

Avant-propos

Avez-vous bonne mémoire ? Lorsque, dans une conversation, vous devez citer des chiffres, une anecdote lue dans le journal, prononcer l'adjectif qui décrit précisément votre pensée, y parvenez-vous sans hésiter ? Si vous y arrivez la plupart du temps de manière satisfaisante, n'y a-t-il pas, parfois, des mots qui vous échappent, des informations qui vous restent sur le bout de la langue, jusqu'à ce que vous vous en souveniez subitement, lorsqu'il est bien trop tard ?

Parfois, votre mémoire est plus qu'un outil de conversation : lorsque vous devez retrouver votre chemin dans les rues d'une métropole, ou vous souvenir s'il faut couper le fil bleu ou le fil rouge pour désamorcer une bombe prête à exploser... Même si vous n'êtes pas souvent confronté – espérons-le – à ce dernier cas, vous comprenez que parfois, accéder à vos souvenirs rapidement, de façon fluide, sans à-coups, est une nécessité.

Un système de gestion de bases de données relationnelles (SGBDR) est comme une mémoire : il contient des données importantes, sinon vitales, pour l'entreprise, et la capacité donnée aux acteurs de cette entreprise de pouvoir y accéder efficacement, rapidement, avec des temps de réponse consistants, est essentielle.

Le problème des performances des bases de données s'est posé dès leur apparition. L'abstraction bâtie entre le stockage physique des données et leur représentation logique à travers différents modèles – le modèle relationnel étant aujourd'hui prédominant – nécessite de la part des éditeurs de ces logiciels un très important travail de développement. Il faut d'abord traduire les requêtes exprimées dans des langages déclaratifs comme le SQL (*Structured Query Language*) en stratégies de recherche de données optimales, puis parvenir à stocker les informations de la façon la plus favorable à leur parcours et à leur extraction. *A priori*, rien dans la norme SQL ni dans la

théorie relationnelle ne concerne les performances. C'est une préoccupation qui ne peut venir qu'ensuite, comme on ne peint les façades que lorsque la maçonnerie est terminée.

Toutefois, les performances ne font pas partie de la façade, mais plutôt du moteur de l'ascenseur, et sont donc notre préoccupation quotidienne. Le modèle relationnel et le langage SQL sont des existants à notre disposition, notre SGBDR est installé et fonctionnel, notre responsabilité, en tant qu'administrateur de bases de données, développeur ou responsable IT, est d'en assurer le fonctionnement optimal. Pour cette tâche, il n'y a pas de norme, de commandements gravés dans la pierre, d'outils ou de modèles préexistants qui vont effectuer le travail pour nous. L'optimisation est une préoccupation constante, une tâche toujours à continuer, à modifier, à compléter. C'est aussi une forme d'artisanat, qui nécessite de bonnes connaissances du système, de la curiosité et de la patience.

Concrètement

Optimiser, cela veut dire concrétiser notre connaissance de la théorie, dans la pratique, c'est-à-dire prendre en compte la façon dont cette théorie a été mise en œuvre dans une implémentation particulière. Cet ouvrage est dédié à l'optimisation pour SQL Server 2005 et 2008, le SGBDR de Microsoft. Les deux versions ne présentant pas d'énormes différences de ce point de vue, la plupart des informations contenues dans ce livre peuvent s'appliquer indifféremment aux deux versions. Nous indiquerons dans le corps de l'ouvrage lorsque telle ou telle fonctionnalité est propre à une version.

Nous y aborderons tous les points de l'optimisation (matériel, configuration, requêtes SQL), d'un point de vue essentiellement pratique, tout en vous présentant ce qui se passe « sous le capot », connaissance indispensable pour vous permettre de mettre en œuvre efficacement les méthodes proposées. Notre objectif est de vous guider à travers les outils, concepts et pratiques propres à l'optimisation des performances, pour que vous puissiez, concrètement et de façon autonome, non seulement bâtir un système performant, mais aussi résoudre au quotidien les problèmes et lenteurs qui pourraient survenir. L'optimisation, le *tuning* sont des pratiques quotidiennes : ce livre cherche donc à vous donner tous les bons outils et les bons réflexes pour vous permettre d'être efficace jour après jour.

À qui s'adresse ce livre ?

Il s'agit donc d'un livre essentiellement pratique. Il s'adresse à toute personne – développeur, DBA (administrateur de bases de données), consultant, professionnel IT – qui doit s'assurer du fonctionnement optimal de SQL Server. Nous ne présenterons pas les bases théoriques des SGBDR, et ce livre ne contient pas d'initiation à SQL Server. L'optimisation étant un sujet plus avancé, nous partons donc du principe que vous êtes déjà familier avec les bases de données, et spécifiquement SQL Server. Pour profiter de toute la partie dédiée à l'optimisation du code SQL, vous devez connaître au moins les bases du langage SQL et de ses extensions Transact-SQL (utilisation de variables, structures de contrôle...).

L'optimisation recouvre plusieurs domaines, qui peuvent être clairement délimités dans votre entreprise, ou non. Les parties de ce livre développent les différents éléments d'optimisation à prendre en compte. Si vous vous occupez exclusivement d'administration et d'installation matérielles, la première partie, « Optimisation du système », traite du matériel et de la configuration. Elle présente l'architecture de SQL Server et les principes de l'optimisation, et jette donc des bases théoriques pour aborder les parties pratiques. Si vous êtes principalement développeur et que votre souhait est d'écrire du code SQL plus performant, la seconde partie, « Optimisation des requêtes », vous est dédiée. Elle traite non seulement du code SQL lui-même, mais des mécanismes transactionnels pouvant provoquer des attentes et des blocages, et de l'utilisation d'index pour offrir les meilleures performances possibles.

L'auteur

Rudi Bruchez travaille avec SQL Server depuis 1998. Il est aujourd'hui expert indépendant spécialisé en SQL Server, basé à Paris. Il est certifié MCDBA SQL Server 2000 et MCITP SQL Server 2005. Il est également MVP (*Most Valuable Professional*) SQL Server depuis 2008. Dans son activité de consultant et formateur, il répond quotidiennement à toutes les problématiques touchant aux parties relationnelles et décisionnelles de SQL Server, notamment en modélisation, administration et optimisation.

Vous trouverez une page dédiée aux éventuels errata de ce livre sur www.babaluga.com le site de l'auteur. Vous pouvez également consulter des articles concernant SQL Server sur rudi.developpez.com. L'auteur répond régulièrement sur le forum SQL Server de developpez.net et sur le [newsgroup microsoft.public.fr.sqlserver](http://newsgroup.microsoft.public.fr.sqlserver).

Termes utilisés

Pour éviter les répétitions excessives, nous utiliserons les abréviations courantes dans le monde SQL Server, et le monde des systèmes de gestion de bases de données relationnelles (SGBDR). En voici la liste :

- **BOL** (*Books Online*) : l'aide en ligne de SQL Server.
- **SSMS** (*SQL Server Management Studio*) : l'outil client d'administration et de développement de SQL Server.
- **RTM** (*Release to manufacturing*) : version stable, en première livraison, avant la sortie de *service packs*.
- **SP** (*Service Pack*) : mise à jour importante d'une version, comportant des correctifs et des améliorations.
- **VLDB** (*Very Large DataBase*) : base de données de très grande volumétrie.
- **CTP** (*Common Technology Preview*) : versions de pré-sortie de SQL Server, livrées bien avant la sortie officielle du produit, et contenant uniquement les fonctionnalités implémentées complètement à ce moment.

- **OLTP** (*Online Transactional Processing*) : une utilisation particulière des données, principalement transactionnelle : écritures constantes, lectures de petits volumes. Ce sont les bases de données opérationnelles traditionnelles.
- **OLAP** (*Online Analytical Processing*) : une utilisation d'un entrepôt de données dans un but d'analyse. Ce qu'on nomme parfois l'utilisation décisionnelle, ou la *Business Intelligence*. Les bases de données OLAP sont utilisées principalement en lecture, pour des extractions massives.
- **MSDN** (*Microsoft Developer Network*) : documentation et articles en ligne sur Internet ou sur CD-ROM/DVD, représentant une base de connaissance très fournie sur les produits Microsoft, à destination principalement des développeurs. Accès libre sur <http://msdn.microsoft.com/>
- **Drapeau de trace** (*Trace flags*) : *switches* numériques, qui permettent de modifier le comportement de SQL Server. On les utilise comme paramètres au lancement du service, ou on les active/désactive à chaud, à l'aide des commandes DBCC TRACEON et DBCC TRACEOFF.

Exemples de code

Le code contenu dans cet ouvrage est disponible sur la page dédiée à l'ouvrage sur le site des éditions Dunod, www.dunod.com, et sur le site de l'auteur, www.babaluga.com

Il est fondé sur la base de données d'exemple de Microsoft, nommée AdventureWorks. Elle représente une entreprise fictive du même nom qui vend des bicyclettes. Vous pouvez télécharger AdventureWorks pour SQL Server 2005 ou SQL Server 2008 à l'adresse :

<http://www.codeplex.com/MSFTDBProdSamples>.

Nous utiliserons souvent en exemple la table Person.Contact, elle contient 19 972 lignes de contacts, avec des colonnes simples comme FirstName, LastName et EmailAddress.

Utilisation de l'anglais

Nous avons fait le choix d'utiliser une version anglaise de SQL Server, principalement parce que, à l'heure où nous rédigeons ce livre, SQL Server 2008 n'est pas encore sorti en version finale, et n'existe donc qu'en anglais. De plus, certaines traductions de l'interface française sont plus troublantes qu'utiles. Nous indiquerons parfois en regard le terme français. La correspondance n'est pas difficile à faire avec une version française : les entrées de menus et la disposition dans les boîtes de dialogue sont les mêmes.

Remerciements

Merci à Frédéric BROUARD, MVP SQL Server, fondateur de la société SQL Spot, pour ses conseils et suggestions et sa relecture. Merci également à Pascale DOZ, experte indépendante, pour sa relecture et son aide. Merci à Christian ROBERT, MVP SQL Server, pour ses astuces. Et enfin, merci également à Edgar Frank CODD pour avoir tout inventé.