

# SQL Server 2005 pour l'entreprise

en coédition avec  
Learning Tree  
FRANCE



*On vous l'a sans doute déjà dit, cette nouvelle version de SQL Server s'avère moins cher en terme d'achat de licences que son principal concurrent. Moins cher aussi en terme de développement, utilisation et surtout d'administration. Mais ce n'est pas sur le terrain commercial qu'il nous faut comprendre en quoi SQL Server 2005 apporte de multiples bénéfices à l'entreprise. Non, c'est plutôt en terme de fonctionnalité que le sujet est intéressant. Qu'apporte SQL Server de mieux que la concurrence, qu'apporte t-il de plus que les autres n'ont pas ?*

*Notons tout d'abord que le produit se pose en concurrent solide et inévitable sur trois axes majeurs : les VLDB (Very Large DataBases, c'est à dire des solutions de données en ligne de plus d'un téra octets), le décisionnel (le data warehouse, comme les data marts) et enfin le web et Internet. Mais dans le trésor se cache une pépite : service broker et son complément « http endpoint » pourrait bien être l'outsider qui fera de cette version un produit unique et indispensable en terme de souplesse, renouvelant ainsi la microinformatique dans ce qu'elle avait de noble et pur à ses origines : mettre l'ordinateur au plus près de l'utilisateur...*



Par Frédéric Brouard - MVP SQL Server  
Expert SQL et SGBDR, Auteur de :

- SQL, Développement, Campus Press 2001
- SQL, collection Synthex, Pearson Education 2005, co écrit avec Christian Soutou
- <http://sqlpro.developpez.com> (site de ressources sur le langage SQL et les SGBDR)
- Enseignant aux Arts & Métiers et à l'ISEN Toulon

Copyright et droits d'auteurs : la Loi du 11 mars 1957 n'autorisant aux termes des alinéas 2 et 3 de l'article 41, d'une part que *des copies ou reproductions strictement réservées à l'usage privé et non [...] à une utilisation collective*, et d'autre part que les analyses et courtes citations dans un but d'illustration, toute reproduction intégrale ou partielle faite sans le consentement de l'auteur [...] est illicite. Le présent article étant la propriété intellectuelle conjointe de Frédéric Brouard et de Learning Tree, prière de contacter l'auteur pour toute demande d'utilisation, autre que prévu par la Loi à [SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)

## 1) Percée significative dans les VLDB

Vous ne le savez peut être pas, mais SQL Server est déjà leader sur sa plateforme dans le monde des VLDB. Près de 80 % des bases de données de plus de 1 To sont déjà sous MS SQL Server, bien avant que ne pointe la version 2005 (source WinterCorp).

Il manquait au produit quelques éléments, quelques améliorations pour en faire un serveur apte aux contraintes des très grandes bases de données, pour lesquels le maître mot est la continuité de service.

Les VLDB (Very Large DataBases) mettent en oeuvre des techniques particulières pour répartir la charge et assurer une haute disponibilité des données. Elles nécessitent des procédures spécifiques afin d'assurer des opérations de maintenance qui perturbent le moins possible la qualité du service.

C'est donc dans ce sens que Microsoft a travaillé : désormais la plupart des opérations de maintenance peuvent se faire à chaud, sans interruption de service, tout en minimisant les ressources affectées par leurs exécutions. Même l'OS Windows 2003 Server a été revu afin, par exemple, de proposer sur les ordinateurs qui disposent de cette fonctionnalité, d'ajouter de la mémoire vive à chaud et bien entendu de la gérer.

Bien des opérations traditionnelles de maintenance peuvent désormais être opérées « in line ». C'est le cas notamment de tout ce qui permet d'améliorer l'indexation (défragmentation, reconstruction, activation, désactivation...).

Dans le même esprit SQL Server est capable de travailler sur une base de données alors qu'elle est en cours de restauration. Il est même possible de restaurer un fragment de la base, une page de données par exemple, sans altérer le service (restauration « *peacemeal* »).

En vue de d'améliorer la rapidité des recherches de données, SQL Server 2005 apporte la possibilité de créer des index plus couvrant par copie de certaines informations. On aura intérêt à utiliser une telle technique afin que certaines des requêtes les plus consommatrices évite une double lecture : celle de l'index puis celle de la table.

C'est aussi l'optique des VLDB qu'a été mise en place la possibilité de répartir les données d'une table sur plusieurs fichiers logiques (par exemple sur plusieurs agrégats de disques). Le principe mis en place par les équipes de développement de Redmond est simple : un schéma de répartition, une fonction de répartition qui utilise le schéma de répartition et un objet (table

ou index) qui s'appuie sur la fonction de répartition. Tout le reste est absolument transparent aux développeurs comme aux utilisateurs.

*Exemple de partitionnement de données pour une table :*

```
-- création de la fonction de répartition (trois bornes)
CREATE PARTITION FUNCTION PF_CUTTER (CHAR(25))
AS RANGE LEFT FOR VALUE ('COSTAZ', 'HOENNER', 'PANIER');

-- création du schéma de partitionnement des données (quatre espaces de stockage)
CREATE PARTITION SCHEMA PS_CUTTER
AS PARTITION PF_CUTTER
TO (FileGroup1, FileGroup2, FileGroup3, FileGroup4);

-- création de la table s'appuyant sur le schéma de partitionnement
CREATE TABLE T_PAGES_JAUNES_PGJ
(PGJ_ID          INTEGER          NOT NULL PRIMARY KEY NONCLUSTERED,
 PGJ_ENSEIGNE   VARCHAR(128)    NOT NULL,
 PGJ_TEL        CHAR(10)         NOT NULL,
 ...
) ON PS_CUTTER (PGJ_ENSEIGNE);

-- requête permettant de déterminer dans quelle partition se trouve les données
SELECT PGJ_ID, PGJ_ENSEIGNE, $PARTITION.PF_CUTTER(PGJ_ENSEIGNE) AS PARTITION
FROM   T_PAGES_JAUNES_PGJ;
```

*Une table des « pages jaunes » d'un annuaire est équirépartie en quatre partitions suivant une « table » de Cutter (répartition équitable des données suivant les fréquences d'apparition statistiques des noms de famille français).*

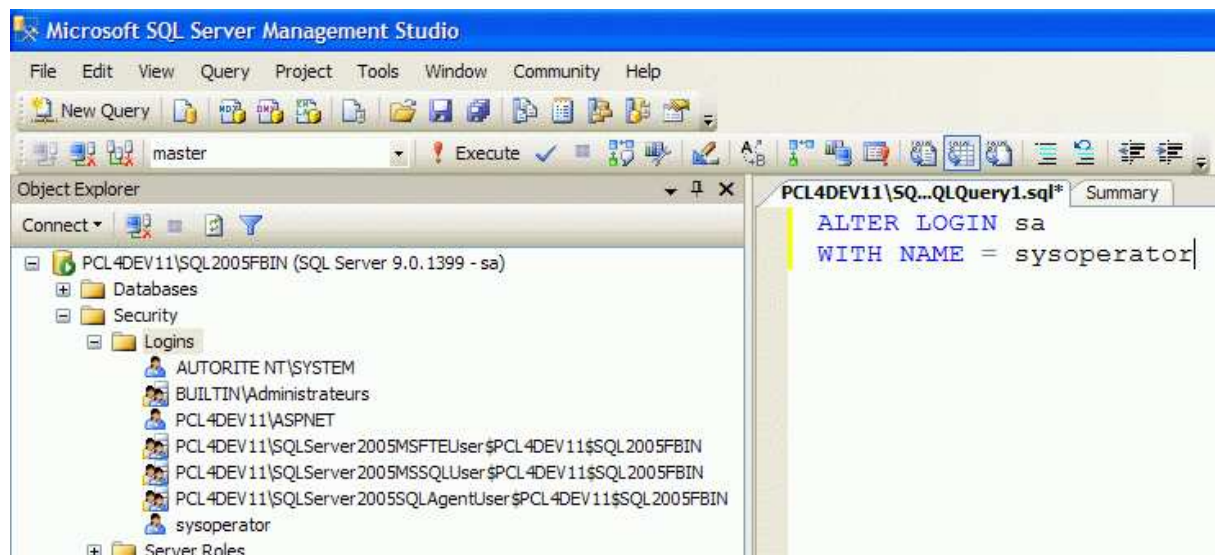
## Le snapshot

Il est un point qui naguère faisait la différence entre SQL Server et son plus agressif concurrent : la possibilité de lectures stables et cohérentes sans nécessité de poser des verrous. Cela est désormais possible à l'aide du concept de « snapshot » que l'on peut traduire par « cliché » ou « instantané ». Il s'agit de photographier à un instant  $t$  les données de la base afin de les rendre lisibles et manipulables à souhait sans jamais n'être bloqué par ce que font d'autres utilisateurs. Ce concept de snapshot a été introduit à différents niveaux : une simple modification du comportement par défaut de la base suffit à établir ce mécanisme sans qu'il soit nécessaire de retoucher son code. Mais on peut aussi bien utiliser le snapshot pour certaines transactions en pilotant le niveau d'isolation de la transaction. Enfin, on peut créer une base de données sur ce principe, c'est à dire une copie des données source vers une base cible à un instant  $t$ . Le mécanisme sous-jacent est suffisamment intelligent pour ne copier que le nécessaire, c'est à dire le différentiel entre la base source et le cliché minimisant ainsi le volume des données et les risques de redondance. La création du cliché est donc instantanée.

En matière de temps de réponse, la différence entre la version 2005 de MS SQL Server apporte un gain de 35 % en moyenne par rapport à la précédente. Mais certains traitements vont jusqu'à 1,5 fois plus vite, notamment dans les mécanismes d'import/export.

## Une sécurité nettement améliorée

S'il est un point dur dans les grandes bases de données, c'est celui de qui à accès et à quoi. En terme de sécurité, cette nouvelle version propose de multiples nouveautés : authentification et cryptage en sont les principaux atouts. L'introduction d'une véritable gestion des schémas facilite l'octroi des privilèges sur les éléments des bases de données. Même la syntaxe des commandes pour ce faire a été entièrement revue. Fini les procédures stockées aux noms barbares comme *sp\_addlogin*, vive le CREATE LOGIN... Et pour couronner le tout le compte par défaut, ce fameux « sa » est redéfinissable à votre gré ! Quand à la création des mots de passe pour les connexions à SQL Server, ces derniers peuvent être contraints à se conformer à des règles de formations élaborées pour les comptes OS, tel qu'avoir des caractères alphabétiques et numériques, une longueur supérieure à *n*, ect...



*Même le compte de connexion « sa » peut maintenant disparaître !*

Enfin, la possibilité de « connexion administrative dédié » (DAC : Dedicated Administrator Connection) a été ajoutée à l'utilitaire SQLcmd.exe et permet d'attaquer une instance de SQL Server 2005 en ligne de commande, afin de procéder à tout type d'opération et particulièrement les manipulations à effectuer dans l'urgence.

Un nouveaux type de réplication voit le jour avec cette version : la réplication « poste à poste » (Peer to Peer Transactionnal Replication). Elle permet de synchroniser les données de différents serveurs lorsque ces derniers sont segmentés pour la mise à jour et veulent quand même pouvoir interroger localement la base complète.

Mais s'il est une amélioration phare en terme de haute disponibilité et donc de VLDB, c'est certainement le concept de « mirroring » : une base est

répliquée de manière continue (pas de temps de latence) sur un serveur distant. C'est le principe du miroir. Un troisième serveur SQL (qui peut être le « run time » SQL Server Express) observe le comportement du répliquant et du répliqué. En cas d'anomalie le basculement automatique s'effectue en trois secondes. Aucune nécessité de serveurs identiques, ni de matériel approuvé. Bref une économie très substantielle tant sur le hardware que sur l'administration.

Enfin, SQL Server 2005 intègre un ensemble de plusieurs centaines de vues « systèmes » dites Dynamic Management View, afin de permettre d'auditer en permanence l'activité du serveur sur tous les plans. De requêtes plus simples et plus profondes, pour savoir, en temps réel si le besoin s'en fait sentir, ce qui se passe dans la machine à un instant t.

\* \* \*

De grand éditeurs comme SAP ne s'y sont pas trompé : SQL Server 2005 devient un acteur majeur et indispensable du monde des VLDB. Depuis quelques temps, SAP s'investit fortement dans MS SQL Server. Et les résultats sont là !

L'exemple des papeteries Koehler en Allemagne est révélateur : opérant sur cinq sites différents cette entreprise ne peut se permettre d'interrompre sa production pour un « plantage » du système informatique. Elle en a fait la cruelle expérience. Le bilan de trois jours d'arrêt de production fut une perte sèche de 3 000 000 \$. L'entreprise s'est alors doté de MS SQL Server 2005 pour sa capacité de mirroring. Ce fut l'un des tout premier à adopter ce système, en plateforme 64 bits, pour le pilotage de sa production sous SAP. Koehler a analysé en termes financier sa solution : une seule panne évitée dans les cinq ans c'est un retour sur investissement de 200 %. De même, les coûts administratifs ont été réduits d'environ 120 000 € par an.

VLDB aussi pour la Mediterranean Shipping Company qui exploite de façon intensive des bases de données de plusieurs Tera octets et ce de façon mondiale : au sommet de sa pyramide informatique on trouve 18 serveurs octoprocesseurs dotés de 8 Go de mémoire vive sous forme de clusters à deux nœuds en mode actif/passif. Tout cela pour plus de 15 millions de transactions par jour. Autant dire que la haute disponibilité à 99,999 % n'est pas simplement nécessaire : elle est indispensable... Et à ce niveau, l'indispensable haute disponibilité se doit d'être doublée par des outils d'audit en ligne très efficaces : « les différents outils de surveillance, notamment Dynamic Management View, intégrés à SQL Server 2005, ajoutés à la combinaison PerfMon/SQL Trace nous permettent de localiser très rapidement un problème de performances » explique Bob Erickson, Vice Président Directeur de MSC, chargé du développement des logiciels.

## 2) le décisionnel comme fer de lance

S'il est un marché dont la croissance à bien profité à Microsoft, c'est celui du décisionnel. Loin devant les spécialistes du secteur, la firme de Redmond a affiché une insolente croissance à deux chiffres depuis quelques années, tandis que les acteurs traditionnels stagnaient. Pourquoi en est-il ainsi ? Tout simplement parce que Microsoft apporte une offre cohérente...

*Part de marché OLAP en 2004 selon "olapreport.com"*

Microsoft	27.4%
Hyperion	20.7%
Cognos	14.1%
Business Objects	7.2%
MicroStrategy	7.1%
SAP	6.0%
Oracle	3.7%
Applix	3.1%
Cartesis	3.1%
Systems Union/MIS AG	3.0%
Geac	1.9%
SAS Institute	0.9%

Une solution de décisionnel doit être constitué de différents éléments :

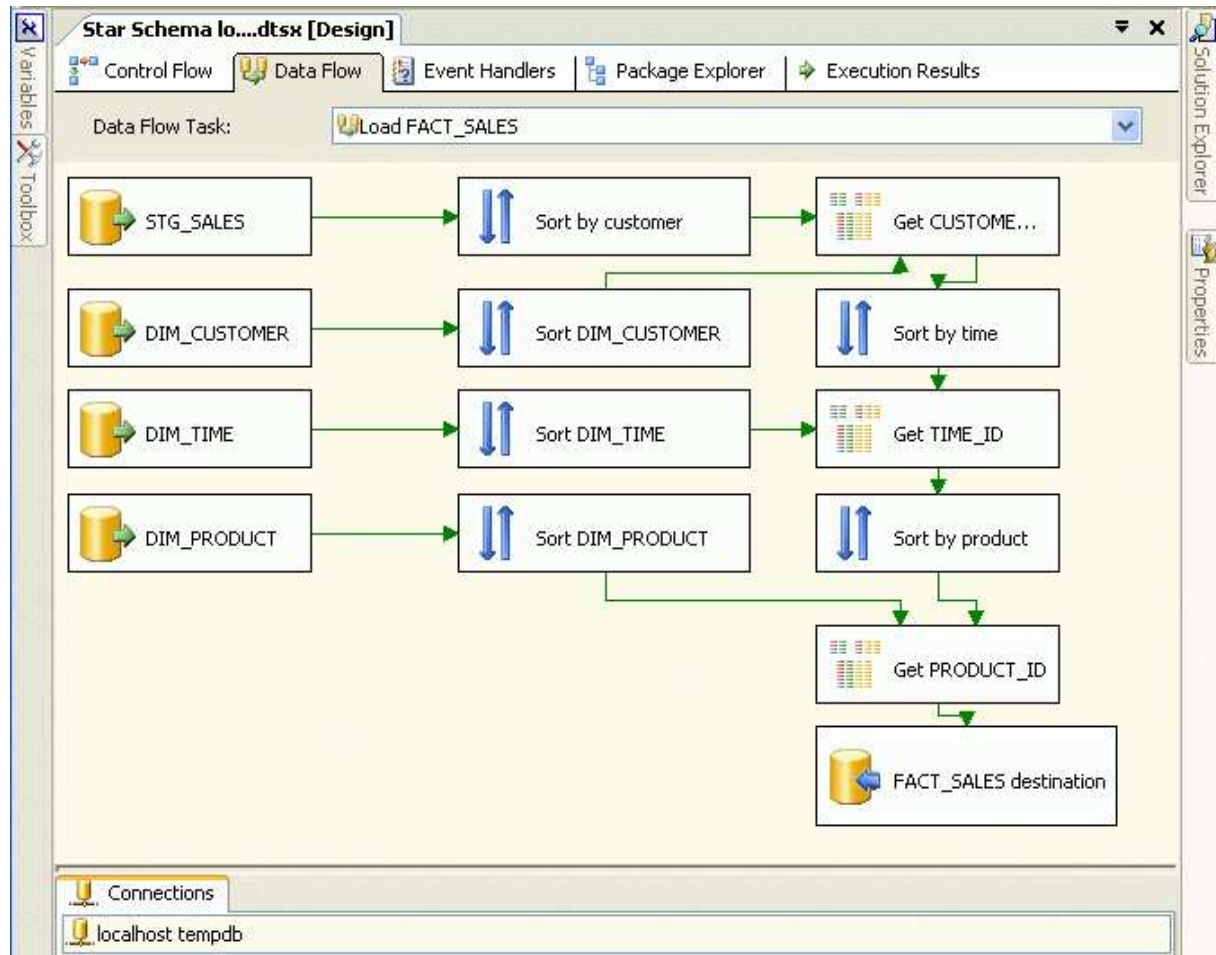
- un outil d'extraction transformation et chargement (ETL : Extract, transform and Load)
- un espace de stockage des données (la base de données)
- des outils pour interroger les données et en extraire des corrélations
- un générateur d'état pour présenter les données

Jusqu'à présent, peu d'éditeurs proposaient l'ensemble de ces services de manière cohérente et à un coût acceptable. Microsoft à réalisé cela en se basant sur trois briques et un standard : SQL Server, Analysis Services et Reporting Services, le tout associé au langage MDX. La dernière des briques, Reporting Services étant la plus récente, ce n'est donc que depuis quelques années que cette offre est proposée. Et constatons que le langage MDX de requête dans les cubes est devenu en quelques années, le standard de l'interrogation décisionnelle (citons parmi les éditeurs ayant adoptés MDX : Cognos, Business Objects, Crystal, Hyperion, Panorama, Proclarity, Brio...). Avec la version 2005, d'importantes améliorations ont été apportées.

### Un véritable ETL

Tout d'abord l'outil d'ETL (Extract, transform and Load) a été entièrement revu. La mauvaise nouvelle c'est que l'ancien module de nom DTS (Data Transformation Services) est remplacé par SSIS (SQL Server Integration

Services) et qu'il faut bien l'avouer, même si Microsoft fournit des outils pour migrer ses « lots DTS » vers des packages SSIS, la plupart des choses sont à récrire. La bonne, c'est que la fiabilité a été très largement éprouvée et que les vitesses de traitement de SSIS, sont largement supérieures à celles de DTS.



*Un « lot » SSIS : des sources (extractions de données), des puits (chargements de données), des parcours avec des règles de gestion (transformations).*

*© Jamie Thompson, avec son aimable autorisation*

Mais les enrichissements les plus significatifs consistent dans le traitement qualitatif des données lors de la phase d'importation, et l'arrivée de nouveaux algorithmes de data mining.

Désormais la phase de recollement des données entre la base OLTP et les dimensions de la base décisionnelle peuvent être opérés à l'aide d'algorithmes en logique flou (cartographie des données par approches) voire, à l'aide des algorithmes du data mining. Ce sont d'ailleurs cinq nouveaux algorithmes qui ont été ajoutés à la panoplie de l'existant. Mais rien ne vous empêche d'en rajouter, car désormais les algorithmes sont enfichables (plugs in), ni rien ne vous empêche d'en développer de nouveaux vous même !

Analysis Services, qui est à la fois le moteur de stockage et d'interrogation des données, a lui aussi évolué : les performances de manipulation des cubes OLAP ont presque doublées avec l'introduction d'UDM (Unified Dimension Model) dont les composantes clefs sont l'accès à des données hétérogènes, un modèle utilisateur final riche, des capacités d'analytiques sophistiquées et une mise en cache proactive.

## Générer des rapports et les présenter

Le système d'information décisionnel constitue une synthèse des données opérationnelles, dont les dimensions sont choisies pour leur pertinence et leur transversalité fonctionnelles. Cette masse d'information est stockée dans des structures particulières telles que les datawarehouse ou les bases de données OLAP.

L'intérêt du décisionnel est d'offrir au décideur une vision transversale de l'entreprise à travers des indicateurs calculés et présentés à l'aide d'outils spécifiques. La finalité d'un système décisionnel est le pilotage de l'entreprise.

Bien entendu la solution de génération d'état imprimés basés sur l'analyse dimensionnelle repose sur Reporting Services, une solution qui n'est plus à vanter tant elle s'est rapidement introduite dans l'entreprise et qui, surtout, permet de manipuler les données à partir du poste client. Ajoutons à cela Maestro et SharePoint deux solutions complémentaires pour la diffusion de l'information provenant de l'analyse dimensionnelle.

Avec Maestro, Microsoft propose un outil d'élaboration et de présentation d'indicateurs de performance pour l'entreprise, en remplacement de Business Scorecard Accelerator. Cette solution permet de construire, alimenter et diffuser des indicateurs de performance regroupés au sein de tableaux de bord. Les indicateurs sont doublés par des pictogrammes et le dépassement de certains seuils peut générer différent type d'alertes envoyées notamment par email. Enfin, Maestro s'intègre dans SharePoint afin de bénéficier de la diffusion des informations dans un environnement collaboratif.

Parmi les bonnes nouvelles, on peut désormais auditer les requêtes produites lors des manipulations d'informations via Analysis Services. En effet, le Profiler SQL a été étendu à ce module ce qui permet désormais d'optimiser plus facilement le code le plus coûteux en priorité.

\* \* \*

C'est avec tous ces avantages et son faible coût de licences que l'entreprise de la grande distribution Conforama, s'est lancé dans le décisionnel avec SQL Server, après avoir testé différentes solution via Oracle ou DB2. A ce jour, la navigation dans le cube se fait au travers de quelques 2,5 Téra octets de données ! « Outre sa facilité et sa rapidité de mise en oeuvre, la solution de Microsoft s'est distinguée par des temps de réponse divisés par dix par rapport à notre ancienne solution, et par un délai de calcul des cubes de quarante-cinq minutes, là où il nous fallait huit heures auparavant » indique Joël Nicloux, directeur de l'organisation et des systèmes d'information.



C'est dans ce même esprit que le groupe Casino démarre sa restructuration informatique avec comme pierre angulaire SQL Server 2005. Pour le responsable des études, le choix du décisionnel a été guidé, outre les performances, par la facilité d'intégration : « plus on utilise de technologies, plus on tombe sur des problèmes d'intégration. Avec l'offre Business Intelligence de Microsoft intégrée à SQL Server, nous ne serons pas confrontés à des assemblages de briques qui se traduisent par des coûts en développement et des délais de mise en oeuvre inacceptables dans un secteur tel que le notre ou la réactivité est primordiale. » car chez Casino, tout comme chez Conforama, lorsque l'on parle de décisionnel, on parle de Téra octets...

Pour 123 Multimedia, c'est déjà 4,5 To de données qui sont en ligne sur SQL Server 2005 pour 90 millions de requêtes SQL par jour avec des pointes de 2 000 transactions par seconde et parfois plus de 45 000 utilisateurs simultanés...

### 3) Tout pour le Web

Cela fait maintenant plusieurs années que le couple SQL Server / IIS sous Windows constitue la solution la plus puissante en terme de sites actifs Internet (voir le TPC-W sur tpc.org). On peut donc se demander s'il y a bien quelque chose de nouveaux dans la version 2005 de SQL Server ? Bien entendu, la réponse est oui. Il ne s'agit pas pour Microsoft de se reposer pour ses lauriers. La concurrence est encore possible et la position de leader absolu dans ce secteur ne peut se maintenir qu'au prix d'une évolution positive.

#### XML dans les données

Parmi les attentes majeures de la clientèle, il y avait l'introduction d'XML au sein des types de données qu'une base peut prendre en compte.

C'est désormais fait et cela de manière élégante. Contrairement à certains SGBDR, le type XML de MS SQL Server 2005, n'est pas un « BLOB ». C'est un véritable type de données, qui n'admet le stockage de documents XML que s'ils sont « biens formés ». De plus, la définition d'une telle colonne au sein d'une table, peut faire l'objet d'un contrôle de validité renforcé sur la structure du document. Ainsi, la clause CONTENT ou DOCUMENT permet de spécifier une collection de schémas XML pour laquelle un document XML doit se conformer.

Le marché de l'e-commerce à représenté un chiffre d'affaire de 8,7 milliards d'euros en 2005, soit une croissance de 53% d'une année à l'autre.

Source FEVAD (Fédération de la vente à distance)

Enfin une colonne de type XML peut être indexée afin d'accélérer les recherches sur les valeurs, les attributs ou encore les chemins...

Microsoft a poussé très loin ses investigations sur les problématiques de manipulation des données de type

XML. Par exemple l'une des difficultés qui consiste à mettre en relation les données internes d'un document XML et les colonnes de la table dans laquelle le document figure, a été résolu avec l'introduction d'un nouvel opérateur relationnel : APPLY.

Il permet de faire une jointure entre les colonnes de la table et les données XML qui sont, elle-même des données tabulaires, en quelques sortes une table dans une colonne d'une table.

Ce paradoxe de mise en relation est donc opéré avec OUTER APPLY ou CROSS APPLY entre colonne de la table et attribut XML.

#### *Intéropération de données relationnelles et xml*

```
-- création d'un schéma XML pour validation des données d'une colonne XML
```

```
CREATE XML SCHEMA COLLECTION XSC_DONNEES_ENTREPRISE as N'
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:company="http://www.exemple.com/entreprise"
targetNamespace="http://www.exemple.com/entreprise"
elementFormDefault="qualified">
  <xs:element name="personnel">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="employe">
          <xs:complexType>
            <xs:all>
              <xs:element name="nom" type="xs:string" />
            </xs:all>
            <xs:attribute name="matricule" type="xs:ID" />
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>'
```

```
-- cette table possède une colonne de type XML dont les données
-- doivent se conformer au schéma ci avant
```

```
CREATE TABLE T_ENTREPRISE_EMPLOIS_EEP
(EEP_ID          INTEGER PRIMARY KEY IDENTITY,
 EEP_NAME        VARCHAR(12),
 EEP_XML_DATA    XML (XSC_DONNEES_ENTREPRISE))
```

```
-- insertion de deux listes d'employés
-- pour les entreprises IBM et Microsoft
```

```
INSERT INTO T_ENTREPRISE_EMPLOIS_EEP
VALUES ('IBM',
'<personnel xmlns="http://www.exemple.com/entreprise">
  <employe matricule="M0108"><nom>Jean</nom></employe>
  <employe matricule="X0017"><nom>Jacques</nom></employe>
  <employe matricule="V0009"><nom>Jules</nom></employe>
</personnel>')
```

```
INSERT INTO T_ENTREPRISE_EMPLOIS_EEP
VALUES ('Microsoft',
'<personnel xmlns="http://www.exemple.com/entreprise">
  <employe matricule="AC23"><nom>Marc</nom></employe>
  <employe matricule="VF42"><nom>Marcel</nom></employe>
</personnel>')
```

```
-- extraction de la liste des employés de toutes les entreprises
-- dont le nom comporte la lettre 'a'
```

```

WITH XMLNAMESPACES (DEFAULT 'http://www.exemple.com/entreprise')
SELECT EEP_ID as [Clef entreprise],
       NoeudXML.value( './@matricule', 'nvarchar(10)' ) as [Matricule
employé],
       NoeudXML.value( './nom', 'nvarchar(16)' ) as [Nom employé]
FROM   T_ENTREPRISE_EMPLOIS_EEP
       CROSS APPLY EEP_XML_DATA.nodes('/personnel/employe') as T(NoeudXML)
WHERE  NoeudXML.value( './nom', 'nvarchar(16)' ) LIKE '%a%'

```

Clef entreprise	Matricule employé	Nom employé
3	M0108	Jean
3	X0017	Jacques
4	AC23	Marc
4	VF42	Marcel

*Dans cet exemple, la liste des employés a été extraite de chaque document xml et mise en relation par jointure croisée avec les informations des colonnes de la table. La requête mêle les langages SQL, XQuery et XPath sur des données SQL et XML...*

## MARS à la rescousse du Web

Dans le cadre d'une minimisation de l'accès aux données, et ce afin de diminuer le trafic réseau entre serveur SQL et client web (IIS/ASP par exemple), SQL Server couplé à OLEdb permet maintenant d'envoyer simultanément et dans un seul aller l'ensemble des jeux de résultats de différentes requêtes SQL afin de composer d'un seul coup une page web aussi complexe soit-elle.

Cette technique appelée « MARS » pour Multiple Active Result Sets, n'est disponible qu'avec le middleware ADO.net 2.0 pour SQL Server 2005. Avec MARS, différentes requêtes lancées en parallèles peuvent leur jeu de résultats simultanément. Il ne reste qu'à parcourir les différents « result set » pour écrire les données. Pour peu que tout cela soit fait dans une transaction pilotée au niveau SNAPSHOT, la lecture en sera consistante et ne posera aucun verrou ! Bref, une diminution drastique des temps de latence, des allers et retour réseau et donc des temps de réponse notablement raccourcis associés à une réduction sensible des phénomènes de contention.

Fini les multiples allers et retours entre le serveur web et SQL Server afin de générer une page dynamique dans laquelle figure cinq ou six tableaux, sous tableaux et résultats agrégés. Désormais l'ensemble des données à présenter sur une page web peut être servit en un seul jet.

Enfin, deux nouveautés et pas des moindres :

- La possibilité d'exposer des services web (SOAP) directement depuis le serveur SQL sans passer par un quelconque serveur web ! Bien entendu il ne s'agit pas de remplacer IIS, mais de permettre, via une encapsulation de protocoles, de communiquer de manière applicatives plutôt que par le simple flux de données. C'est le module *http endpoint* que Microsoft a intégré à SQL Server 2005.

- La possibilité de notifier une information à des utilisateurs abonné à un service (Notification Services). Ce service d'information personnalisée se déclenche lors des mises à jour de données et transmet l'information actualisée aux abonnés via une grande variété de périphériques mobiles.

## 4) Faciliter le développement

S'il a fallu si longtemps à Microsoft pour sortir enfin une nouvelle version de son SGBDR, c'est que l'on a préféré attendre la mise au point de la nouvelle version de Visual Studio (et plus exactement la version 2 du framework .net) pour l'intégrer à SQL Server. D'où les numéros de version 2005 pour les deux produits. En fait un « CLR » (Common Language Runtime) destiné à exécuter le code .net propre à SQL est intégré au sein de MS SQL Server 2005.

Mais en quoi cela était-il nécessaire ?

Le langage SQL, et donc la manière de travailler avec un SGBDR (système de gestion de bases de données relationnelles) est normalisé depuis 1986. Les dernières versions : SQL:2003 (release mineure) et surtout SQL:1999 (release majeure) ont introduit le relationnel objet, comblant ainsi une partie du fossé entre les langages modernes et la façon d'attaquer les données.

Simplement parce que Microsoft offre désormais la possibilité de réaliser le code des procédures, fonctions et triggers à l'aide d'un quelconque des langages .net comme C# ou Visual Basic for .net. L'avantage est simple : si vos développeurs connaissent VB par exemple, il peuvent déjà commencer à coder pour SQL Server sans autre besoin de formation que de maîtriser les bases de données relationnelles. De plus, les langages .net offrent l'agrément de pouvoir faire l'objet d'une mise au point en mode débogage sous Visual Studio, ce qui n'est pas le cas du code Transact SQL spécifique à SQL Server. Même si le codage des procédures stockées en .net n'est pas toujours performant au regard de celui fait en Transact SQL, il peut s'avérer payant pour de petites

applications pour lesquelles la performance n'est pas le critère absolu. En revanche il se révèle performant pour du code portant sur de faibles accès aux données, avec des calculs complexes (calculs scientifiques ou financiers).

Mais le codage .net réserve quelques avantages certains : il rend possible l'écriture de fonctions de type « agrégat » et il permet de créer de nouveaux types de données SQL. Une fonction d'agrégation souvent demandée consiste à concaténer les données d'une colonne en une seule ligne. De même il sera possible de réaliser son propre code de calculs de médiane. Pour ce qui est de nouveaux types de données SQL, l'imagination sera sans doute sans limite : des points dans le plan ou l'espace, des période temporaires, des matrices, vecteur, nombres complexes... et cela dans un interfaçage simplissime et avec des requêtes SQL amenées à prendre en charge de tels

types et de telles fonctions en toute transparence. Car la force de SQL CLR, c'est sa simplicité d'invocation...

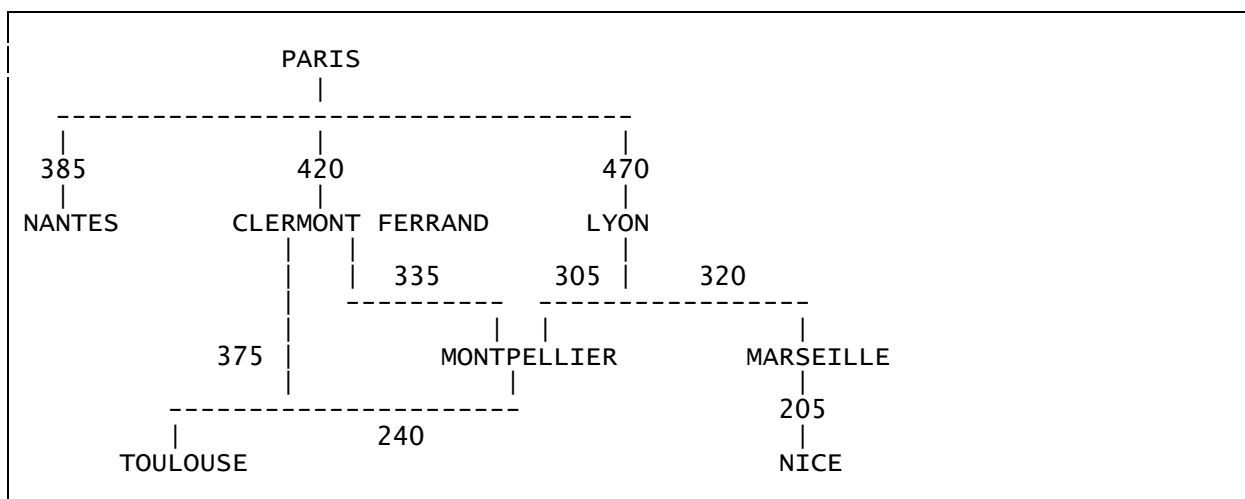
En matière de requêtes d'extraction, SQL Server 2005 apporte de nombreuses nouveautés. Je ne voudrais en citer que trois : les fonctions de fenêtrage statistiques (RANK, DENSE\_RANK, ROW\_NUMBER, NTILE), la clause OUTPUT et les expressions de table.

## SQL, un langage complet

C'est sans doute avec le concept d'expression de table, conforme à la norme SQL:1999 que SQL Server gagne du terrain. Le CTE ou « Common Table Expression », consiste à définir une table de travail à la volée, à l'aide d'une simple requête SELECT précédant la véritable extraction de données. En quelques sortes l'expression de table est une vue instanciée à l'exécution de la requête. De plus, de multiples CTE, se référant les uns aux autres peuvent ainsi s'enchaîner pour former un traitement relativement complexe. Mais finalement, le plus spectaculaire est que le CTE peut faire référence à lui même, introduisant ainsi la récursivité au sein de SQL...

*La CTE (Common Table Expression, en français "expression de table") vient au secours du développeur pour résoudre les extractions les plus complexes, y compris le traitement massivement récursif des données...*

*Soit le réseau autoroutier de la France (un peu simplifié pour les besoins de la démonstration..) :*



*Modélisé de la façon suivante :*

```

-- tables des routes
CREATE TABLE T_ROUTE_RTE
(RTE_DEPART VARCHAR(16),
 RTE_ARRIVEE VARCHAR(16),
 RTE_KM INTEGER)

-- insertion des données
INSERT INTO T_ROUTE_RTE VALUES ('PARIS', 'NANTES', 385)
INSERT INTO T_ROUTE_RTE VALUES ('PARIS', 'CLERMONT-FERRAND', 420)
INSERT INTO T_ROUTE_RTE VALUES ('PARIS', 'LYON', 470)

```

```
INSERT INTO T_ROUTE_RTE VALUES ('CLERMONT-FERRAND', 'MONTPELLIER', 335)
INSERT INTO T_ROUTE_RTE VALUES ('CLERMONT-FERRAND', 'TOULOUSE', 375)
INSERT INTO T_ROUTE_RTE VALUES ('LYON', 'MONTPELLIER', 305)
INSERT INTO T_ROUTE_RTE VALUES ('LYON', 'MARSEILLE', 320)
INSERT INTO T_ROUTE_RTE VALUES ('MONTPELLIER', 'TOULOUSE', 240)
INSERT INTO T_ROUTE_RTE VALUES ('MARSEILLE', 'NICE', 205)
```

*La question est de trouver à l'aide d'une requête SQL le plus court chemin pour aller de Paris à Toulouse ?*

```
-- la requête suivante qui utilise deux CTE imbriquées dont une récursive
-- donne la solution à ce problème dit « du voyageur de commerce »
WITH
  trajet (DESTINATION, ETAPES, DISTANCE, CHEMIN)
  AS
    (SELECT DISTINCT RTE_DEPART, 0, 0, CAST('PARIS' AS VARCHAR(MAX))
     FROM   T_ROUTE_RTE
     WHERE  RTE_DEPART = 'PARIS'
     UNION ALL
     SELECT RTE_ARRIVEE, part.ETAPES + 1,
            part.DISTANCE + arrive.RTE_KM,
            part.CHEMIN + ', ' + arrive.RTE_ARRIVEE
     FROM   T_ROUTE_RTE AS arrive
            INNER JOIN trajet AS part
                ON part.DESTINATION = arrive.RTE_DEPART),
  short (DISTANCE)
  AS
    (SELECT MIN(DISTANCE)
     FROM   trajet
     WHERE  DESTINATION = 'TOULOUSE')
SELECT *
FROM   trajet j
       INNER JOIN short s
           ON j.DISTANCE = s.DISTANCE
WHERE  DESTINATION = 'TOULOUSE'
```

*Dans cet exemple on cherche le plus court chemin dans un réseau routier (en fait un graphe au sens de la théorie mathématique). La CTE, par le fait qu'elle peut faire référence à elle-même au sein de la requête qui la construit à l'aide de la fonction ensembliste UNION ALL, permet de résoudre le problème ardu de la fermeture transitive et propose une solution élégante pour les parcours d'arbre ou de graphe.*

C'ainsi que les problèmes de recherche opérationnelle, tels l'optimisation d'affectations (logistiques), sont désormais solvable à l'aide d'un SELECT... Fini les curseurs de lecture, fini les tables temporaires et leurs résultats intermédiaires, fini le code itératif entre deux ordres SQL... Désormais, avec le concept de CTE, toute extraction, aussi complexe soit-elle, n'a besoin que d'une seule requête !

## Ce qu'il manque à SQL Server

Tout serait parfait dans le meilleur des mondes s'il ne manquait pas quelques fonctionnalités à SQL Server. Il suffit de demander à la communauté des développeurs ce dont elle a besoin, pour en connaître les lacunes.

Parmi les plus cruciales, on relève l'absence des types SQL "DATE" et "TIME", l'impossibilité de déferer les contraintes (en fait les appliquer en fin de transaction plutôt qu'immédiatement), l'absence du "row value constructor" (il permet d'associer différentes colonnes afin de globaliser les expressions), la carence en prédicat de type "expressions régulières" (prédicat SIMILAR, pour des recherches textuelles avancées) et enfin l'absence de l'ordre MERGE (un combiné d'UPDATE et d'INSERT).

Tous ces manques font d'ailleurs partie de la norme SQL, mais aucun éditeur de SGBDR n'a encore réussi à tous les implémenter. Or Microsoft à travers son engagement envers la communauté des informaticiens est à l'écoute des demandes. La preuve, les types DATE et TIME était présent dans la bêta 1 de SQL Server 2005, mais l'implémentation qui avait été réalisée à ce stade était tellement mauvais que la version finale y a heureusement échappée !

## 5) Le renouveau de la micro informatique

Service Broker est un système de messagerie asynchrone permettant de développer des applications SQL Server 2005 en répartissant ses données et les traitements associés sur différentes bases dans différents serveurs, quelque soit la distance entre les serveurs (Internet).

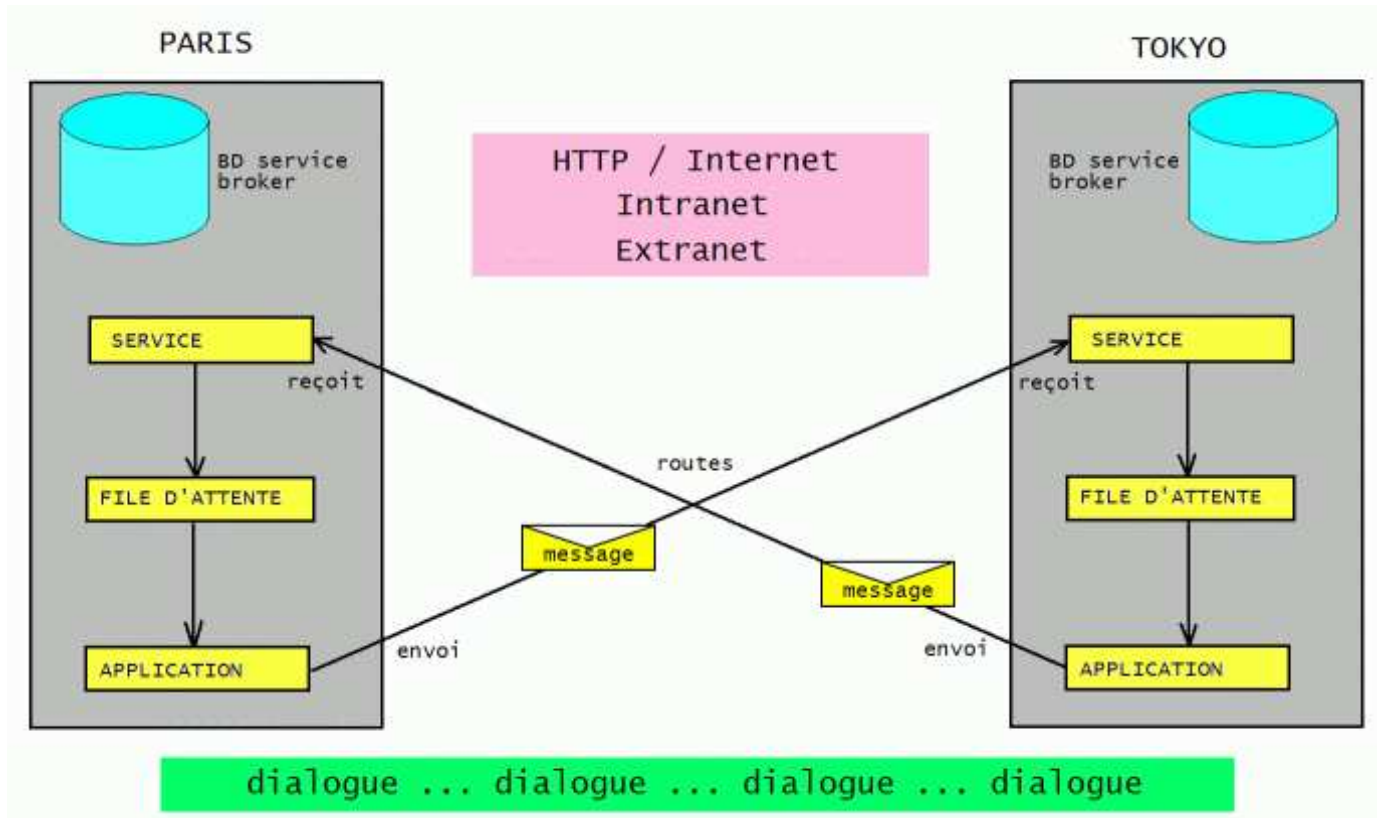
Si Microsoft en est arrivé à un tel système, c'est parce que l'on commence à comprendre l'importance du concept d'applications pilotées par les données, mais aussi que répartir ses données et ses traitements sur différents serveurs, met l'information plus proche de l'utilisateur. En quelques sortes, Microsoft entame avec Service Broker un renouveau de la micro informatique côté serveur...

Les systèmes de messageries informatiques asynchrones permettent de réaliser des applications collaboratives et réparties. Basées sur une logique applicatives, elles permettent de résoudre du même coup certains problèmes tel que la possibilité d'alerter lorsqu'il y a défaillance de la chaîne de traitement.

Service Broker propose donc un système de messagerie intégrée au serveur SQL : il permet d'élaborer des stratégies de communication d'information (véhiculée via le protocole http), sans jamais quitter l'univers de la base de données et cela de manière asynchrone et transactionné, ce qui permet d'obtenir un bon lissage de la charge.

Étudions maintenant quelques un des concepts du système : Les messages sont constitué par des informations binaires encapsulant tout type de données. Des dialogues sont initiés entre les applications qui les envoient, et les services qui les consomment. Ces dialogues peuvent être routés de manière spécifique et l'information peut être cryptée. Les messages sont reçus dans l'ordre de leur expédition et sont aiguillé dans les files d'attente (QUEUE). Ces files d'attente peuvent être à consommateurs multiples ce qui permet par exemple de paralléliser des traitements sur une batterie de serveurs.

On entre ainsi dans le domaine de la base de données répartie et cela quelque soit l'OS, la machine et la version de SQL Server 2005... En effet Microsoft a intégré Service Broker dans la version "run-time", SQL Server 2005 Express, libre et gratuite.



*Architecture de communication de Service Broker*

Afin de ne pas recevoir n'importe quoi, un service s'appuie sur des contrats qui définissent quels types de messages il consomme. Vous l'aurez sans doute compris, toute la cinématique de flux des messages est élaborée à l'aide de documents XML et les services sont basés sur le protocole SOAP. En fait les points de livraison des messages sont des "http endpoint" dont nous avons déjà parlé.


C'est avec cette technique que la SNCF a décidé de renouveler son back office des recettes "voyageurs" : Nefertiti permet de vérifier les sommes encaissées par les points de vente et prévoit la reconnaissance physique des versements, leur comptabilisation et remise en banque ainsi que la gestion de coffre. En fait les versements suivent deux chemins : le transfert de fonds (chemin physique) et les flux informatiques (chemin logique). Nefertiti assure le rapprochement entre flux physique et logique notamment à travers de périphériques de reconnaissance monétaires. On comprend bien qu'une telle application est doublement critique : la comptabilisation oblige au zéro défaut et les flux doivent opérer au plus vite car tout retard entraîne de coûteux délais de remise d'argent en banque. Et puis il y a la pression de la



clientèle au guichet... Chaque caisse principale enregistrant environ deux millions de transactions par jour.

Afin d'éviter l'indisponibilité du système central et donc l'impossibilité de servir le client, les flux seront enregistrés dans les files d'attente et envoyés de manière asynchrone au serveur central. En cas de coupure du réseau, la caisse est autonome... Un projet de 17 années hommes devant se terminer en 2007.

Les critères d'Aline Longchamp, chef de projet Paiement à la division Finances de la DSIT à la SNCF, pour ce choix : fiabilité, performances et haute disponibilité.



## SQLspot : un focus sur vos données !

---

SQLSPOT vous apporte les solutions dont vous avez besoin pour vos bases de données **Microsoft SQL Server**

### GAGNEZ DU TEMPS ET DE L'ARGENT


pour toutes vos problématiques Microsoft SQL server avec **Frédéric BROUARD**, expert SQL Server, enseignant aux Arts & Métiers et à l'Institut Supérieur d'Électronique et du Numérique (Toulon).

Tél. : **06 11 86 40 66**

*Interventions sur Nice, Aix, Marseille, Toulouse, Lyon, Nantes, Paris...*

SQLspot a été créée en mars 2007 à l'initiative de Frédéric Brouard, après trois ans d'activité sur le conseil en matière de SGBDR SQL Server, afin de proposer des services à valeur ajoutée à la problématique des données de l'entreprise :

- conseil (par exemple stratégie de gestion des données),
- modélisation de données (modèles conceptuels, logiques et physiques, rétro ingénierie...),
- qualification des données (validation, vérifications, reformatage automatique de données...),
- réalisation d'algorithmes de traitement de données (indexation textuelle avancée, gestion de méta modèles, traitements récursif de données arborescentes ou en graphe...),
- formation (aux concepts des SGBDR, au langage SQL, à la modélisation de données, à SQL Server ...)
- audit (audit de structure de base de données, de serveur de données, d'architecture de données...)
- tuning (affinage des paramètres OS, réseau et serveur pour une exploitation au mieux des ressources)
- optimisation (réécriture de requêtes, étude d'indexation, maintenance de données, refonte de code serveur...)



**Microsoft®  
Most Valuable  
Professional**

mail :  
[SQLpro@SQLspot.com](mailto:SQLpro@SQLspot.com)

*Vos données constituent le capital essentiel de votre système informatique. Pensez à les entretenir aussi bien que le reste...*

## Quelques sources pour compléter votre information.

### Références citées :

Wintercorp :

[http://www.wintercorp.com/VLDB/2005\\_TopTen\\_Survey/TopTenWinners\\_2005.asp](http://www.wintercorp.com/VLDB/2005_TopTen_Survey/TopTenWinners_2005.asp)  
édite chaque année le "top ten" des plus grosses bases de données.

OlapReport

<http://www.olapreport.com/market.htm>

un organisme de recherche indépendant sur les technologies OLAP

TPC :

<http://www.tpc.org/default.asp>

FEVAD (Fédération de la vente à distance) :

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

**Expériences citées :**

SAP :

<http://www.microsoft.com/france/events/event.aspx?EventID=118766832>

Koëlher :

[http://download.microsoft.com/documents/customerevidence/21470\\_Koehler\\_Final.doc](http://download.microsoft.com/documents/customerevidence/21470_Koehler_Final.doc)

Mediterranean Shipping Company :

[http://www.microsoft.com/france/temoignages/2005/info.asp?mar=/france/temoignages/2005/p\\_medship\\_1105.html](http://www.microsoft.com/france/temoignages/2005/info.asp?mar=/france/temoignages/2005/p_medship_1105.html)

Conforama :

[http://www.microsoft.com/france/temoignages/info.asp?mar=/france/temoignages/2004/p\\_conforama\\_0204.html](http://www.microsoft.com/france/temoignages/info.asp?mar=/france/temoignages/2004/p_conforama_0204.html)

Casino :

[http://www.microsoft.com/france/temoignages/2005/info.asp?mar=/france/temoignages/2005/p\\_casino\\_1105.html](http://www.microsoft.com/france/temoignages/2005/info.asp?mar=/france/temoignages/2005/p_casino_1105.html)

123 Multimedia :

[http://www.microsoft.com/france/temoignages/2005/info.asp?mar=/france/temoignages/2005/p\\_123Multimedia\\_1105.html](http://www.microsoft.com/france/temoignages/2005/info.asp?mar=/france/temoignages/2005/p_123Multimedia_1105.html)

**TECHNIQUES**

Quelques liens pour en savoir plus sur les techniques évoquées dans le présent document.

**Snapshot :**

<http://www.informit.com/articles/article.asp?p=357098&rl=1>

<http://www.informit.com/articles/article.asp?p=439602>

*Ravindra Okade consultant à New York à fait deux articles de fond sur ce sujet et compare la version Oracle et SQL Server du snapshot.*

<http://www.code-magazine.com/Article.aspx?quickid=0407071>

<http://www.code-magazine.com/article.aspx?quickid=0501061&page=1>

*Deux articles par Ron Talmage, un expert SQL Server spécialisé sur les problèmes de verrouillage et bien connu pour ses méthodes de repérage de deadlocks.*

**Mirroring :**

<http://www.databasejournal.com/features/mssql/article.php/3440511>

*Une petite introduction au mirroring.*

**Dynamic Management View et DMO (Dynamic Management Objects) :**

<http://www.sqlservercentral.com/columnists/chedgate/dynamicmanagementobjects.asp>

*Un aperçut rapide du concept*

<http://www.sqlskills.com/resources/Whitepapers/SQL%20Server%20DBA%20Guide%20to%20SQLCLR.htm>

*Une étude générale sur l'aide qu'apporte les Dynamic Management View sur l'administration .net/CLR des serveurs SQL, par Kimberly Tripp, une dynamique MVP SQL Server.*

**SSIS :**

<http://www.sqlservercentral.com/columnists/jthomson/thenewetlparadigm.asp>

<http://blogs.conchango.com/jamiethomson/>

*Jamie Thompson est MVP SQL Server et s'est spécialisée sur l'intégration de données dans les datawarehouses*

**XML dans SQL Server 2005 :**

<http://www.15seconds.com/issue/050803.htm>

*Une brillante et complète étude sur XML dans SQL Server par Alex Homer, MVP ASP .net.*

<http://www.dbazine.com/sql/sql-articles/cotter3/>

*Une autre étude par Hilary Cotter, un MVP SQL Server spécialiste de la réplication.*

**MARS :**

<http://blogs.msdn.com/angelsb/archive/2004/09/07/226597.aspx>

*Un jeu de question réponse sur le sujet.*

<http://dotnet.sys-con.com/read/45539.htm>

*Petite introduction à MARS.*

**http endpoint :**

<http://www.developer.com/db/article.php/3390241>

*Un article bien fait de Peter de Betta, auteur d'un des tout premiers livres sur SQL Server 2005.*

<http://www.eggheadcafe.com/articles/20040730.asp>

*Un exemple simple de service SOAP avec les http endpoint, par un MVP C#*

**Notification Services :**

<http://www.sqljunkies.com/Article/38D95C18-D0AB-4B00-9CF5-80940309E68C.scuk>

*Un aperçut de la chose.*

<http://www.devx.com/dbzone/Article/28522>

*Un article très complet avec un bon exemple.*

**SQL CLR :**

<http://www.developer.com/net/csharp/article.php/3399881>

*Un article d'introduction très complet*

<http://www.sqljunkies.com/Article/4CD01686-5178-490C-A90A-5AEEF5E35915.scuk>

*Un exemple d'utilisation de CLR : création d'un trigger générique d'audit (en VB hélas)*

<http://www.aspfree.com/c/a/MS-SQL-Server/NET-CLR-stored-procedures-in-SQL-Server-2005-DB-Good-News-to-Programmers-Shock-to-DBAs/1/>

*Utilisation de Visual Studio 2005 pour coder en SQL CLR.*

**Fonctions de fenêtrage :**

<http://www.sqljunkies.com/Article/4E65FA2D-F1FE-4C29-BF4F-543AB384AFBB.scuk>

*Un papier clair et simple sur le sujet*

<http://www.sqlservercentral.com/columnists/mcoles/sequentialordering.asp>

*Quelques exemples d'utilisation des fonctions d'ordonnancement*

**Clause OUTPUT :**

<http://www.dbazine.com/sql/sql-articles/cook18/>

*Aperçut rapide et complet avec exemples basiques, sur la clause OUTPUT*

**CTE :**

<http://www.sqlservercentral.com/columnists/fBROUARD/recursivequeriesinsql1999andsqlserver2005.asp>

*Mon étude sur le sujet (elle est parue en français dans SQL Server Magazine d'octobre 2005).*

**Manques :**

[http://blogs.msdn.com/sqltips/archive/2005/10/07/Top\\_5\\_Features\\_Request.aspx](http://blogs.msdn.com/sqltips/archive/2005/10/07/Top_5_Features_Request.aspx)

[http://umachandar.members.winisp.net/files/top\\_5\\_features\\_summary.htm](http://umachandar.members.winisp.net/files/top_5_features_summary.htm)

*Les manques de SQL servers, réclamés par la communauté des développeurs.*

**Service Broker :**

<http://www.bedotnet.com/Default.aspx?PageContentID=37&tabid=94>

<http://www.guss.fr/DesktopModules/Articles/ArticlesView.aspx?tabID=0&ItemID=191&mid=2889>

*Deux petits exemples de mise en oeuvre de Service Broker*

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

*Le site de Service Broker*

**Autres nouveautés de SQL Server 2005**

[http://sqlpro.developpez.com/SQL\\_Server\\_2K5/N1.php](http://sqlpro.developpez.com/SQL_Server_2K5/N1.php)

[http://sqlpro.developpez.com/SQL\\_Server\\_2K5/N2.php](http://sqlpro.developpez.com/SQL_Server_2K5/N2.php)

*Une série d'article que je dois finaliser sur l'essentiel des nouveautés de MS SQL Server 2005.*