84



```
BEGIN TRANSACTION  
  
UPDATE Verite  
SET EstCertain = 0  
WHERE Source =  
'Persuasion'  
  
SELECT * FROM  
Verite  
  
COMMIT TRANSACTION
```



```
BEGIN TRANSACTION  
  
SELECT * FROM  
Verite  
  
UPDATE Verite  
SET EstCertain = 1  
WHERE Source =  
'Persuasion'  
  
COMMIT TRANSACTION
```

## 6 / Verrous mortels

85

- Pourquoi cela arrive-t-il
  - Mauvaise structuration de la base
  - Mauvais développement
  - Manque d'index
- Comment l'éviter
  - ... bonne structuration de la base (normalisation)
  - ... développement correct
    - Notamment, implémenter la même séquence d'actions
  - Conserver des transactions courtes
  - Utiliser des niveaux d'isolations plus légers
  - Rajouter des index

## 6 / Verrous mortels

86

- Comment les détecter
  - Moniteur Système
  - Profileur
- Comment déterminer la source
  - Profileur
  - Drapeaux de trace
- Comment les gérer
  - Dans votre code, récupérez l'erreur et relancez l'instruction

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

### Profileur

Locks : Deadlock Chain

Locks : Deadlock Graph

### Drapeaux de trace

1204 : informations succinctes

1205 : informations détaillées

## 6 / Le collecteur de données

87

La collecte de données permet de connaître l'état de santé de votre instance à intervalles réguliers

- ❑ Il est bien d'avoir une vision quand le serveur est en bonne santé, pour pouvoir s'y référer plus tard
  - comprendre les écarts !
  - Par exemple, depuis le démarrage de cette application, la consommation de tempdb a considérablement augmenté...
- ❑ La collecte produit des rapports standard pour
  - L'activité du serveur
  - Le suivi de la volumétrie
  - La surveillance des requêtes principales
- ❑ Attention à la consommation de cette collecte
  - La base de données peut devenir très volumineuse !

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

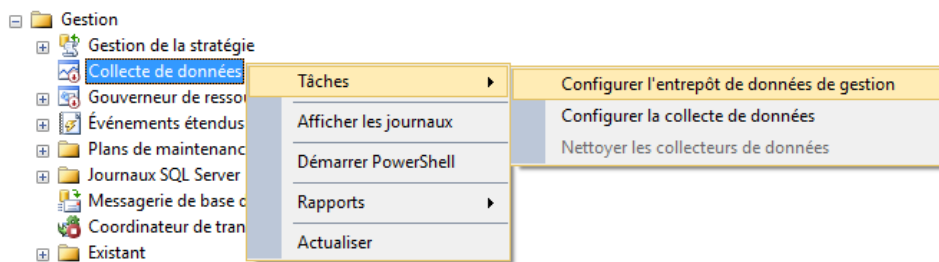
Pour les versions antérieures à la 2008, vous pouvez utiliser SQL H2 (SQL Health & History tool), téléchargeable sur le site de microsoft.  
Cependant, cet outil est bugué si vous l'installez sur un serveur en version française (problématique de format de dates)

## 6 / Le collecteur de données

88

Mise en œuvre...

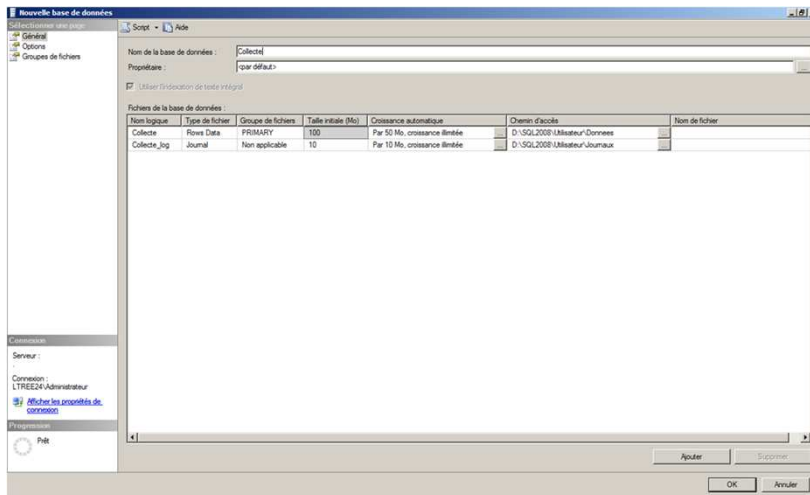
Dans l'arborescence serveur, allez à « Gestion / Collecte de données »



# 6 / Le collecteur de données

89

Il faut créer une base de données de collecte



Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

Une fois la base créée l'assistant règle la sécurité

# 6 / Le collecteur de données

90

Puis y ajouter les différents collecteurs

Assistant Configuration de la collecte de données

**Installer les jeux d'éléments de collecte de données**  
Choisissez un ensemble de collecteurs de données à créer et démarrer.

Sélectionnez un serveur et une base de données représentant l'hôte de votre entrepôt de données de gestion.

Nom du serveur : HPZ840FB\SQL2012FBIN2 ...

Nom de la base de données : DB\_COLLECTOR

Entrez l'emplacement du cache local où vous voulez stocker les données collectées avant leur téléchargement vers l'entrepôt de données de gestion. Une valeur vide utilise le répertoire TEMP du processus du collecteur.

Répertoire du cache : C:\ ...

Sélectionnez les ensembles de collecteurs de données que vous souhaitez activer :

| Nom  | Description  |
|--|--|
| <input checked="" type="checkbox"/> Jeux d'éléments de collecte de donn...   | Collecter les statistiques de performances pour le dépannage général.    |
| <input checked="" type="checkbox"/> Jeux d'éléments de collecte de perfor... | Collecter les statistiques des problèmes de performances de transacti... |

Utilisez un proxy SQL Server Agent pour les téléchargements distants.

Aide < Précédent Suivant > Terminer >> Annuler

Administration SQL Server - CH 06 - Surveillance & performances

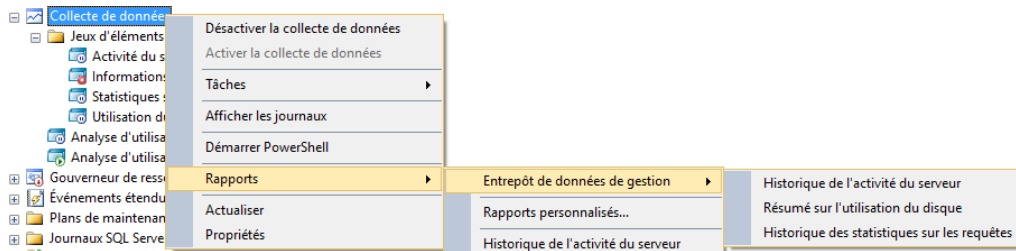
V3.0 10/07/2017

Certains nécessitent des répertoire pour stocker temporairement des résultats

## 6 / Le collecteur de données

91

Différents rapports sont disponibles





























Ils sont cliquables par « drill down »



## 6 / Le collecteur de données

92

Dans notre exemple, il y a un problème sur tempdb :

| Nom de la base de données                    | Base de données      |   |                      | Journal                      |                      |  |                      |                              |
|--|----------------------|---|----------------------|------------------------------|----------------------|--|----------------------|------------------------------|
|  | Taille de début (Mo) | Tendance  | Taille actuelle (Mo) | Croissance moyenne (Mo/jour) | Taille de début (Mo) | Tendance   | Taille actuelle (Mo) | Croissance moyenne (Mo/jour) |
| <a href="#">AnalyseVentes</a>                | 116.50               |  | 116.50               | 0                            | 34.00                |  | 34.00                | 0                            |
| <a href="#">Collecte</a>                     | 100.00               |  | 100.00               | 0                            | 10.00                |  | 10.00                | 0                            |
| <a href="#">DemoGeoSpatialR2</a>             | 2.25                 |  | 2.25                 | 0                            | 0.56                 |  | 0.56                 | 0                            |
| <a href="#">DemoSQS</a>                      | 300.00               |  | 300.00               | 0                            | 50.00                |  | 50.00                | 0                            |
| <a href="#">Digitime2008</a>                 | 250.00               |  | 250.00               | 0                            | 129.75               |  | 129.75               | 0                            |
| <a href="#">master</a>                       | 4.00                 |  | 4.00                 | 0                            | 1.25                 |  | 1.25                 | 0                            |
| <a href="#">model</a>                        | 2.25                 |  | 2.25                 | 0                            | 0.75                 |  | 0.75                 | 0                            |
| <a href="#">msdb</a>                         | 14.75                |  | 16.25                | 1.5                          | 3.06                 |  | 3.06                 | 0                            |
| <a href="#">Performances</a>                 | 100.00               |  | 100.00               | 0                            | 10.00                |  | 10.00                | 0                            |
| <a href="#">ReportServerSQL2008 R2</a>       | 4.25                 |  | 4.25                 | 0                            | 6.25                 |  | 6.25                 | 0                            |
| <a href="#">ReportServerSQL2008 R2TempDB</a> | 2.25                 |  | 2.25                 | 0                            | 0.81                 |  | 0.81                 | 0                            |
| <a href="#">SuiviPerformances</a>            | 6,250.00             |  | 6,250.00             | 0                            | 760.00               |  | 760.00               | 0                            |
| <a href="#">tempdb</a>                       | 8.00                 |  | 208.56               | 200.563                      | 1.00                 |  | 5.06                 | 4.063                        |

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

# 6 / Le collecteur de données

93

Investiguer pour trouver la cause...

## Historique de l'activité du serveur

Le LTREE24\SQL2008R2 à 14/06/2010 15:38:45

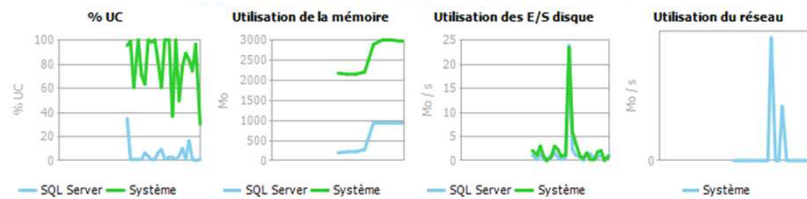


Ce rapport fournit une vue d'ensemble de la consommation des ressources et de l'activité du serveur pour l'instance SQL Server et pour le système d'exploitation hôte.

Naviguez à travers les instantanés des données à l'aide de la chronologie ci-après.



Plage temporelle sélectionnée : 14/06/2010 15:23:29 à 14/06/2010 15:38:29



# 6 / Le collecteur de données

94

## La coupable :

### Requête

[Modifier le texte de la requête :](#) `select formatted_value from snapshots.performance_counter_values order by raw_value_second,raw_value_first,formatted_value`

### Statistiques sur l'exécution des requêtes

|   |                                       |                                      |                                   |
|---|---------------------------------------|--------------------------------------|-----------------------------------|
| Temps processeur moyen (ms) par exécution : | <input type="text" value="2 465,0"/>  | Temps processeur total (s) :         | <input type="text" value="2,5"/>  |
| Durée moyenne (ms) par exécution :          | <input type="text" value="36 683,0"/> | Durée totale (s) :                   | <input type="text" value="36,7"/> |
| Lectures physiques moyennes par exécution : | <input type="text" value="3293"/>     | Nombre total de lectures physiques : | <input type="text" value="3293"/> |
| Écritures logiques moyennes par exécution : | <input type="text" value="7"/>        | Nombre total d'écritures logiques :  | <input type="text" value="7"/>    |
| Exécutions moyennes par minute :            | <input type="text" value="0"/>        | Nombre total d'exécutions :          | <input type="text" value="0"/>    |
| Temps processeur moyen (ms) par seconde :   | <input type="text" value="2"/>        | Nombre de plans de requête :         | <input type="text" value="1"/>    |
| Durée moyenne (ms) par seconde :            | <input type="text" value="40"/>       |                                      |                                   |
| Lectures physiques moyennes par seconde :   | <input type="text" value="3"/>        |                                      |                                   |
| Écritures logiques moyennes par seconde :   | <input type="text" value="0"/>        |                                      |                                   |

[Afficher les attentes échantillonnées pour cette requête](#)

### Principaux plans de requête par temps processeur moyen par exécution

| Plan num éro | Nombre d'exécution s | Heure de création du plan | Coût moyen par exécution |            |                    |                    | Taux d'utilisation des ressources |           |                        |                        |
|--------------|----------------------|---------------------------|--------------------------|------------|--------------------|--------------------|-----------------------------------|-----------|------------------------|------------------------|
|              |                      |                           | UC (ms)                  | Durée (ms) | Lectures physiques | Écritures logiques | Exécutions / min                  | UC ms / s | Lectures physiques / s | Écritures logiques / s |
| 1            | 0                    | 14/06/2010 15:33:55       | 2 465,0                  | 36 683,0   | 3293               | 7                  | 0                                 | 2         | 3                      | 0                      |

## 6 / Le gouverneur de ressources

95

- Gère des listes de ressources (*resource pool*) et des groupes de charge (*workload groups*).
- **Workload Group**
  - Container de ressources SQL (p.ex. Requêtes, max degree of Parallelism, ...)
- **Resource Pool**
  - Listes de ressources système (RAM, CPU, ...)
- **Classification**
  - Règles pour rediriger une demande vers un workload group.

## 6 / Le gouverneur de ressources

96

- Créer des « resource pools »
- Créer des « workload groups » utilisant les « resource pools »
- Ecrire une fonction de classification
- Attribuer la fonction de classification au gouverneur de ressources

### Reconfigurer le gouverneur de ressources

- Activation
  - ALTER RESOURCE GOVERNOR RECONFIGURE
- Désactivation
  - ALTER RESOURCE GOVERNOR DISABLE

Administration SQL Server - CH 06 - Surveillance & performances

V3.0 10/07/2017

### Exemple :

```
CREATE RESOURCE POOL poolAdhoc
WITH (MAX_CPU_PERCENT = 50)

CREATE WORKLOAD GROUP groupAdmin
USING poolAdhoc;

CREATE FUNCTION rgclassifier() RETURNS SYSNAME
WITH SCHEMABINDING
AS BEGIN
    DECLARE @grp_name AS SYSNAME
    IF (SUSER_NAME() = 'sa') SET @grp_name = 'groupAdmin'
    RETURN @grp_name
END
GO
ALTER RESOURCE GOVERNOR WITH (CLASSIFIER_FUNCTION=
dbo.rgclassifier)

ALTER RESOURCE GOVERNOR RECONFIGURE
```



# SQL SERVER 2012/2017 ADMINISTRATION

Frédéric Brouard (SQLpro) [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)  
Richard Gaillard - [richard.gaillard@horapax.com](mailto:richard.gaillard@horapax.com)

Version 3.000  
Date 10/07/2016



X Introduction à la haute disponibilité

Administration SQL Server - CH XX - Introduction à la haute disponibilité V3.0 10/07/2017

Ce support de cours a été réalisé par :

- Richard Gaillard - richard.gaillard@horapax.com – 06 17 60 00 05
- Frédéric Brouard (SQLpro) SQLpro@SQLspot.com – Sté SQL SPOT (Paris / PACA) – 06 11 86 40 66

# X – Intro. à la haute disponibilité

3



- ❑ Principes
- ❑ Sauvegarde / Restauration
- ❑ Log Shipping
- ❑ Database Mirroring
- ❑ AlwaysOn



## X / Principe

4

**But** : permettre aux applications de continuer d'utiliser les services de base e données, lorsqu'une défaillance survient

**Moyen** : disposer d'un « clone » ayant les mêmes données et pouvoir basculer sur ce clone en cas de défaillance du serveur principal

**Mécanisme** : se baser sur les ordres SQL, les transactions ou les données binaires modifiés sur le principal pour alimenter un serveur secondaire

## X / Principe

5

### Systemes :

- Réplication des données : réplication de table par snapshot
- Réplication des transactions : réplication transactionnelle de données, log shipping
- Réplication binaire : mirroring, AlwaysOn
- Copie de fichiers : sauvegarde, restauration
- Copie de VM : snapshot de VM

La réplication binaire des données prend sa source dans le journal de transaction

# X / Principe

6

## Systemes :

|                              | Synchrone | Reprise auto | Latence         |
|------------------------------|-----------|--------------|-----------------|
| Réplication de données       | NON       | NON          | Très importante |
| Réplication transactionnelle | NON       | NON          | Importante      |
| Réplication binaire          | OUI       | Possible     | Faible à nulle  |
| Copie de fichier             | NON       | NON          | Très importante |
| Copie de VM                  | NON       | NON          | Très importante |
| Copie IO (bas niveau)        | NON       | NON          | Très faible     |
| Cluster de basculement       | OUI       | OUI          | Faible à nulle  |

Seuls les systèmes intégrés à SQL Server assurent une reprise automatique et sans perte du service

Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

VM : Virtual Machine

Réplication binaire (Mirroring, AlwaysOn) et cluster de basculement (Windows Clustering Services) permettent un mode synchrone sans perte de données.

Microsoft n'offre pas de support sur les VM et en cas d'appel à la hot line, Microsoft peut vous demander de reproduire le bogue sur une machine physique.

Notre expérience montre que les VM Hyper V sont nettement plus fiable que celle par Wmware/Vsphère

**NOTA** : la réplication binaire se basant sur les journaux de transaction, ces derniers ne se purgent plus si les serveurs de secours sont en retard ou indisponible.

### ATTENTION :

la copie d'IO à bas niveau ne permet pas de garantir l'intégrité des bases (désynchronisation probable des LNS entre fichier de données et journaux de transaction)

La copie de VM, pour palier à cet inconvénient doit impérativement passer par le service VSS qui gèle les activités IO des bases (ce qui les rend indisponibles)

## X / Réplication de données

7

Réplication de données :

- Mode snapshot : agit table par table en copiant toutes les données d'une table dans un format spécifique
- Mode transactionnel : intercepte le code SQL pour l'envoyer au destinataire

Très couteux en ressources !

À limiter sur quelques tables...

Plutôt destiné à la répartition de charge

Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Le mode snapshot ne peut garantir l'intégrité de la base du fait que certaines tables peuvent être copiées avant d'autres qui leur serait nécessaire pour garantir l'intégrité référentielle.

Le mode transactionnel peut garantir l'intégrité de la base sans certains mécanismes particuliers. Pour ce faire SQL Server modifie le paramétrage des tables pour éviter le relancement de certaines routines comme l'obtention d'un GUID, d'un auto incrément ou la date/heure courante... (option NOT FOR REPLICATION... dans la création des tables)

Avantage : la structure des base peut être différente

## X / Réplifications binaires

8

### Réplication de pages :

- Consiste à renvoyer le binaire des lignes des pages mises à jour au destinataire
  - Léger en ressources
  - Fonctionne base par base
- La structure des bases doit être strictement identique.

### Idéal pour la haute disponibilité, car :

- synchrone possible
- basculement automatique possible

Le Log Shipping, le mirroring ou AlwaysOn utilise cette technologie

## X / Copie

9

### Copie :

- De fichiers : nécessite l'arrêt de la base
- De VM : fige la base pendant l'opération de copie, pas de garantie apportée par Microsoft
- D'IO : complexe et dangereux si latence et cache en écriture

Peu intéressant pour gérer la haute disponibilité

L'arrêt de la base vide le cache.

Le temps que la VM effectue son cliché, les bases deviennent indisponible pour les écritures et les transactions sont suspendues.

## X / Cluster physique

10

Cluster de basculement :

- Deux machines se partagent une même baie de disques
- Sur l'une, l'instance est en production, sur l'autre, l'instance est à l'arrêt.
- En cas de panne de l'instance de production, le service Windows de cluster démarre l'instance à l'arrêt qui reprend les fichiers « communs »
- SPOF : la baie de disque partagée

Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

SPOF : Single Point Of Failure (maillon faible de la chaîne !)

Le cluster de basculement est une vieille technologie qui n'est plus réellement d'actualité avec AlwaysOn qui marie Mirroring et Cluster de basculement dans un principe plus élégant, plus rapide et moins contraignant (plus de SPOF)

## X / Log shipping

11

Consiste à envoyer à la base de secours les portions successives du journal de transaction pour réimplanter le binaire des lignes modifiées

- Mise en place simple
- Solution robuste et fiable
- Permet de multiples secours
- Pas de lecture possible sur les secours

Mise en place possible dans toutes les éditions (y compris Express)

Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Inconvénients :

- les fichiers véhiculés disposent des informations en clair sauf si mise en place de TDE (Transparent Data Encryption qui crypte le stockage).
- asynchrone, latence quelques minutes
- pas de basculement automatique

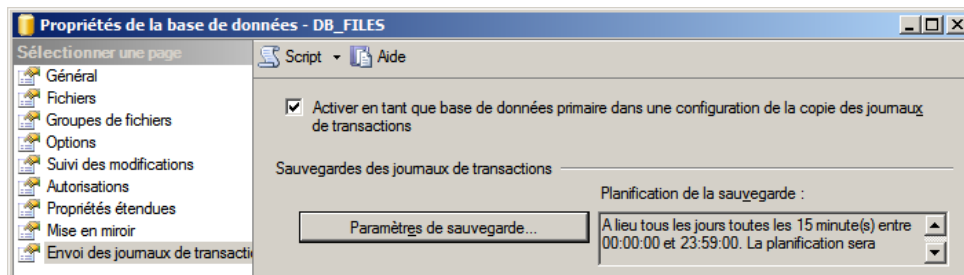


# X / Log shipping

12

## Assistant de mise en place

- Dans les propriétés de la base
- Onglet « Envoi des journaux de transactions »



Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Laissez vous guider !

Si le serveur de destination ne produit pas pour d'autres bases, alors, aucune licence n'est à payer.

## X / Mirroring

13

Consiste à envoyer les portions de pages modifiées dans le journal d'un serveur à l'autre par le biais du protocole http.

- Mise en place relativement simple
- Solution robuste et relativement fiable
- Ne permet qu'un seul secours
- Pas de lecture possible sur le secours
- Les envois sont chiffrés/authentifiés
- « deprecated » depuis la version 2012 !

Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Il est possible de se passer du chiffrement (authentification) si les serveurs sont dans le même domaine.

Bien que la solution soit considérée come « deprecated » depuis la version 2012, c'est le seul moyen de faire une haute disponibilité synchrone si vous possédez l'édition standard.

Cette solution ne pose pas de problèmes majeurs !

## X / Mirroring

14

3 modes :

- Haute résilience : synchrone à basculement automatique
- Haute performance : asynchrone à basculement manuel
- Haute protection : synchrone à basculement manuel

Le basculement automatique nécessite un 3<sup>e</sup> partenaire pour assurer le quorum

Le mode synchrone tire les performances vers le bas

Le mode asynchrone nécessite plus de ressources

Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Le partenaire assurant le quorum peut être une édition Express de SQL Server et ne doit pas être sur le même serveur physique et si possible ne pas passer par les mêmes routes physiques (séparation des flux réseau).

La version asynchrone n'est disponible que dans l'édition Enterprise.

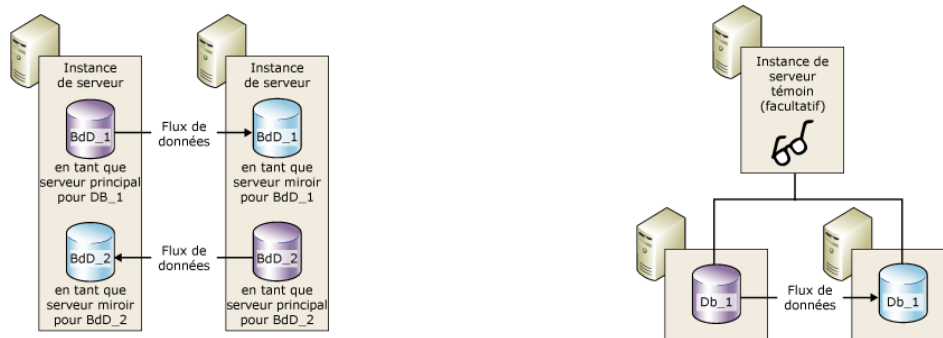
Le mirroring s'effectuant base par base, vous pouvez croiser les flux entre serveurs

# X / Mirroring

15

À gauche un mirroring croisé.

À droite un mirroring synchrone à basculement automatique. Notez la présence du tiers



Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Les bases ne peuvent être lues, mais on peut effectuer un snapshot de base de données sur le miroir afin d'obtenir une base en lecture seule ayant les données à une heure précise (celle du moment ou a eut lieu la création du snapshot)

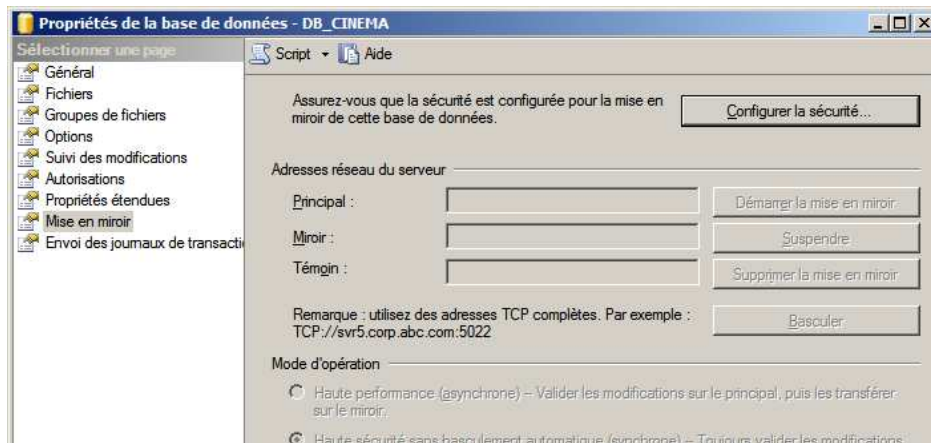
Si le serveur de destination ne produit pas pour d'autres bases, alors, aucune licence n'est à payer. Vous pouvez néanmoins tester la solution un maximum de 30 jours, chaque instant d'un jour faisant perdre le jour entier.

Le système fonctionnant sur la couche http, vous pouvez faire des miroirs de vos bases à grande distance. Dans ce cas, préférez le mode asynchrone, sinon les performances vont chuter rapidement !

# X / Mirroring, mise en œuvre

16

Utilisez l'assistant !



Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Dans les propriétés de la base, onglet « Mise en miroir »

## X / AlwaysOn

17

Arrivée avec la version 2012 (et Windows 2008 R2)

- Nécessite la mise en place du service WSFC (Windows Server Failover Clustering) au niveau de l'OS
- Fonctionne par groupe de disponibilité (ensemble de bases de données à « répliquer »)
- Permet plusieurs destinations :
  - 4 en v. 2012 dont 1 en synchrone
  - 8 en v. 2014 dont 2 en synchrones

La couche WSFC nécessite un quorum (ressource partagée)

Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Le système AlwaysOn résous un problème récurrent (qui ne devrait d'ailleurs pas exister), celui d'une application éclatée en de multiples bases. En effet, le groupe de disponibilité permet de basculer un ensemble de base lorsqu'une seule d'entre elles est défaillante.

Les destinations synchrones sont appelées « replicas primaires » et peuvent basculer automatiquement.

Les destinations asynchrones sont appelées « replicas secondaire » et ne peuvent que basculer manuellement.

La couche WSFC assure les service de surveillance et de basculement. Pour ce dernier cas, si le basculement automatique des nœuds synchrones est paramétré.

Le quorum peut être un disque ou un fichier et doit se situer sur une serveur tiers.

Une même base ne peut pas participer à plusieurs groupes de disponibilité.

Groupe de disponibilité, en anglais « availability group »

## X / AlwaysOn

18

Toutes les destinations sont lisibles :

- Vous pouvez effectuer des sauvegardes sur les réplicas (évitez les sauvegardes des journaux qui sont très actifs sur les réplicas)
- Pour la lecture des réplicas, des « listeners » doivent être mis en place

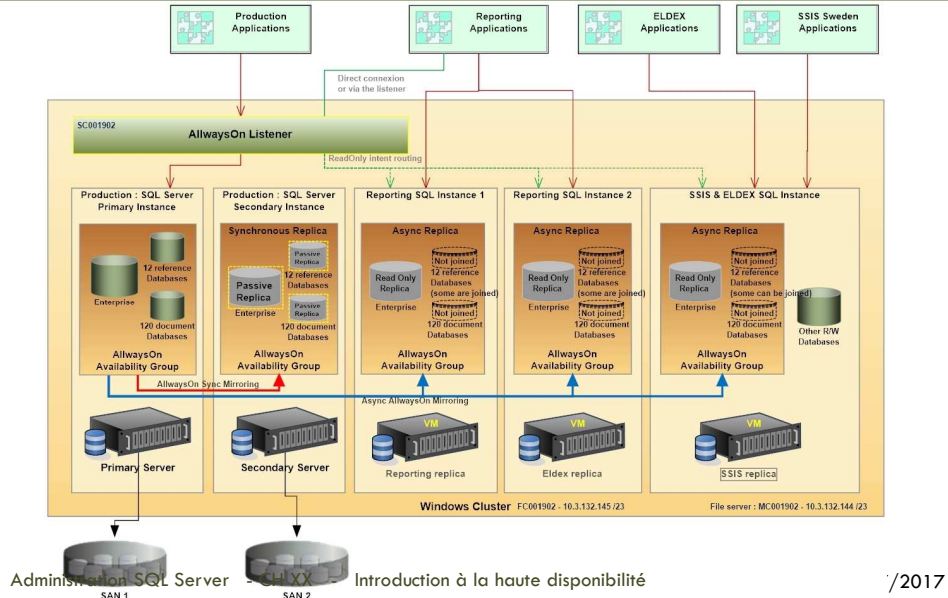
Pour des raisons de performance utilisez le mode asynchrone pour les réplicas à longue distance

Listener : service d'écoute

Si les serveurs de destination ne produisent pas pour d'autres bases ou ne permettent pas la lecture des bases répliquées, alors, aucune licence n'est à payer. Vous pouvez néanmoins tester la solution un maximum de 30 jours, chaque instant d'un jour faisant perdre le jour entier.

# X / AlwaysOn, exemple d'archi.

19



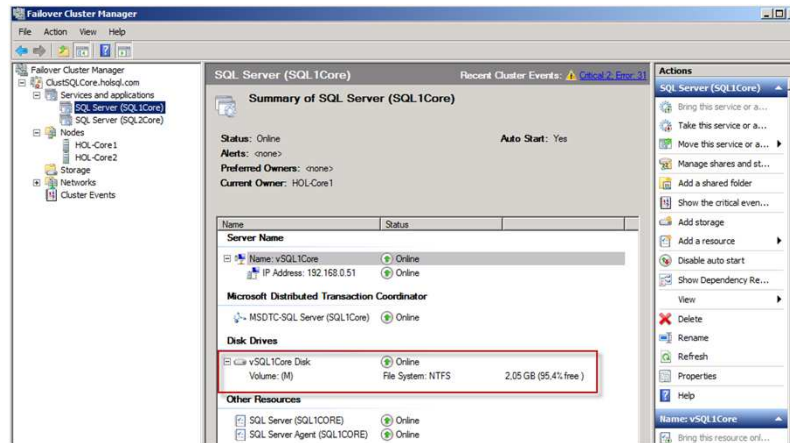
Cette architecture a été proposée pour l'application CargoWise de Geodis (16 To plus de 140 bases) pour sécuriser les données mondiales de routage logistique



# X / AlwaysOn, mise en œuvre

20

## Installer WSFC avec l'assistants Windows !



Administration SQL Server - CH XX - Introduction à la haute disponibilité

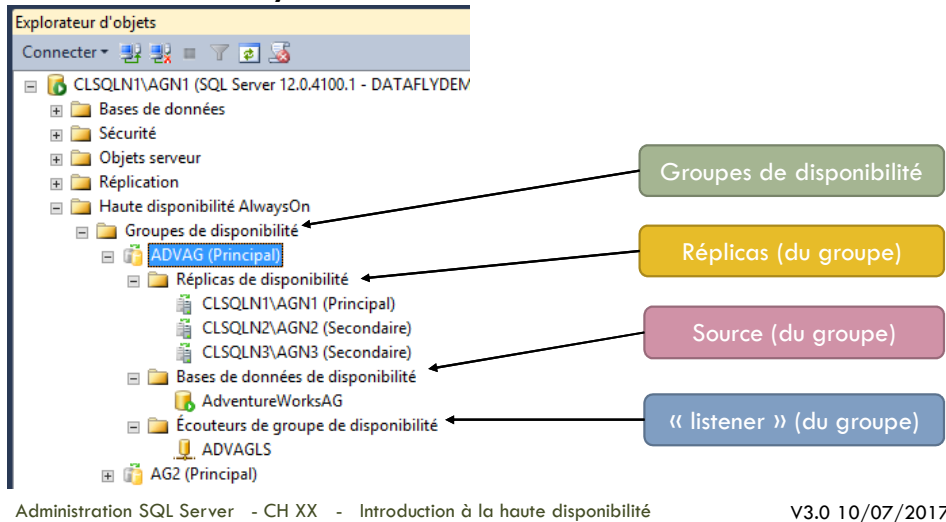
V3.0 10/07/2017

Il faut aussi autoriser le service SQL Server à utiliser AlwaysOn; pour cela allez dans le gestionnaire de configuration de SQL Server et dans le service SQL Server considéré, cliquez droit, puis sélectionnez propriété et allez dans l'onglet « haute disponibilité AlwaysOn »

# X / AlwaysOn, mise en œuvre

21

## Installer AlwaysOn avec les assistants SQL Server !



Commencez par créer le groupe de disponibilité, puis

- placez-y les bases que vous voulez regrouper en basculement
- choisissez les serveurs de destinations (réplicas)
- définissez le mode de synchronisation initial
- validez le résumé de l'assistant

Réplicas possible vers Azure en mode asynchrone.

# X / AlwaysOn, mise en œuvre

22

## Définir le nom du groupe de disponibilité

Nouveau groupe de disponibilité

**Spécifier le nom du groupe de disponibilité**

Introduction [Aide](#)

**Spécifier le nom**

Sélectionner les bases de données

Spécifier les répliques

Spécifiez le nom d'un groupe de disponibilité.

Nom du groupe de disponibilité :

ADVAG

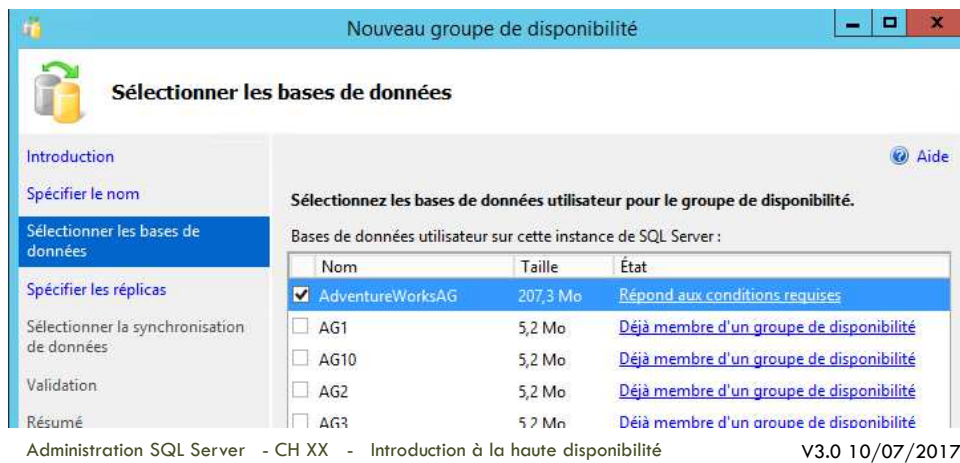
Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

# X / AlwaysOn, mise en œuvre

23

Définir les bases qui y participent :



Une même base ne peut pas participer à plusieurs groupes de disponibilité

# X / AlwaysOn, mise en œuvre

24

## Spécifier les réplicas (serveurs distants) :

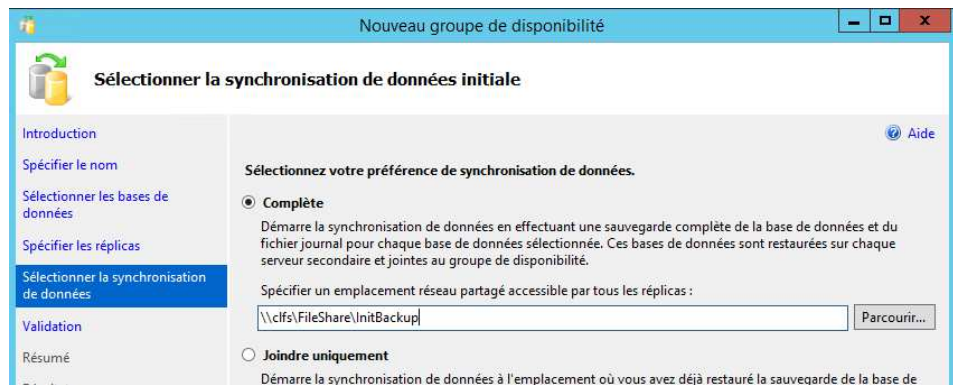
Administration SQL Server - CH XX - Introduction à la haute disponibilité V3.0 10/07/2017

Les serveurs distants doivent être dans le même domaine

# X / AlwaysOn, mise en œuvre

25

## Effectuer la synchronisation initiale :



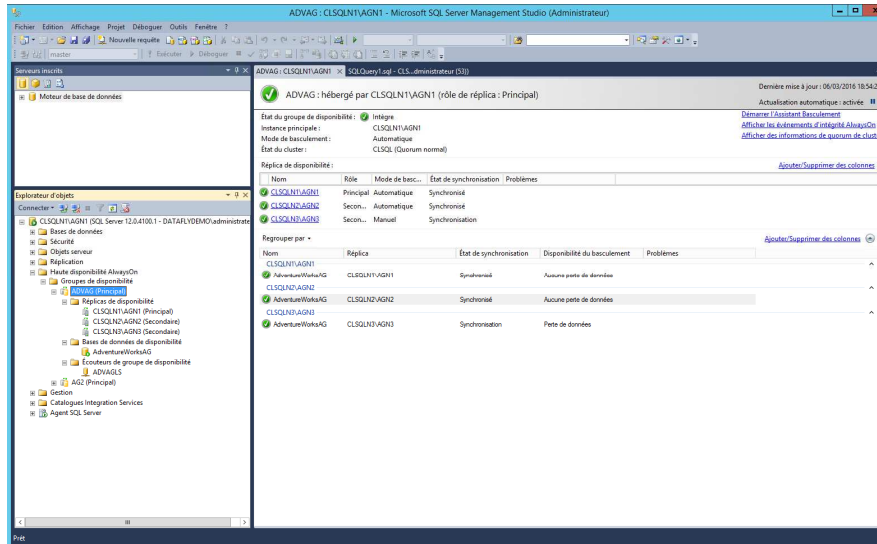
Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Ceci effectue une copie par sauvegarde de la base originale

# X / AlwaysOn, monitorer

26



Administration SQL Server - CH XX - Introduction à la haute disponibilité

V3.0 10/07/2017

Des outils comme Sentry Performance Advisor permettent de monitorer plus finement encore le fonctionnement de AlwaysOn