

## Planning intervention SQL du 2 et 3 octobre 2017

### Lundi matin

- Sujet** ● Vue dématérialisée et complexe devenant un point de contention sur une base de production
- Contexte** Une vue obèse utilisée pour plusieurs besoins de reporting commence à poser des problèmes de performance  
Le nombre de champs très importants grossit régulièrement et se rapprochent de la limite des 1 024 colonnes
- Objectif** Déterminer si le recours à certaines alternatives SQL (voir liste en annexe) pourrait désengorger ce point de contention

### Lundi après-midi

- Sujet** ● Fludification du transfert de données entre environnement de production et de reporting
- Contexte** Les besoins en reporting sont traités avec une solution ETL assez complexe malgré des besoins très simples
- Objectif** Analyser si certaines des fonctionnalités avancées de SQL Server (voir liste en annexe) seraient appropriées pour suppléer certaines des tâches ETL
- Sujet** ● Gestion de la transposition matricielle des données à catégorisation fluctuante
- Contexte** Certaines données sont classifiées dans des catégories dont le nombre d'occurrences augmente régulièrement  
A des fins de reporting, ces catégories sont affichées sous forme de classification verticale, ce qui augmente le nombre de colonnes dans les vues
- Objectif** Réfléchir à une solution qui permettrait de gérer cette catégorisation en mode ligne de façon dynamique (ne sont affichés que les catégories utilisées pour chaque occurrence)

### Mardi matin

- Sujet** ● Gestion des verrous de lecture sur les vues d'une base de données de production
- Une base de données de traçabilité communiquant avec plusieurs outils ETL est parfois bloquée en raison de problèmes de verrous  
Trouver la procédure à suivre pour s'affranchir de ce blocage
- Sujet** ● Tâche de maintenance pour reconstruire régulièrement les index des bases de production
- Contexte** La tâche de maintenance est régulièrement bloquée en raison de verrous posés sur les données
- Objectif** Déterminer la procédure à suivre pour contourner ce problème  
Elaborer le script permettant d'obtenir la liste des carences d'index

### Mardi après-midi

- Sujet** ● Base PostgreSQL ayant des temps de réponse et de traitement des requêtes anormalement excessifs
- Contexte** Pour des besoins de traçabilité, une requête est régulièrement lancée sur une base de données PostgreSQL hébergée sur un équipement  
Celle-ci met en moyenne 8 minutes pour traiter une requête simple (petite jointure entre 2 tables) retournant 1200 lignes
- Objectif** Trouver des axes d'investigation pour savoir si il s'agit d'un problème purement SQL ou plutôt système / hardware
- Sujet** ● Planification de tâches de maintenance diverses qui tombent en erreur
- Contexte** La planification des tâches de maintenance exécutées sur les bases de production ne fonctionnent plus depuis le départ du DBA
- Objectif** Déceler le correctif à apporter afin de permettre à nouveau le bon fonctionnement de ces maintenances
- Sujet** ● Délégation d'authentification dans le cadre du transfert de données entre 2 serveurs SQL distincts
- Contexte** Le transfert des jetons Kerberos entre les 2 serveurs ne fonctionnent pas et une solution de bridge intermédiaire avec compte de service spécifique a dû être mis en place
- Objectif** Identifier l'origine du dysfonctionnement de la délégation d'IWA

## Questions : Vue dématérialisée et complexe devenant un point de contention sur une base de production

### Vues indexées

- Comment fonctionne la mise à jour des vues indexées ? Sont-elles synchrones ou asynchrones ?
- Si la mise à jour est synchrone, quel est l'impact sur les tables lors de mises à jour importantes ?
- L'option "Query rewrite" peut-elle apporter un gain de performances ?
- Comment vérifier que la vue indexée est bien déterministe ?
- Existe-t-il des risques d'arrondis sur les champs à précision flottante (Real / Float) ?
- Les vues indexées permettent-elles de référencer d'autres vues dématérialisées ?
- Peut-on faire appel à des données stockées sur d'autres bases voire d'autres serveurs SQL ?
- Est-il possible de morceller une vue dématérialisée complexe en plusieurs vues indexées plus simples ?
- L'opérateur Cross Apply est-il compatible avec la vue indexée ?
- Est-il possible d'utiliser tous les agrégats de calcul avec une vue indexée ?

### Vues matérialisées

- Le concept de vues matérialisées est-il possible avec SQL Server ?
- Si ce système de vue existe bien, utilise-t-il une mise à jour asynchrone permettant de ne pas avoir à recalculer toutes les données mais juste un différentiel de la vue (données qui ont été modifiées entre le dernier rafraichissement et la lecture) ?
- A partir de quel volume de données, peut-il être pertinent d'avoir recours à des vues partitionnées ?
- Pour gérer les données éclatées sur plusieurs serveurs SQL, peut-on utiliser des vues distribuées ?

### Autres

- L'utilisation de la procédure Sp\_refreshview combinée avec des Select \* est-elle recommandée ?
- Si oui comment automatiser son implémentation ?

## Sujet : Fludification du transfert de données entre environnement de production et de reporting

### Fonctionnalités avancées SQL Server appliquées au reporting

- Pour la partie Back-end d'un reporting, quelle plus value peuvent représenter les tables en colonne et les Columnstores qui utilisent des indexations à structure verticale ?
- Le recours à des tables en mémoire / In memory avec le moteur Hekaton peut-il offrir des gains de performance sur les requêtes de reporting (lecture seule) ?
- Peut-il être pertinent d'utiliser des assemblages DotNet pour les opérations de calculs matriciels de type transposition de données pour améliorer les performances des opérateurs Pivot / Unpivot ?
- Les snapshots et instantanés de bases de données sont-ils envisageables pour éviter les connexions directes sur les bases de données de production induits par des besoins de reporting de type pseudo temps réel ?
- L'agent SQL peut-il être une alternative aux systèmes de planification de tâches des outils ETL ?

### Gestion des modifications et des mises à jour de données

- Pour alimenter des bases de données de reporting depuis des bases de production alimentées par des applications de traçabilité, est-il préférable d'opter pour un mode Pull avec planification déterminée et synchrone ou un mode Push au fil de l'eau avec un système de Queue asynchrone ?
- Pour effectuer des mises à jour d'une base de reporting en mode delta (prise en compte des seules données nouvelles ou modifiées), est-ce que la fonctionnalité intitulée Service Broker peut-elle être pertinente ?
- Les procédures stockées d'extraction des changements de type "fn\_cdc\_get\_all\_change()" sont-elles facilement utilisables pour pallier des traitements ETL de reporting ?
- A partir de quel volume de triggers, les déclencheurs peuvent-ils impacter les performances d'une base de données ?
- Pour obtenir les dernières données ajoutées, modifiées voire supprimées, quelles sont les alternatives les plus appropriées :
  - Système artisanal à base de Timestamp et de triggers implémentés sur toutes les tables des bases de production,
  - Des tâches planifiées déclenchant régulièrement un audit de changement de données,
  - Le recours à un suivi de tables de journalisation mises à jour par des déclencheurs,
  - Les fonctionnalités de Change Tracking ?
  - Les fonctionnalités de Change Data Capture ?