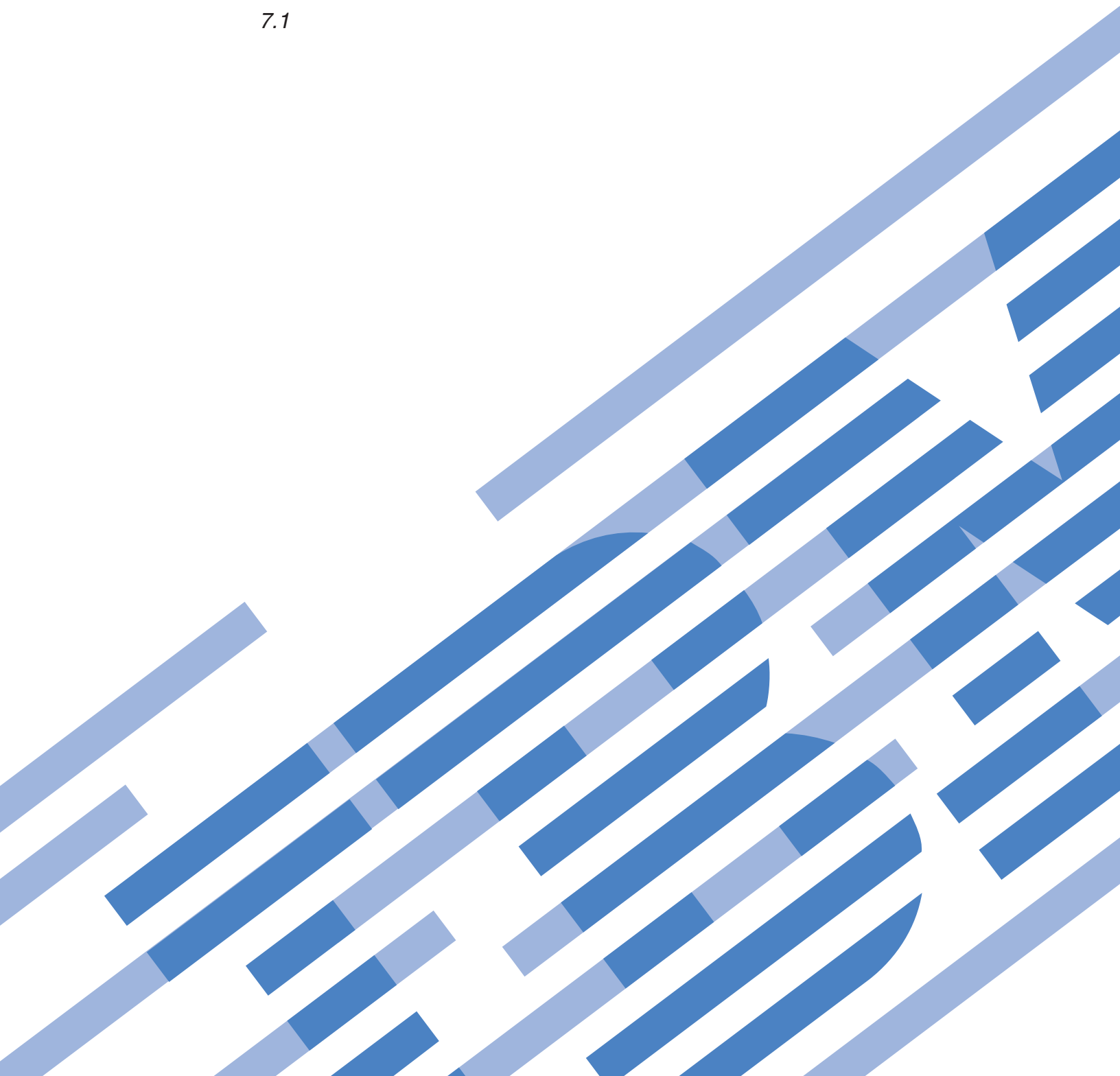




IBM i

OmniFind Text Search Server for DB2 for i (serveur de recherche de texte pour le e-commerce et les services Web)

7.1





IBM i

OmniFind Text Search Server for DB2 for i (serveur de recherche de texte pour le e-commerce et les services Web)

7.1

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 113.

Première édition - février 2010

Réf. US : RZAS-H000-00

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
17 avenue de l'Europe
92275 Bois-Colombes Cedex*

© Copyright IBM France 2009. Tous droits réservés.

© Copyright International Business Machines Corporation 2002, 2010.

Table des matières

OmniFind Text Search Server for DB2 for i	1	Recherche XML	48
Nouveautés d'IBM i 7.1.	1	Administration d'un serveur de recherche de texte	
Fichier PDF d'OmniFind Text Search Server for DB2 for i	1	OmniFind(r) pour DB2(r) for i	65
Présentation du serveur de recherche de texte		Démarrage de OmniFind Text Search Server for DB2 for i	65
OmniFind for DB2 for i.	2	Arrêt de OmniFind Text Search Server for DB2 for i	65
Présentation générale d'OmniFind Text Search Server for DB2 for i	2	Sauvegarde et restauration des index de recherche de texte	66
Configuration système requise pour l'installation d'OmniFind Text Search Server for DB2 for i	4	Identification des problèmes	69
Concepts clés	4	Affichage et enregistrement des journaux serveur	70
Création et mise à jour d'un index de recherche de texte	4	Outils d'administration	71
Indexation asynchrone et déclencheurs	5	Utilitaire ServerInstance	81
Formats de document pris en charge	6	Remarques relatives aux ASP indépendants pour OmniFind Text Search Server for DB2 for i	83
Types de données pris en charge.	7	Accessibilité avancée	84
Prise en charge des synonymes et notation de texte	8	Analyse de la performance	85
Traitement linguistique	8	Remarques relatives aux transactions	88
Nom d'alias de serveur	11	Utilisation d'IBM Systems Director Navigator for i	89
Installation et configuration des fonctions de recherche de texte	12	Utilisation de System i Navigator	95
Installation d'OmniFind Text Search Server for DB2 for i	12	Tables d'administration de recherche de texte.	102
Lancement des fonctions de recherche de texte	13	Table d'administration	
Création d'un index de recherche de texte	14	QSYS2.SYSTEXTDEFAULTS	102
Mise à jour d'un index de recherche de texte	14	Table d'administration	
Recherche d'un index de recherche de texte	15	QSYS2.SYSTEXTINDEXES	103
Troncature des documents	15	Table d'administration	
Procédures stockées d'administration pour la recherche de texte	15	QSYS2.SYSTEXTCOLUMNS	105
SYSPROC.SYSTS_START	16	Table d'administration	
SYSPROC.SYSTS_STOP	17	QSYS2.SYSTEXTSERVERS	106
SYSPROC.SYSTS_CREATE	19	Table d'administration	
SYSPROC.SYSTS_ALTER	28	QSYS2.SYSTEXTCONFIGURATION	107
SYSPROC.SYSTS_DROP	33	Table d'administration	
SYSPROC.SYSTS_UPDATE	35	QSYS2.SYSTEXTSERVERHISTORY	107
Recherche avec un index de recherche de texte	38	Messages et codes	108
CONTAINS	38	Messages d'OmniFind	108
SCORE	41		
Syntaxe des arguments de recherche	43		
		Annexe. Remarques	113
		Documentation sur l'interface de programmation	115
		Marques	115
		Dispositions	115

OmniFind Text Search Server for DB2 for i

OmniFind Text Search Server for DB2 for i vous permet d'exécuter des instructions SQL semblables à des requêtes de recherche de texte standard sur des documents stockés dans une base de données DB2.

Remarque : L'utilisation des exemples de codes implique que vous acceptez les termes de la rubrique «Licence du code et informations de limitation de responsabilité», à la page 111.

Nouveautés d'IBM i 7.1



Découvrez les nouveautés ou les modifications apportées à l'ensemble de rubriques OmniFind Text Search Server for DB2 for i.

Les principales nouveautés sont les suivantes :

- Amélioration de «Recherche XML», à la page 48
 - «Utilisation des espaces de noms dans les recherches», à la page 56
 - «Comparaisons de valeurs numériques», à la page 51
 - «Comparaisons de dates et d'horodates», à la page 52
- Index de recherche de texte pour les colonnes XML DB2 : «Exemple de recherche XML», à la page 60
- «Sauvegarde et restauration d'un index de recherche de texte sans données», à la page 66
- «Nom d'alias de serveur», à la page 11
- Deux interfaces utilisateur graphiques :
 - «Utilisation d'IBM Systems Director Navigator for i», à la page 89
 - «Utilisation de System i Navigator», à la page 95
- «Accessibilité avancée», à la page 84
- «Analyse de la performance», à la page 85
- «Remarques relatives aux transactions», à la page 88

Identification des nouveautés et des changements

Pour vous aider à localiser les modifications techniques, l'Information Center utilise les méthodes suivantes :

- Le symbole , qui indique le début des informations nouvelles ou modifiées
- Le symbole , qui indique la fin des informations nouvelles ou modifiées

Dans les fichiers PDF, vous verrez des barres de révision (|) dans la marge de gauche des informations nouvelles ou modifiées.

Pour plus d'informations sur les nouveautés et les modifications de cette édition, voir la Note aux utilisateurs.



Fichier PDF d'OmniFind Text Search Server for DB2 for i

Utilisez cette page pour afficher et imprimer le fichier PDF de ce document.

Pour visualiser ou télécharger le fichier PDF de ce document, sélectionnez OmniFind Text Search Server for DB2 for i (environ 1192 Ko).

| Informations complémentaires

| Vous pouvez visualiser ou imprimer les fichiers PDF suivants :

- | • Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS 
- | • SQL Performance Diagnosis on IBM® DB2 Universal Database for iSeries 
- | .

| Enregistrement des fichiers PDF

| Pour sauvegarder un fichier PDF sur votre poste de travail afin de l'afficher ou de l'imprimer, procédez comme suit :

- | 1. Cliquez avec le bouton droit sur le fichier PDF dans votre navigateur (cliquez avec le bouton droit de la souris sur le lien précédent).
- | 2. Cliquez sur l'option de sauvegarde locale du fichier PDF.
- | 3. Accédez au répertoire dans lequel vous souhaitez sauvegarder le PDF.
- | 4. Cliquez sur **Enregistrer**.

| Téléchargement d'Adobe® Reader

| Adobe Reader doit être installé sur votre système pour que vous puissiez afficher ou imprimer ces fichiers PDF. Vous pouvez en télécharger un exemplaire gratuit à partir du site Web d'Adobe

| (<http://get.adobe.com/reader/>) .

| Présentation du serveur de recherche de texte OmniFind for DB2 for i

| DB2 for i utilise le serveur de recherche de texte OmniFind(r) comme moteur de recherche et d'indexation pour les documents stockés dans une base de données DB2.

| OMNIFIND est une solution de recherche de texte qui permet aux utilisateurs d'IBM i de rechercher des texte non structurés dans une colonne d'une table DB2 for IBM i. Le texte stocké dans une colonne peut être un texte composé de caractères simples, un document XML ou différents types de textes enrichis, par exemple un fichier PDF ou DOC. Le programme permet d'indexer des données non structurées sans analyser leur syntaxe dans un format structuré tel qu'une table SQL.

| Le serveur de recherche de texte OmniFind est un moteur de recherche basé sur le contexte. Il prend en charge des fonctions de recherche avec correspondance partielle. Par exemple, une recherche sur le terme "cheval" peut isoler des documents contenant le terme "cheval" ou "chevaux". Le moteur de recherche prend également en charge le contexte linguistique. Par exemple, il peut comprendre les équivalences partielles de "cheval" et "chevaux" à la fois en français et en anglais.

| Les applications qui peuvent bénéficier de ces capacités sont nombreuses. Prenons l'exemple d'une base de données de gestion des ressources humaines. Les descriptifs des candidats peuvent être stockés dans une base de données dans n'importe quel format. Vous pouvez ensuite utiliser OmniFind pour y rechercher les candidats possédant certaines compétences.

| Présentation générale d'OmniFind Text Search Server for DB2 for i

| OmniFind Text Search Server for DB2 for i fournit un ensemble de procédures stockées d'administration et deux fonctions intégrées : CONTAINS et SCORE. Ces fonctions permettent de rechercher des index de recherche de texte créés à partir des documents stockés dans une table de base de données DB2. Les procédures stockées d'administration permettent d'activer et de désactiver la recherche de texte et de créer, mettre à jour et supprimer des index de texte.

- | Vous pouvez créer un index de texte sur n'importe quelle colonne contenant les types de données suivants :
 - | • CHAR
 - | • VARCHAR
 - | • CLOB
 - | • BLOB
 - | • DBCLOB
 - | • GRAPHIC
 - | • VARGRAPHIC
 - | • BINARY
 - | • VARBINARY
 - | • XML
- | Les données peuvent contenir du texte en clair, du code HTML, XML, ou d'autres types de documents enrichis comme des fichiers PDF. Les données sont lues dans les colonnes de texte puis converties au format Unicode (CCSID 1208) avant d'être indexées.
- | Les index de texte ne sont pas comme les index traditionnels de DB2. Leur maintenance n'est pas automatique, ils ne sont pas journalisés et vous ne pouvez pas les sauvegarder avec les méthodes classiques de sauvegarde et de restauration. Les index de texte sont créés et stockés sur un serveur de recherche de texte.
- | Par défaut, le serveur de recherche de texte est créé sur le même système que les données stockées dans la base de données DB2. Toutefois, vous pouvez créer un serveur de recherche de texte sur un autre serveur exécutant IBM i, Linux®, UNIX®, AIX ou Windows®.
- | Le serveur de recherche de texte contient une collection de termes significatifs de chaque ligne de la colonne. Une connexion TCP/IP est utilisée pour communiquer avec le serveur de recherche de texte.
- | Les fonctions CONTAINS et SCORE sont des fonctions intégrées dans DB2 for i.
- | DB2 for i utilise le serveur de recherche de texte OmniFind(r) comme moteur de recherche et d'indexation pour les documents stockés dans une base de données DB2.
- | OMNIFIND prend en charge plusieurs collections. Une collection contient un index de recherche de texte et ses options spécifiques pour l'analyse syntaxique, l'indexation et la recherche.
- | OMNIFIND possède une interface utilisateur graphique servant à l'administration des serveurs et des index de texte.
- | Le serveur de recherche de texte fournit également des procédures stockées SQL et des outils de ligne de commande que vous pouvez utiliser pour exécuter des tâches communes comme, par exemple, configurer et administrer le serveur de recherche de texte, créer un dictionnaire de synonymes pour une collection et effectuer un diagnostic des incidents.

| Concepts associés


| «Procédures stockées d'administration pour la recherche de texte», à la page 15
| Vous pouvez démarrer et arrêter des fonctions de recherche de texte et créer, supprimer ou mettre à jour des index de recherche de texte à l'aide d'un ensemble de procédures stockées d'administration en SQL.
| Vous pouvez appeler ces procédures depuis n'importe quelle interface SQL. Vous ne pouvez pas appeler ces procédures à partir d'une ligne de commande IBM i avec des commandes CL.

| Référence associée

| «CONTAINS», à la page 38
| Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.
| «SCORE», à la page 41
| Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

| Configuration système requise pour l'installation d'OmniFind Text Search Server for DB2 for i

| Avant d'installer OmniFind Text Search Server for DB2 for i, vérifiez que votre système comporte tous les matériels, logiciels et système d'exploitation requis.

| Quand vous installez OmniFind Text Search Server for DB2 for i, le programme d'installation crée un serveur de recherche de texte pour IBM i. Vous pouvez installer d'autres serveurs de recherche de texte sur des serveurs distants exécutant Linux ou Windows. Ces serveurs font partie du produit DB2 Accessories Suite for z/OS (5655-R14) (<http://www-01.ibm.com/software/data/db2imstools/db2tools/accessories-suite/>) . Le lien fournit des informations sur le téléchargement de la suite logicielle.

| Configuration logicielle requise

| Vérifiez que votre système possède la configuration logicielle minimum requise suivante :

- | • IBM Java™ Runtime 1.5 SR4
- | • Pilote DB2 Universal Java Driver installé et configuré sur le serveur de recherche de texte
- | • Pour IBM i, les programmes suivants doivent être installés :
 - | – 5770SS1 Option 30 Qshell
 - | – 5770SS1 Option 33 IBM i Portable Application Solutions Environment (IBM i PASE)
 - | – 5770SS1 Option 39 International Components for Unicode
 - | – Le dernier groupe de correctifs de programme provisoire (PTF) pour IBM DB2 for i est appliqué au système.

| Concepts clés

| Il est important de comprendre les concepts clés des fonctions de recherche de texte pour tirer profit des fonctionnalités de OmniFind Text Search for DB2 for i. Ces concepts clés concernent les types de documents et les langues prises en charge.

| Création et mise à jour d'un index de recherche de texte

| Pour créer un index de recherche de texte, vous pouvez définir et déclarer les propriétés de l'index. Pour mettre à jour un index de recherche de texte, vous pouvez ajouter de nouvelles données à l'index à partir d'une table DB2. Vous pouvez également mettre à jour un index de recherche de texte en modifiant les données dans l'index.

| Pour chaque index de recherche de texte que vous créez, une collection est créée dans OmniFind Text Search Server for DB2 for i. L'index de recherche de texte ne contient pas de données après sa création.

| Pour ajouter des données à l'index de recherche de texte, vous devez appeler la procédure stockée SYSPROC.SYSTS_UPDATE. Le premier processus de mise à jour ajoute tous les documents texte depuis la colonne de texte vers l'index de recherche de texte. Ce processus porte le nom de *mise à jour initiale*. Les mises à jour suivantes sont incrémentielles.

| Quand vous créez un index de recherche de texte, les objets suivants sont créés ou mis à jour :

- | • La table de transfert est créée dans la bibliothèque QSYS2.
- | • Les déclencheurs INSERT, DELETE et UPDATE sont ajoutés à la table de base.
- | • Une vue représentant l'index de recherche de texte est créée dans le schéma de l'index de recherche de texte. Cette vue contient des informations sur l'index de texte. Par exemple, vous pouvez utiliser cette vue pour obtenir le nom de la table de base et celui de la table de transfert. La vue indique également le nombre de modifications en attente dans la table de base qui n'ont pas encore été répercutées dans l'index de recherche de texte.
- | • Les catalogues d'index de recherche de texte (SYSTEXTINDEXES et SYSTEXTCOLUMNS) contenus dans la bibliothèque QSYS2 sont mis à jour avec une nouvelle entrée correspondant au nouvel index de recherche de texte.

| Remarques sur la table de transfert :

- | • N'exécutez aucune opération de base de données sur la table de transfert en dehors de la sauvegarde ou de la restauration du fichier, ou de la modification des droits d'accès.
- | • Si vous modifiez les droits d'accès de la table de base, changez aussi ceux de la table de transfert.

| Remarques sur la table de base :

- | • Ne supprimez pas les déclencheurs DELETE, UPDATE et INSERT qui sont ajoutés lors de la création d'un index de recherche de texte.
- | • La suppression de l'index de recherche de texte a pour effet de supprimer les déclencheurs.
- | • Ne modifiez ou ne supprimez pas le ROWID, la clé primaire ou la colonne unique utilisée comme clé dans l'index de recherche de texte.
- | • La modification de la colonne de données de la table de base qui entraîne la troncature des données peut occasionner des occurrences positives erronées dans l'index de recherche de texte.

| **Référence associée**

| «SYSPROC.SYSTS_UPDATE», à la page 35

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

| «Table d'administration QSYS2.SYSTEXTINDEXES», à la page 103

| La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de schéma et un nom de collection associée sur le serveur de recherche de texte.

| «Table d'administration QSYS2.SYSTEXTCOLUMNS», à la page 105

| La table d'administration QSYS2.SYSTEXTCOLUMNS vous permet de visualiser des informations sur les colonnes de texte d'un index de recherche de texte. Chaque index de recherche de texte contient un ID d'index, des noms de colonne de texte et le nom de schéma de la table de base.

| **Indexation asynchrone et déclencheurs**

| Vous pouvez mettre à jour l'index de recherche de texte dans OmniFind Text Search Server for DB2 for i manuellement ou planifier cette mise à jour de manière qu'elle s'exécute automatiquement.

| L'index de recherche de texte géré dans OmniFind Text Search Server for DB2 for i n'est pas mis à jour de manière synchrone quand la table DB2 est mise à jour. La mise à jour d'un index de recherche de texte est une opération extensive.

| Les modifications de la colonne de table DB2 sont capturées dans une table de journal locale par des déclencheurs. Cette table de journal est également appelée une table de transfert. Les déclencheurs stockent automatiquement dans une table de journal les informations relatives aux documents nouveaux, modifiés et supprimés. Chaque table de journal est associée à un index de recherche de texte. L'application du contenu d'une table de journal à l'index de recherche de texte correspondant est appelée une *mise à jour incrémentielle*.

| Vous devez périodiquement mettre à jour l'index de recherche de texte pour que les modifications soient prises en compte dans les requêtes.

| Pour mettre à jour manuellement l'index de recherche de texte, vous devez appeler la procédure stockée SYSPROC.SYSTS_UPDATE.

| Vous pouvez aussi planifier les mises à jour de manière qu'elles s'exécutent automatiquement, à l'aide de la clause UPDATE FREQUENCY dans la procédure SYSPROC.SYSTS_CREATE, au moment de la création de l'index de texte.

| **Référence associée**

| «SYSPROC.SYSTS_CREATE», à la page 19

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte.

| Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

| «SYSPROC.SYSTS_UPDATE», à la page 35

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

| **Formats de document pris en charge**

| Les données d'une colonne de texte peuvent comporter du texte en clair, un document HTML, un document XML, ou tout autre document reconnu par le moteur de recherche.

| OmniFind Text Search Server for DB2 for i analyse la syntaxe des documents afin d'extraire les parties pertinentes et les rendre interrogeables. Par exemple, les balises et les métadonnées d'un document HTML ne sont pas indexées.

| L'analyse syntaxique des formats de document suivants est prise en charge :

- | • TEXT : texte à plat
- | • HTML : Hypertext Markup Language
- | • XML : Extensible Markup Language
- | • INSO : OmniFind Text Search Server for DB2 for i utilise des filtres pour détecter le format des documents texte. Les formats de document INSO pris en charge sont les suivants :
 - | – XML
 - | – HTML
 - | – JustSystems Ichitaro
 - | – Lotus 123
 - | – Lotus Freelance
 - | – Lotus WordPro
 - | – Microsoft® Excel
 - | – Microsoft PowerPoint

- | – Microsoft Rich Text Format
- | – Microsoft Visio
- | – Microsoft Word
- | – Microsoft Write
- | – PDF (Portable Document Format)
- | – Quattro Pro
- | – Rich Text RTF
- | – StarOffice Calc et OpenOffice Calc

| Tous les documents d'une colonne de texte indexée doivent avoir le même format (TEXT, HTML, XML ou INSO).

| **Données XML**

| La structure XML des données XML est indexée dans OmniFind Text Search Server for DB2 for i après l'analyse syntaxique des données avec un analyseur syntaxique XML. Vous pouvez ensuite utiliser la syntaxe des requêtes de recherche XML pour récupérer les résultats.

| **Concepts associés**

| «Recherche XML», à la page 48

| Vous pouvez indexer et rechercher des documents XML. La syntaxe des recherches XML utilise un sous-ensemble du langage W3 XPath avec des extensions développées pour la recherche de texte. Ces extensions permettent de rechercher des plages de valeurs numériques, de dates et d'horodates associées à un élément ou un attribut XML. Les éléments structurels peuvent s'utiliser séparément ou de manière combinée avec du texte libre dans les requêtes.

| **Types de données pris en charge**

| Les données des colonnes de texte que vous voulez indexer et interroger peuvent être des données binaires ou des données de type caractères.

| Les types de données suivants correspondent à des données binaires :

- | • BINARY
- | • VARBINARY
- | • BLOB

| De plus, OMNIFIND gère les types de données suivants de la même manière que les données binaires :

- | • CHAR FOR BIT DATA
- | • VARCHAR FOR BIT DATA

| Les types de données suivants correspondent à des données de type caractères :

- | • CHAR FOR SBCS DATA ou FOR MIXED DATA
- | • VARCHAR FOR SBCS DATA ou FOR MIXED DATA
- | • CLOB
- | • DBCLOB
- | • GRAPHIC
- | • VARGRAPHIC
- | • XML

| Si les données sont des données binaires, vous pouvez indiquer l'ID de jeu de caractères codés (CCSID) utilisé pour construire l'index de recherche de texte. Pour les données de type caractères, la base de données DB2 reconnaît le codage et, si vous indiquez explicitement un CCSID, cette spécification est ignorée.

| **Prise en charge des synonymes et notation de texte**

| Vous pouvez utiliser des synonymes pour améliorer les résultats d'une requête. Vous pouvez également appliquer une notation de texte pour mesurer à quel degré un résultat concorde avec une requête.

| **Notation de texte**

| La notation de texte est calculée pendant la recherche. Elle peut être incluse dans les résultats de la requête. Une notation de texte est une valeur comprise entre 0 et 1 comptant jusqu'à trois décimales (de 0,000 à 1,000). La notation de texte indique à quel degré un résultat concorde avec une requête par rapport aux autres documents référencés dans l'index de recherche de texte.

| OMNIFIND génère la notation de texte à partir de plusieurs facteurs tels que l'importance générale des termes recherchés et la proximité des occurrences de ces termes. L'importance générale repose sur la fréquence des termes dans chaque document et sur la fréquence des mêmes termes dans l'ensemble des documents.

| **Prise en charge des synonymes**

| L'outil OmniFind Text Search Server for DB2 for i permet d'utiliser des synonymes afin de modifier les résultats de la requête. L'utilisation de synonymes peut accroître le nombre de résultats en augmentant le nombre de documents correspondant à la requête. Toutefois, l'utilisation de synonymes peut également diminuer la précision d'une requête ; il sera plus difficile d'isoler les documents qui correspondent exactement aux critères de recherche.

| Par défaut, les synonymes ne sont pas utilisés dans les requêtes. Pour utiliser des synonymes, vous devez créer un dictionnaire de synonymes puis l'ajouter à une collection avec l'utilitaire Synonym.

| Pour plus d'informations sur les synonymes, voir «Dictionnaires de synonymes», à la page 77.

| **Référence associée**

| «SCORE», à la page 41

| Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

| **Traitement linguistique**

| OmniFind Text Search Server for DB2 for i contient des modules de dictionnaires qui prennent en charge le traitement linguistique des documents et des requêtes qui ne sont pas en anglais.

| Au lieu d'une segmentation des mots basée sur un dictionnaire, OmniFind Text Search Server for DB2 for i utilise la *segmentation n-gram* pour les langues telles que le chinois, le japonais et le coréen. La *segmentation n-gram* est une méthode d'analyse qui considère les répétitions de suites de caractères comme constituant un même mot. Comparativement, la *segmentation par blanc* basée sur Unicode utilise les blancs pour délimiter les mots.

| Si un document texte utilise l'une des langues prises en charge, le traitement linguistique est appliqué quand la syntaxe du texte est analysée et convertie en jetons. Pour les langues non prises en charge, un code d'erreur est renvoyé.

Quand vous analysez un index de recherche de texte, le moteur de recherche indique une occurrence qui contient des variantes linguistique des termes de la requête. Les variantes du mot dépendent de la langue de la requête.

Langues prises en charge

Vous pouvez demander que les documents texte soient traités dans une langue spécifique.

Vous pouvez indiquer la langue des données de texte indexées dans la procédure stockée d'administration SYSPROC.SYSTS_CREATE. Si vous entrez la valeur AUTO, OmniFind Text Search Server for DB2 for i essaie de déterminer la langue de manière automatique. Pour les documents courts, la détection automatique peut être imprécise et est donc déconseillée. La langue utilisée par défaut pour le traitement linguistique est l'anglais (en_US).

Le tableau suivant indique les codes de langue (cinq caractères) des langues prises en charge.

Tableau 1. Codes de langue des langues prises en charge

Code de langue	Langue
ar_AA	Arabe
cs_CZ	Tchèque
da_DK	Danois
de_CH	Allemand (Suisse)
de_DE	Allemand (Allemagne)
el_GR	Grec
en_AU	Anglais (Australie)
en_GB	Anglais (Royaume Uni)
en_US	Anglais (Etats Unis)
es_ES	Espagnol (Espagne)
fi_FI	Finnois
fr_CA	Français (Canada)
fr_FR	Français (France)
it_IT	Italien
ja_JP	Japonais
ko_KR	Coréen
nb_NO	Norvégien (Bokmal)
nl_NL	Néerlandais
nn_NO	Norvégien (Nynorsk)
pl_PL	Polonais
pt_BR	Portugais (Brésil)
pt_PT	Portugais (Portugal)
ru_RU	Russe
sv_SE	Suédois
zh_CN	Chinois simplifié
zh_TW	Chinois traditionnel

| **Référence associée**

- | «SYSPROC.SYSTS_CREATE», à la page 19
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte.
- | Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

| **Traitement linguistique pour les documents en chinois, japonais et coréen**

- | Vous pouvez traiter des documents rédigés en chinois, en japonais ou en coréen à l'aide d'une segmentation basée sur dictionnaire ou à l'aide de la segmentation n-gram.
- | L'obtention de résultats pertinents avec un moteur de recherche dépend largement des techniques appliquées au traitement du texte. Une fois le texte extrait du document, la première étape de son traitement consiste à identifier de manière individuelle les mots contenus dans le texte. Cette identification individuelle des mots du texte est appelée la *segmentation*. Pour plusieurs langues, les limites des mots peuvent être identifiées grâce aux espaces (blancs, fins de ligne et certains signes de ponctuation). En revanche, le chinois, le japonais et le coréen n'utilisent pas d'espace entre les caractères pour séparer les mots, aussi d'autres techniques sont nécessaires pour ces langues.
- | OmniFind Text Search Server for DB2 for i fournit les deux méthodes suivantes pour prendre en charge le traitement linguistique du chinois, du japonais et du coréen :
 - Segmentation des mots basée sur dictionnaire (également appelée analyse morphologique)
 - Segmentation n-gram

| **Segmentation des mots basée sur dictionnaire**

- | La *segmentation des mots basée sur dictionnaire* utilise des dictionnaires spécifiques à certaines langues pour identifier les mots présents dans une suite de caractères dans le document. Cette technique fournit des résultats de recherche précis car elle utilise des dictionnaires pour identifier les limites des mots.
- | Toutefois, la segmentation des mots basée sur dictionnaire peut laisser échapper certaines occurrences.

| **Segmentation n-gram**

- | La *segmentation n-gram* n'identifie pas les limites des mots mais indexe les répétitions de paires de caractères. Dans la mesure où OmniFind Text Search Server for DB2 for i emploie deux caractères, cette technique est également appelée la "segmentation bi-gram".
- | La segmentation n-gram renvoie tous les documents qui contiennent les termes de la recherche mais cette technique renvoie parfois des documents qui ne répondent pas à la requête.
- | Par défaut, OmniFind Text Search Server for DB2 for i est fourni avec un index préconfiguré qui utilise la segmentation n-gram pour le chinois, le japonais et le coréen.
- | Pour voir comment fonctionnent les deux types de traitement linguistique, examinons le texte du document suivant : Election for governor of Kanagawa prefecture. En japonais, ce texte contient huit caractères. Dans cet exemple, les huit caractères sont représentés par A B C D E F G H. Un utilisateur peut saisir la requête election for governor, qui demande quatre caractères en japonais, représentés par E F G H. (Le texte du document et l'exemple de requête utilisent les mêmes caractères.)

| **Si vous utilisez la segmentation n-gram :**

- | Une fois le document indexé, le moteur de recherche segmente le texte election for governor of Kanagawa prefecture et le représente par les paires de caractères suivantes : AB BC CD DE EF FG GH

Le modèle de requête election for governor est segmenté et représenté par les paires de caractères suivantes : DE EF FG GH. Si vous exécutez une recherche avec le modèle de requête election for governor, le document est isolé car les jetons du texte du document comme ceux de la requête apparaissent dans le même ordre.

Quand vous activez la segmentation n-gram, vous pouvez voir des résultats supplémentaires mais moins précis. Par exemple, en japonais, si vous saisissez la requête Kyoto et qu'un document présent dans l'index contient le texte City of Tokyo, le document est isolé car City of Tokyo et Kyoto ont en commun deux caractères japonais identiques.

Si vous n'utilisez pas la segmentation n-gram :

Une fois le document indexé, le moteur de recherche segmente le texte election for governor of Kanagawa prefecture et le représente par les séquences de caractères suivantes : ABC DEF GH

Le modèle de requête election for governor est segmenté et représenté par les paires de caractères suivantes : EF GH. Les caractères EF n'apparaissent pas dans les jetons du texte du document. (Le document ne contient pas EF mais il contient DEF).

Le texte du document contient DEF mais la requête contient uniquement EF. Le document a moins de chance d'être isolé avec le modèle de requête.

Quand vous n'activez pas la segmentation n-gram, vous obtenez moins de résultats mais ils sont plus précis.

Nom d'alias de serveur

Vous pouvez utiliser un nom d'alias de serveur pour affecter un nom significatif à un serveur.

Chaque serveur de recherche de texte est identifié individuellement par la colonne SERVERID dans le catalogue QSYS2.SYSTEXTSERVERS. La colonne SERVERID contient un entier généré par incrément par la base de données.

La colonne ALIASNAME du catalogue QSYS2.SYSTEXTSERVERS contient un nom d'alias significatif à affecter à chaque serveur. Les noms d'alias de serveur peuvent s'utiliser pour référencer des serveurs dans les procédures SYSTS_START, SYSTS_STOP et SYSTS_CREATE.

Remarque : Vous pouvez changer le nom d'alias de serveur directement dans la table du catalogue QSYS2.SYSTEXTSERVERS en modifiant la valeur de la colonne ALIASNAME.

Affectez au serveur numéro 1 le nom d'alias "PRIMARY_LOCAL_SERVER".

```
UPDATE QSYS2.SYSTEXTSERVERS
SET ALIASNAME = 'PRIMARY_LOCAL_SERVER'
WHERE SERVERID = 1
```

Pour supprimer un nom d'alias d'un serveur, affectez à la colonne la valeur NULL.

Supprimez un nom d'alias du serveur numéro 1.

```
UPDATE QSYS2.SYSTEXTSERVERS
SET ALIASNAME = NULL
WHERE SERVERID = 1
```

Référence associée

«Table d'administration QSYS2.SYSTEXTSERVERS», à la page 106

La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.

«SYSPROC.SYSTS_START», à la page 16

Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.

«SYSPROC.SYSTS_STOP», à la page 17

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_STOP pour arrêter les fonctions de recherche de texte de DB2. Cette procédure stockée affecte la valeur 1 (arrêté) au paramètre SERVERSTATUS dans le catalogue QSYS2.SYSTEXTSERVERS.

«SYSPROC.SYSTS_CREATE», à la page 19

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte.

Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

Installation et configuration des fonctions de recherche de texte

Vous pouvez installer et configurer OmniFind Text Search Server for DB2 for i. Vous pouvez également créer et mettre à jour un index de recherche de texte afin d'utiliser les fonctions de recherche de texte sur une colonne d'une table.

Installation d'OmniFind Text Search Server for DB2 for i

Vous pouvez installer OmniFind Text Search Server for DB2 for i à l'aide des procédures d'installation standard applicables aux programmes sous licence installables sur IBM i. Un serveur de texte par défaut est créé et la table QSYS2.SYSTEXTSERVERS est remplie avec des données de serveur par défaut. Facultativement, vous pouvez ajouter d'autres serveurs de recherche de texte après l'installation.

Remplissage de la table QSYS2.SYSTEXTSERVERS

OmniFind Text Search Server for DB2 for i est le programme sous licence 5733-OMF IBM. Pour plus d'informations sur l'installation d'un programme sous licence, voir Installation de programmes sous licence supplémentaires. Pour rechercher ce produit, entrez GO LICPGM dans la ligne de commande et sélectionnez l'option 10 (Afficher les programmes sous licence installés). Elle apparaît sous la liste des programmes sous licence.

La table QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte IBM OmniFind installés et disponibles pour DB2 for i. Lors de l'installation initiale du produit OmniFind Text Search Server for DB2 for i, un serveur de texte par défaut est créé sur le système IBM i. La QSYS.SYSTEXTSERVERS est également remplie avec des données de serveur par défaut.

Création de serveurs de recherche de texte supplémentaires

Si vous utilisez des serveurs de recherche de texte sur un système IBM distant, ou que vous utilisez des serveurs autres que des serveurs IBM, par exemple des serveurs Windows ou Linux remplissez explicitement cette table en exécutant une instruction SQL INSERT.

Si vous voulez ajouter d'autres données de serveur dans la table QSYS2.SYSTEXTSERVERS, procédez comme suit pour entrer l'instruction SQL INSERT :

1. Indiquez le numéro de port de serveur et le nom de serveur de chaque serveur de recherche de texte dans la colonne SERVERPORT et les colonnes SERVERNAME de la table QSYS2.SYSTEXTSERVERS en entrant une instruction SQL INSERT.

2. Indiquez le jeton d'authentification de chaque serveur de recherche de texte dans la colonne SERVERAUTHTOKEN de la table QSYS2.SYSTEXTSERVERS dans l'instruction SQL INSERT.
Quand la base de données DB2 communique avec un serveur de recherche de texte, un jeton d'authentification est demandé. Ce jeton est généré sur le serveur de recherche de texte pendant l'installation.
3. Indiquez la clé de serveur de chaque serveur de recherche de texte dans la colonne SERVERMASTERKEY de la table QSYS2.SYSTEXTSERVERS dans l'instruction SQL INSERT.
4. Facultatif : Indiquez un nom d'alias pour le serveur dans la colonne ALIASNAME. Vous pouvez utiliser ce nom d'alias pour référencer le serveur au cours des opérations suivantes.

Exemple

Dans l'exemple suivant, une instruction SQL INSERT copie les informations requises pour un serveur de recherche de texte dans les colonnes de la table QSYS2.SYSTEXTSERVERS :

```
INSERT INTO QSYS2.SYSTEXTSERVERS(SERVERNAME,
                                SERVERADINFO,
                                SERVERPORT,
                                SERVERTYPE,
                                SERVERAUTHTOKEN,
                                SERVERMASTERKEY,
                                SERVERPATH,
                                ALIASNAME)
VALUES('127.0.0.1',
      VARBINARY(X'0000'),
      49200,
      0,
      'AH2X4w==',
      'b1YhcR90858ArwxLJeIY/Q==',
      '/QOpenSys/QIBM/ProdData/TextSearch/server1/bin/',
      'LOCALSERVER2');
```

Concepts associés

«Nom d'alias de serveur», à la page 11

Vous pouvez utiliser un nom d'alias de serveur pour affecter un nom significatif à un serveur.

Référence associée

«Table d'administration QSYS2.SYSTEXTSERVERS», à la page 106

La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.

Lancement des fonctions de recherche de texte

Avant de commencer à utiliser les fonctions de recherche de texte, appelez la procédure stockée SYSPROC.SYSTS_START. En appelant cette procédure, vous pouvez démarrer tous les serveurs de production que vous avez définis en mode local sur le système.

La prise en charge de la recherche de texte comprend des instructions SQL qui utilisent la fonction CONTAINS, la fonction SCORE et les procédures stockées d'administration suivantes :

- SYSPROC.SYSTS_CREATE
- SYSPROC.SYSTS_UPDATE
- SYSPROC.SYSTS_DROP

| **Référence associée**

- | «SYSPROC.SYSTS_START», à la page 16
- | Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.
- | «SYSPROC.SYSTS_CREATE», à la page 19
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte.
- | Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.
- | «SYSPROC.SYSTS_UPDATE», à la page 35
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.
- | «SYSPROC.SYSTS_DROP», à la page 33
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_DROP pour supprimer un index de recherche de texte créé avec la procédure stockée SYSPROC.SYSTS_CREATE.
- | «CONTAINS», à la page 38
- | Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.
- | «SCORE», à la page 41
- | Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

| **Création d'un index de recherche de texte**

- | Vous pouvez créer un index de recherche de texte en appelant la procédure stockée SYSPROC.SYSTS_CREATE.
- | La table de base DB2 doit contenir une colonne ROWID, une clé unique ou une clé primaire.
- | Pour créer un index de recherche de texte sur une table DB2 existante avec une colonne contenant du texte, procédez comme suit :
- | Appelez la procédure stockée SYSPROC.SYSTS_CREATE.
- | L'index de recherche de texte est vide jusqu'à la première mise à jour de l'index.

| **Référence associée**

- | «SYSPROC.SYSTS_CREATE», à la page 19
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte.
- | Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

| **Mise à jour d'un index de recherche de texte**

- | Pour mettre à jour un index de recherche de texte, vous devez appeler la procédure stockée SYSPROC.SYSTS_UPDATE.
- | L'appel à la procédure SYSTS_UPDATE permet de remplir pour la première fois l'index de recherche de texte. Vous l'utilisez également chaque fois que le contenu de la table DB2 change et que vous voulez synchroniser l'index de recherche de texte avec ces changements.
- | Après la mise à jour d'un index de recherche de texte, vous pouvez exécuter des requêtes de recherche sur cet index de recherche de texte. Vous pouvez changer la colonne de recherche de texte de la table de

| base après la mise à jour. Dans ce cas, les résultats de la requête de recherche ne refléteront les modifications qu'après une nouvelle mise à jour de l'index de recherche de texte.

| **Référence associée**

| «SYSPROC.SYSTS_UPDATE», à la page 35

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

| **Recherche d'un index de recherche de texte**

| Vous pouvez rechercher un index de recherche de texte à l'aide d'une instruction SQL avec une fonction CONTAINS ou SCORE. Les critères de l'argument de recherche sont spécifiés avec la fonction.

| L'utilisateur qui exécute les requêtes de texte sur une table DB2 doit détenir l'ensemble de privilèges standard requis pour toute forme de requête, comme indiqué dans *DB2 SQL Reference*.

| **Référence associée**

| «CONTAINS», à la page 38

| Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

| «SCORE», à la page 41

| Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

| **Troncature des documents**

| OmniFind Text Search Server for DB2 for i limite le nombre de caractères pouvant être indexés pour chaque document texte. Il arrive que cette limite occasionne la troncature des grands documents texte dans l'index de recherche de texte.

| Les documents qui contiennent plus de 10 millions de caractères Unicode peuvent être tronqués par le serveur de recherche de texte. Dans le cas d'un document texte enrichi, cette limite est appliquée une fois que le document a été converti en texte normal.

| Si un document texte est tronqué pendant la phase d'analyse syntaxique, un message d'avertissement vous informe que certains documents n'ont pas été totalement traités. L'avertissement apparaît dans le journal de travail. Le document est partiellement indexé. Le texte figurant au delà de la limite dans le document n'est ni indexé ni pris en compte pendant les recherches.

| Vous souhaitez peut-être supprimer le document tronqué de l'index de recherche de texte afin d'éviter un comportement indésirable pendant le traitement de la recherche. Pour supprimer ce document, vous pouvez supprimer l'enregistrement correspondant dans la table DB2 ou remplacer la valeur du document par une valeur vide ou nulle.

| **Procédures stockées d'administration pour la recherche de texte**

| Vous pouvez démarrer et arrêter des fonctions de recherche de texte et créer, supprimer ou mettre à jour des index de recherche de texte à l'aide d'un ensemble de procédures stockées d'administration en SQL. Vous pouvez appeler ces procédures depuis n'importe quelle interface SQL. Vous ne pouvez pas appeler ces procédures à partir d'une ligne de commande IBM i avec des commandes CL.

| Quand vous consultez les catalogues système avec STRSQL, vous pouvez voir le contenu des colonnes en affectant à votre travail un CCSID différent de 65535.

| **SYSPROC.SYSTS_START**

| Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.

| Les fonctions de recherche de texte prennent en charge les requêtes SQL qui utilisent la fonction CONTAINS, la fonction SCORE et les procédures stockées d'administration utilisées pour la maintenance des index de recherche de texte.

| Exécutez la procédure stockée SYSPROC.SYSTS_START chaque fois qu'un serveur est ajouté ou modifié dans la table QSYS2.SYSTEXTSERVERS.

| Si les fonctions de recherche de texte ne sont pas démarrées, la base de données renvoie le code d'erreur SQLCODE -20424 avec le code raison 4 pour les fonctions CONTAINS et SCORE. Les procédures d'administration SYSPROC.SYSTS_CREATE et SYSPROC.SYSTS_UPDATE échouent également avec le code d'erreur SQLCODE -20424 si le serveur n'est pas démarré.

| Pour les serveurs de recherche de texte contenus dans la table QSYS2.SYSTEXTSERVERS, les noms TCP/IP sont convertis. Les appels répétés à la procédure stockée SYSPROC.SYSTS_START ne sont pas considérés comme une erreur. Ce processus vous permet de vérifier la résolution de l'adresse dans la table QSYS2.SYSTEXTSERVERS.

| Dans le cas d'un serveur local, cet appel de procédure stockée démarre le serveur s'il ne l'est pas déjà. Dans le cas d'un serveur distant, l'appel de procédure vérifie que le serveur est actif mais ne le démarre pas dans la négative.

| **Conditions préalables**

| Avant d'appeler la procédure stockée SYSPROC.SYSTS_START, vérifiez que la table QSYS2.SYSTEXTSERVERS contient au moins une entrée.

| **Autorisation**

| L'ID utilisateur utilisé pour appeler cette procédure stockée doit détenir les privilèges suivants :

- | • Droit *EXECUTE sur la procédure
- | • Privilèges SELECT et UPDATE sur la table SYSTXTSRVRS
- | • Droit *EXECUTE sur la bibliothèque QSYS2 du fichier SYSTXTSRVRS
- | • Droit *JOBCTL ou accès à la fonction de sécurité spéciale QIBM_DB_SQLADM

| Pour plus d'informations sur les droits système correspondant aux privilèges SQL, voir GRANT (privilèges de table ou de vue).

| **Syntaxe**

| ►►SYSPROC.SYSTS_START—(——)►►

| Le qualificateur de schéma est SYSPROC.

| **Paramètre**

| *ID_serveur* ou *nom_alias*

| Indique l'identifiant du serveur à démarrer. La valeur de *ID_serveur* ou *nom_alias* est une chaîne. Si aucun identifiant n'est indiqué, tous les serveurs sont démarrés par défaut. La chaîne d'identifiant

doit être un *ID de serveur* existant dans la colonne SERVERID ou un *nom d'alias* de serveur existant dans la colonne ALIASNAME de la table QSYS2.SYSTEXTSERVERS. Si l'identifiant a été converti en entier, il est interprété comme un *ID de serveur*. Si l'identifiant n'est pas convertible en entier, il est interprété comme un *nom d'alias* de serveur.

Le type de données de ce paramètre est VARCHAR(128).

Remarque : Seuls les serveurs identifiés comme étant des serveurs de production sont démarrés quand aucune valeur n'est fournie pour *ID_serveur* ou *nom_alias*. Indique les serveurs de production identifiés par le paramètre **SERVERCLASS** = 0 dans la table QSYS2.SYSTEXTSERVERS. Les serveur de test doivent être démarrés en indiquant la valeurs de l'*ID de serveur* ou du *nom d'alias* qui lui est associé.

Pour démarrer tous les serveurs de production, entrez :

CALL SYSPROC.SYSTS_START().

Pour démarrer un serveur ayant l'ID 1, entrez :

CALL SYSPROC.SYSTS_START(1)

Pour démarrer un serveur ayant le nom d'alias "LOCAL_SERVER", entrez :

CALL SYSPROC.SYSTS_START('LOCAL_SERVER')

Pour démarrer un serveur ayant le nom d'alias "local_server", entrez :

CALL SYSPROC.SYSTS_START("local_server")

Concepts associés

«Nom d'alias de serveur», à la page 11

Vous pouvez utiliser un nom d'alias de serveur pour affecter un nom significatif à un serveur.

Tâches associées

«Démarrage de OmniFind Text Search Server for DB2 for i», à la page 65

Vous pouvez démarrer OmniFind Text Search Server for DB2 for i en appelant la procédure

SYSPROC.SYSTS_START.

Référence associée

«Table d'administration QSYS2.SYSTEXTSERVERS», à la page 106

La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.

«CONTAINS», à la page 38

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

«SCORE», à la page 41

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

SYSPROC.SYSTS_STOP

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_STOP pour arrêter les fonctions de recherche de texte de DB2. Cette procédure stockée affecte la valeur 1 (arrêté) au paramètre SERVERSTATUS dans le catalogue QSYS2.SYSTEXTSERVERS.

Une fois cette procédure stockée exécutée, les requêtes SQL qui utilisent les fonctions CONTAINS ou SCORE, ou les procédures stockées d'administration dédiées à la maintenance des index, renvoient un message d'échec sans chercher à contacter un serveur de recherche de texte.

| **Remarque :** Les procédures administratives et les requêtes SQL qui utilisaient des fonctions définies par
| l'utilisateur CONTAINS ou SCORE qui s'exécutaient avant l'appel de la procédure
| SYSTS_STOP peuvent s'exécuter jusqu'à leur terme.

| La journalisation des modifications de la table de base de l'index se poursuit, même après l'arrêt du
| serveur. Toutefois, les mises à jour planifiées de l'index ne s'exécutent qu'après l'appel de la procédure
| SYSPROC.SYSTS_START.

| **Autorisation**

| L'ID utilisateur utilisé pour appeler cette procédure stockée doit détenir les privilèges suivants :

- | • Droit *EXECUTE sur la procédure
- | • Privilèges SELECT et UPDATE sur la table SYSTEXTSERVERS
- | • Droit *EXECUTE sur la bibliothèque QSYS2 du fichier SYSTEXTSERVERS
- | • Droit *JOBCTL ou accès à la fonction de sécurité spéciale QIBM_DB_SQLADM

| Pour plus d'informations sur les droits système correspondant aux privilèges SQL, voir GRANT
| (privilèges de table ou de vue).

| **Syntaxe**

| ►►SYSTS_STOP—()—

| Le qualificateur de schéma est SYSPROC.

| **Paramètre**

| *ID_serveur* **ou** *nom_alias*

| Indique l'identifiant du serveur à arrêter. La valeur de *ID_serveur* ou *nom_alias* est une chaîne. Si
| aucun identifiant n'est indiqué, tous les serveurs sont arrêtés par défaut. La chaîne d'identifiant doit
| être un *ID de serveur* existant dans la colonne SERVERID ou un *nom d'alias* de serveur existant dans la
| colonne ALIASNAME de la table QSYS2.SYSTEXTSERVERS. Si l'identifiant a été converti en entier, il
| est interprété comme un *ID de serveur*. Si l'identifiant n'est pas convertible en entier, il est interprété
| comme un *nom d'alias* de serveur.

| Le type de données de ce paramètre est VARCHAR(128).

| Pour arrêter tous les serveurs de production, entrez :

| CALL SYSPROC.SYSTS_STOP().

| Pour arrêter un serveur ayant l'ID 1, entrez :

| CALL SYSPROC.SYSTS_STOP(1)

| Pour arrêter un serveur ayant le nom d'alias "LOCAL_SERVER", entrez :

| CALL SYSPROC.SYSTS_STOP('LOCAL_SERVER')

| Pour arrêter un serveur ayant le nom d'alias "local_server", entrez :

| CALL SYSPROC.SYSTS_STOP('local_server')

Concepts associés

«Nom d’alias de serveur», à la page 11

Vous pouvez utiliser un nom d’alias de serveur pour affecter un nom significatif à un serveur.

Référence associée

«Table d’administration QSYS2.SYSTEXTSERVERS», à la page 106

La table d’administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.

«SYSPROC.SYSTS_START», à la page 16

Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.

«CONTAINS», à la page 38

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l’aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

«SCORE», à la page 41

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l’aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

SYSPROC.SYSTS_CREATE

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l’indexation de recherche de texte. Vous pouvez ensuite utiliser l’index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

L’index de recherche de texte est créé sur l’un des serveurs de recherche de texte répertoriés dans la table QSYS2.SYSTEXTSERVERS. L’index de recherche de texte n’est pas mis à jour en mode synchrone quand la table DB2 est mise à jour. Au lieu de cela, un fichier journal des modifications de la colonne de table DB2 est capturé par des déclencheurs et placé dans une table de transfert.

Remarque : Cette procédure stockée définit uniquement l’index de recherche de texte. L’index de recherche de texte ne contient pas de données jusqu’au premier appel de la procédure stockée SYSPROC.SYSTS_UPDATE pour le nouvel index de recherche de texte. Vous créez l’index de recherche de texte après le premier remplissage de la table. En créant l’index de recherche de texte après le premier remplissage de la table, vous évitez la charge des déclencheurs de changement avant la première mise à jour de l’index.

Conditions préalables

Avant d’appeler la procédure stockée SYSPROC.SYSTS_CREATE, vérifiez que les prérequis suivants sont satisfaits :

- Vous avez démarré les fonctions de recherche de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START et au moins un serveur de recherche de texte est en cours d’exécution.
- La table contient une colonne qui est définie comme clé primaire, index à entrées uniques, ou ROWID.
- La table QSYS2.SYSTEXTSERVERS contient au moins une entrée.

Autorisation

Les privilèges associés à l’ID de l’émetteur de l’instruction doivent comprendre au moins un des privilèges suivants :

- Le privilège de créer le schéma. Pour plus d’informations, voir Autorisation, privilèges et propriété des objets.
- Droits d’administration

| Les privilèges associés à l’ID de l’émetteur de l’instruction doivent comprendre au moins un des privilèges suivants :

- | • Les droits système suivants :
 - | – *USE, sur la commande Create Logical File (CRTLF)
 - | – *CHANGE, sur le dictionnaire de données si la bibliothèque dans laquelle l’index de texte est créé est un schéma SQL contenant un dictionnaire de données
- | • Droits d’administration

| Les privilèges associés à l’ID de l’émetteur de l’instruction doivent aussi comprendre au moins un des privilèges suivants :

- | • Pour la table référencée :
 - | – Le privilège INDEX sur la table
 - | – Le droit système *EXECUTE sur la bibliothèque contenant la table
- | • Droits d’administration
- | • Si des noms SQL sont spécifiés, qu’il existe un profil utilisateur portant le même nom que la bibliothèque dans laquelle l’index de texte est créé et que ce nom est différent de l’ID de l’émetteur de l’instruction, les privilèges associé à cet utilisateur doivent inclure au moins l’un des privilèges suivants :
 - | – Droit système *ADD sur le profil utilisateur portant ce nom
 - | – Droits d’administration

| Si un autre type distinct est référencé, les privilèges associés à l’ID de l’émetteur de l’instruction doivent comprendre au moins un des privilèges suivants :

- | • Pour chaque type distinct identifié dans l’instruction :
 - | – le privilège USAGE sur le type distinct, et
 - | – le droit système *EXECUTE sur la bibliothèque qui contient le type distinct
- | • Droits d’administration

| Pour plus d’informations sur les droits système correspondant aux privilèges SQL, voir GRANT (privilèges de table ou de vue).

| Syntaxe

| ►►SYSTS_CREATE—(—schéma_index—,—nom_index—,—source_texte—,—options—)——————►

| null

| Le qualificateur de schéma est SYSPROC.

| Paramètres

| *schéma_index*
 | Identifie le schéma de l’index de recherche de texte. Si la valeur de ce paramètre est nulle, la valeur du registre spécial CURRENT SCHEMA utilisé pour l’auteur de l’appel est utilisée. Cette valeur doit être un nom SQL valide.

| **Remarque :** Encadrez les noms par des doubles guillemets s’ils entrent en conflit avec des mots clés SQL ou des mots clés OmniFind pouvant être utilisés.

| Le type de données de ce paramètre est VARCHAR(128).

| *nom_index*
 | Identifie le nom de l’index de recherche de texte. Le nom de l’index de recherche de texte associé au

schéma d'index identifie individuellement l'index de recherche de texte dans le sous-système DB2. Vous devez indiquer une valeur non nulle pour ce paramètre. Cette valeur doit être un nom SQL valide.

Remarque : Encadrez les noms par des doubles guillemets si les noms entrent en conflit avec des mots clés SQL ou des mots clés OmniFind pouvant être utilisés.

Le type de données de ce paramètre est VARCHAR(128).

source_texte

Identifie la spécification de colonne et de table associée à la source du document texte. Ce paramètre peut inclure des fonctions définies par l'utilisateur. Vous devez indiquer une valeur non nulle pour ce paramètre.

Le type de données de ce paramètre est VARCHAR(1024).

Paramètre source_texte :

| `schéma_table. nom_table—(nom_colonne_texte—)`
| `| nom_fonction—(—nom_colonne_texte—)`
| `| schéma_fonction | . |`

schéma_table

Identifie le schéma de la table sur laquelle l'index de recherche de texte est créé.

Remarque : Encadrez les noms par des doubles guillemets si les noms entrent en conflit avec des mots clés SQL ou des mots clés OmniFind pouvant être utilisés.

nom_table

Identifie le nom de la table de texte qui contient la colonne sur laquelle l'index de recherche de texte externe est créé.

Remarques :

- Les vues et les fichiers logiques ne sont pas pris en charge.
- Un alias doit pointer sur une table ou un membre spécifique d'un fichier physique.
- Encadrez les noms par des doubles guillemets si les noms entrent en conflit avec des mots clés SQL ou des mots clés OmniFind.

nom_colonne_texte

Identifie le nom de la colonne qui contient le texte utilisé pour créer l'index de recherche de texte. Cette colonne doit contenir des données de type CHAR, CHAR FOR BIT DATA, BINARY, VARCHAR, VARCHAR FOR BIT DATA, VARBINARY, CLOB, DBCLOB, BLOB, XML, GRAPHIC, ou VARGRAPHIC. Si ces données sont d'un autre type, vous pouvez indiquer une fonction externe qui renvoie un types de données pris en charge.

Remarques :

- Il n'est admis qu'un seul index de recherche de texte par colonne. S'il existe déjà un index de recherche de texte pour une colonne, le message SQLCODE-20427 est renvoyé.
- Encadrez les noms par des doubles guillemets si les noms entrent en conflit avec des mots clés SQL ou des mots clés OmniFind.

schéma_fonction. nom_fonction

Identifie le schéma et le nom d'une fonction définie par l'utilisateur intégrée. Vous pouvez utiliser cette fonction pour modifier un document texte stocké dans une colonne. Vous pouvez également l'utiliser pour accéder aux documents texte d'une colonne contenant des données d'un type non pris en charge. Cette fonction peut aussi s'utiliser pour accéder à un

document stocké ailleurs. La fonction possède un paramètre d'entrée qui désigne le type de données de la colonne de texte. Par exemple, un entier utilisé comme clé étrangère pour le contenu d'un document résidant dans une autre table. La fonction renvoie une valeur dans l'un des types de données pris en charge par OmniFind Text Search for DB2 for i. La fonction convertit le contenu de la colonne de texte en contenu de document indexé.

Remarques :

- Les fonctions de transtypage et les fonctions utilisant plus d'un argument ne sont pas prises en charge.
- Encadrez les noms par des doubles guillemets si les noms entrent en conflit avec des mots clés SQL ou des mots clés OmniFind pouvant être utilisés.

options

Chaîne de caractères qui indique les options disponibles pour cette procédure stockée.

Le type de données de ce paramètre est VARCHAR(32000).

options :

<i>informations par défaut_texte</i>	<i>caractéristiques_mise à jour</i>	<i>options_configuration_index</i>
--------------------------------------	-------------------------------------	------------------------------------

informations par défaut_texte :

CCSID— <i>ccsid</i>	LANGUAGE— <i>langue</i>	FORMAT— <i>format</i>
---------------------	-------------------------	-----------------------

informations par défaut_texte

Ce paramètre indique l'ID de jeu de caractères codés utilisé pour l'indexation des documents texte binaires. Il indique également la langue utilisée pour le traitement des documents et le format des documents texte contenus dans la colonne.

CCSID *ccsid*

ID du jeu de caractères codés utilisé pour un index de recherche de texte sur une colonne contenant des données de type binaire. La valeur par défaut est 1208 (UTF-8), elle est fournie par la table QSYS2.SYSTEXTDEFAULTS. Ce paramètre accepte tous les CCSID pris en charge pour la conversion en UTF-8 par les services de conversion d'IBM i.

Ce paramètre est ignoré dans le cas d'un index de recherche de texte dans une colonne contenant des données non binaires. Les colonnes de texte héritent du CCSID de la spécification de table. La valeur de *ccsid* est ignorée quand *format* a la valeur INSO.

LANGUAGE *langue*

Indique la langue utilisée par OmniFind Text Search Server for DB2 for i pour le traitement linguistique des documents texte. La valeur par défaut est en_US (anglais). Si vous entrez la valeur AUTO, OmniFind Text Search Server for DB2 for i essaie de déterminer la langue de manière automatique.

Important : Si la langue des documents n'est pas l'anglais, n'utilisez pas la valeur par défaut en_US. Remplacez cette valeur par la langue des documents, sans quoi le traitement linguistique ne fonctionnera pas comme prévu.

FORMAT *format*

Identifie le format des documents texte contenus dans la colonne, par exemple HTML. OmniFind Text Search Server for DB2 for i a besoin de connaître le format, ou le type de contenu, des documents texte que vous voulez indexer et interroger. Si vous n'indiquez pas le

paramètre *format* la valeur par défaut est extraite de la colonne FORMAT de la table QSYS2.SYSTEXTDEFAULTS. Le paramètre *format* accepte les valeurs TEXT, HTML, XML et INSO.

La valeur de *format* INSO permet à OmniFind Text Search Server for DB2 for i de déterminer le format. Dans ce cas, la valeur de *ccsid* est ignorée. Si OmniFind Text Search Server for DB2 for i ne parvient pas à déterminer le format de document, une erreur est consignée dans le journal de travail au cours du traitement de la procédure stockée SYSPROC.SYSTS_UPDATE.

Remarque : Si vous n'indiquez pas le paramètre *format* quand vous créez l'index sur une colonne de données XML, la valeur par défaut est XML. Si vous affectez au paramètre *format* la valeur TEXT ou INSO, la fonction de recherche XML est indisponible avec cet index. De plus, un message d'avertissement apparaît dans le journal de travail.

caractéristiques_mise à jour :



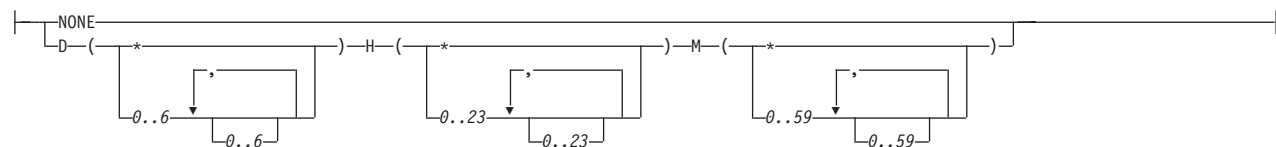
caractéristiques_mise à jour

Indique la fréquence des mises à jour automatiques dans l'index de recherche de texte. Indique aussi le nombre minimum de modifications des documents texte devant intervenir avant que l'index de recherche de texte ne soit mis à jour par incrément au moment spécifié.

UPDATE FREQUENCY *fréquence_mise à jour*

Indique la fréquence des mises à jour automatiques dans l'index de recherche de texte. La valeur par défaut est NONE. Cette option est appropriée pour une colonne de texte ne devant subir aucune modification. L'option *fréquence_mise à jour* accepte deux formats différents :

fréquence_mise à jour (format 1) :



NONE

Si la valeur NONE est spécifiée, aucune mise à jour de l'index n'intervient. La mise à jour doit être exécutée manuellement. Cette option est appropriée pour une colonne de texte ne devant subir aucune modification.

D Indique le ou les jours de la semaine où l'index est mis à jour. L'astérisque (*) signifie "tous les jours". 0 signifie dimanche.

H Indique la ou les heures où l'index est mis à jour. L'astérisque (*) signifie "toutes les heures".

M Indique la ou les minutes où l'index est mis à jour. Vous ne pouvez pas spécifier l'astérisque (*). La fréquence minimum est de 5 minutes.

Exemple : Dans cet exemple, l'index est mis à jour toutes les 30 minutes.

UPDATE FREQUENCY D(*) H(*) M(0,30)

| fréquence_mise à jour (format 2, chronologique) :

| |< minute>—< heure>—<jour du mois>—<mois de l'année>—<jour de la semaine>—|

| L'option *fréquence_mise à jour (chronologique)* se présente sous la forme d'une liste de cinq valeurs séparées par un espace. Les cinq valeurs représentent respectivement les minutes, les heures, les jours du mois, les mois de l'année et les jours de la semaine à partir du dimanche.

| Si vous indiquez un intervalle de valeurs ou un astérisque (*), vous pouvez spécifier une valeur d'écart en ajoutant une barre oblique (/) à la fin de l'intervalle.

| **Exemple:** Dans cet exemple, l'index est mis à jour à chaque premier quart d'heure (0,15,30,45) des heures paires entre 8h00 et 18h45 (8-18/2 équivaut à 8,10,12,14,16,18), du lundi au vendredi tous les mois de l'année (* * 1-5).

| 0,15,30,45 8-18/2 * * 1-5

| *minute* Indique les minutes de l'heure où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier un intervalle de 5 minutes, ou spécifier un entier compris entre 0 et 59. Vous ne pouvez pas répéter les valeurs. La fréquence minimum est de 5 minutes. Les valeurs 1, 4 et 8 sont interdites.

| fréquence_mise à jour (minute) :



| *heure* Indique les heures de la journée où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier toutes les heures ou spécifier un entier compris entre 0 et 23. Vous ne pouvez pas répéter les valeurs.

| fréquence_mise à jour (heure) :



| *jour du mois* Indique les jours du mois où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier tous les jours ou spécifier un entier compris entre 1 et 31. Vous ne pouvez pas répéter les valeurs.

| fréquence_mise à jour (jour du mois) :



mois de l'année

Indique les mois de l'année où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier tous les mois ou spécifier un entier compris entre 1 et 12. Vous ne pouvez pas répéter les valeurs.

fréquence_mise à jour (mois de l'année) :



jour de la semaine

Indique les jours de la semaine où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier tous les jours ou spécifier un entier compris entre 0 et 7. Vous ne pouvez pas répéter les valeurs.

fréquence_mise à jour (jour de la semaine) :



UPDATE MINIMUM *modification minimum*

Indique le nombre minimum de modifications des enregistrements devant intervenir dans la table sous-jacente avant que l'index de recherche de texte ne soit mis à jour par incrément au moment spécifié dans l'option *fréquence_mise à jour*. La valeur doit être un entier compris entre 1 et 2147483647. La valeur par défaut est extraite de la colonne UPDATEMINIMUM de la table QSYS2.SYSTEXTDEFAULTS.

Cette option est ignorée quand vous mettez à jour l'index de recherche de texte, sauf si vous indiquez l'option USING UPDATE MINIMUM dans la procédure stockée SYSPROC.SYSTS_UPDATE.

options_configuration_index :



options_configuration_index

Indique des valeurs spécifiques à l'index sous la forme de paires option-valeur. Vous devez encadrer les valeurs de chaîne par des guillemets simples. Un guillemet simple à l'intérieur d'une valeur de chaîne doit être représenté par deux guillemets simples à la suite.

CJKSEGMENTATION

Indique la méthode de segmentation à utiliser pour indexer les documents en chinois, japonais et coréen. Les valeurs prises en charge sont MORPHOLOGICAL et NGRAM. Si vous n'indiquez pas la valeur de CJKSEGMENTATION, la valeur par défaut est utilisée. Cette valeur par défaut est spécifiée par la valeur de DEFAULTNAME dans la table QSYS2.SYSTEXTDEFAULTS.

COMMENT

Spécifie un commentaire stocké dans la colonne REMARKS de la table d'administration QSYS2.SYSTEXTINDEXES comme description de la collection OmniFind Text Search Server for DB2 for i.

La valeur de cette option est une chaîne inférieure ou égale à 512 octets.

IGNOREEMPTYDOCS

Indique s'il faut représenter les documents vides dans l'index de recherche de texte. Les documents vides sont des documents qui contiennent une chaîne vide ou une valeur nulle.

Les valeurs prises en charge pour cette option sont 0 (zéro) et 1. La valeur par défaut est 1.

Si cette option a la valeur 1, les documents vides ne sont pas représentés dans l'index de recherche de texte. Si vous utilisez cette option et que vous indiquez un contenu de document vide, la mise à jour incrémentielle suivante supprime le document dans l'index de recherche de texte.

KEYCOLUMN

Indique le nom d'une colonne à utiliser comme colonne clé dans l'index de texte. La colonne clé est utilisée pour associer des données de l'index de texte à un document ou à une ligne de la table de base. La colonne spécifiée doit posséder une contrainte de clé primaire ou un index à entrées uniques. Si vous ne spécifiez pas le paramètre KEYCOLUMN, la colonne ROWID de la table est utilisée, si elle existe. Sinon, la clé primaire définie sur la table est utilisée.

SERVER

Indique l'ID ou le nom d'alias du serveur à utiliser pour stocker l'index de recherche de texte. Si un ID est utilisé, sa valeur est un entier qui doit exister dans la colonne SERVERID du catalogue QSYS2.SYSTEXTSERVERS. Si un nom d'alias est utilisé, sa valeur est une chaîne qui doit exister dans la colonne ALIASNAME du catalogue QSYS2.SYSTEXTSERVERS. Si le paramètre SERVER n'est pas spécifié, le programme sélectionne par défaut le serveur contenant le moins d'index de recherche de texte parmi les serveurs répertoriés dans la table QSYS2.SYSTEXTSERVERS dont le paramètre SERVERSTATUS a la valeur 0 (zéro), ce qui signifie que le serveur est disponible.

UPDATEAUTOCOMMIT

Indique la fréquence des opérations de validation lors de l'extraction des documents au cours d'une mise à jour de l'index. La valeur 0 (zéro) signifie qu'une opération de validation intervient à la fin du traitement uniquement.

La valeur doit être un entier compris entre 0 et 2147483647. La valeur par défaut est 100.

Conseil pour les performances : La valeur de UPDATEAUTOCOMMIT peut avoir un impact substantiel sur les performances des mises à jour de l'index. L'opération de validation qui s'exécute à l'intervalle spécifié procure un point de contrôle cohérent à partir duquel vous pouvez redémarrer la mise à jour de l'index en cas d'interruption. Toutefois, cette validation a aussi pour effet de suspendre temporairement le processus de mise à jour. Augmenter la valeur de UPDATEAUTOCOMMIT (ou la mettre à 0) peut améliorer significativement les performances, en particulier lors de la première mise à jour. La valeur que vous indiquez doit procurer un équilibre entre le besoin de performance et le besoin de reprise sur la base de la fréquence des mises à jour de l'index.

Valeurs par défaut du paramètre *options*

Quand vous installez OmniFind Text Search for DB2 for i, la table QSYS2.SYSTEXTDEFAULTS est créée et remplie avec des valeurs par défaut pour le paramètre *options* de la procédure stockée SYSPROC.SYSTS_CREATE.

| Le tableau suivant répertorie les options, les valeurs par défaut et les descriptions de ces options.

| *Tableau 2. Valeurs par défaut du paramètre options*

Option	Valeur par défaut	Description
CCSID	1208	Indique l'ID de jeu de caractères codés utilisé pour l'indexation des documents texte binaires.
CJKSEGMENTATION	NGRAM	Indique la méthode de segmentation à utiliser pour indexer les documents en chinois, japonais et coréen.
LANGUAGE	en_US	Indique la langue utilisée pour traiter les documents texte.
FORMAT	TEXT	Indique le format des documents texte contenus dans la colonne. Le format par défaut est le texte en clair sauf si le type de données est XML.
UPDATEFREQUENCY	NONE	Indique qu'aucune mise à jour automatique n'est planifiée.
UPATEMINIMUM	1	Si au moins un document a changé depuis la dernière mise à jour de l'index, la procédure stockée SYSPROC.SYSTS_UPDATE démarre son traitement.
IGNOREEMPTYDOCS	1	Indique que les documents vides (documents qui contiennent une chaîne vide ou une valeur nulle) ne seront pas représentés dans l'index de recherche de texte. Les zones de métadonnées de ces documents ne seront pas prises en compte lors des recherches.
UPDATEAUTOCOMMIT	100	Indique la fréquence des opérations de validation lors de l'extraction des documents au cours d'une mise à jour de l'index.
MINIMUMUPDATEINTERVAL	5	Indique les intervalles pour l'option UPDATEFREQUENCY. L'intervalle minimum est de 5 minutes.
USEREXITTHREADS	0	Réservé

| Concepts associés

- | «Formats de document pris en charge», à la page 6
- | Les données d'une colonne de texte peuvent comporter du texte en clair, un document HTML, un document XML, ou tout autre document reconnu par le moteur de recherche.
- | «Types de données pris en charge», à la page 7
- | Les données des colonnes de texte que vous voulez indexer et interroger peuvent être des données binaires ou des données de type caractères.

| Référence associée

- | «Table d'administration QSYS2.SYSTEXTSERVERS», à la page 106
- | La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.
- | «Table d'administration QSYS2.SYSTEXTINDEXES», à la page 103
- | La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de schéma et un nom de collection associée sur le serveur de recherche de texte.
- | «Table d'administration QSYS2.SYSTEXTDEFAULTS», à la page 102
- | La table d'administration QSYS2.SYSTEXTDEFAULTS contient des paramètres par défaut et leurs valeur. Cette table est créée pendant l'installation de OmniFind Text Search for DB2 for i.
- | «SYSPROC.SYSTS_UPDATE», à la page 35
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.
- | «SYSPROC.SYSTS_START», à la page 16
- | Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.
- | «Langues prises en charge», à la page 9
- | Vous pouvez demander que les documents texte soient traités dans une langue spécifique.

| SYSPROC.SYSTS_ALTER

- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_ALTER pour modifier les attributs d'un index créé par la procédure SYSPROC.SYSTS_CREATE. Seuls les attributs spécifiés explicitement dans cette procédure sont modifiés. Les autres attributs de l'index demeurent inchangés.
- | Cela est utile si vous avez besoin de changer les attributs de l'index, par exemple la fréquence des mises à jour, après sa création.

| Conditions préalables

- | Vérifiez les prérequis suivants avant d'appeler la procédure stockée SYSPROC.SYSTS_ALTER :
 - L'index de recherche de texte a été créé (en appelant la procédure stockée SYSPROC.SYSTS_CREATE).

| Autorisation

- | Les privilèges associés à l'ID de l'émetteur de l'instruction doivent comprendre au moins un des privilèges suivants :
 - Le privilège de créer le schéma. Pour plus d'informations, voir Autorisation, privilèges et propriété des objets.
 - Droits d'administration
- | Les privilèges associés à l'ID de l'émetteur de l'instruction doivent comprendre au moins un des privilèges suivants :
 - Les droits système suivants :
 - *USE, sur la commande Create Logical File (CRTLF)

- | – *CHANGE, sur le dictionnaire de données si la bibliothèque dans laquelle l’index de texte est créé est un schéma SQL contenant un dictionnaire de données
- | • Droits d’administration

| Les privilèges associés à l’ID de l’émetteur de l’instruction doivent aussi comprendre au moins un des privilèges suivants :

- | • Pour la table référencée :
 - | – Le privilège INDEX sur la table
 - | – Le droit système *EXECUTE sur la bibliothèque contenant la table
- | • Droits d’administration
- | • Si des noms SQL sont spécifiés, qu’il existe un profil utilisateur portant le même nom que la bibliothèque dans laquelle l’index de texte est créé et que ce nom est différent de l’ID de l’émetteur de l’instruction, les privilèges associés à cet utilisateur doivent inclure au moins l’un des droits d’accès suivants :
 - | – Droit système *ADD sur le profil utilisateur portant ce nom
 - | – Droits d’administration

| Si un autre type distinct est référencé, les privilèges associés à l’ID de l’émetteur de l’instruction doivent comprendre au moins un des privilèges suivants :

- | • Pour chaque type distinct identifié dans l’instruction :
 - | – le privilège USAGE sur le type distinct, et
 - | – le droit système *EXECUTE sur la bibliothèque qui contient le type distinct
- | • droits d’administration

| Pour plus d’informations sur les droits système correspondant aux privilèges SQL, voir GRANT (privilèges de table ou de vue).

| **Syntaxe**

| ►►SYSTS_ALTER—(—schéma_index—,—nom_index—,—options—)—————►

| Le qualificateur de schéma est SYSPROC.

| **Paramètres**

| *schéma_index*

| Identifie le schéma de l’index de recherche de texte. Si la valeur de ce paramètre est nulle, la valeur du registre spécial CURRENT SCHEMA utilisé pour l’auteur de l’appel est utilisée.

| Le type de données de ce paramètre est VARCHAR(128).

| *nom_index*

| Identifie le nom de l’index de recherche de texte. Le nom de l’index de recherche de texte associé au schéma d’index identifie de manière unique l’index de recherche de texte dans le sous-système DB2. Vous devez indiquer une valeur non nulle pour ce paramètre.

| Le type de données de ce paramètre est VARCHAR(128).

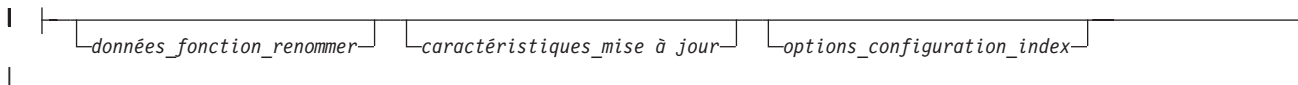
| *options*

| Chaîne de caractères qui indique les options disponibles pour cette procédure stockée.

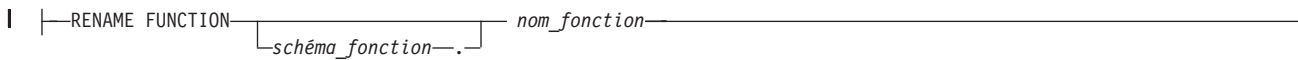
| Le type de données de ce paramètre est VARCHAR(32000).

| La valeur du paramètre ne peut pas être NULL.

options :



données_fonction_renommer :



Indique la fonction définie par l'utilisateur à renommer.

schéma_fonction.nom_fonction

Indique le schéma et le nom d'une fonction définie par l'utilisateur.

Cette option permet de changer une fonction spécifiée pendant la création d'une index. Si la fonction est modifiée, la procédure SYSTS_UPDATE utilise la nouvelle fonction pour indexer la colonne de texte.

Si la fonction a été modifiée, la procédure SYSTS_UPDATE ne change pas les données existantes de l'index. Seules les données modifiées après la dernière mise à jour sont traitées.

caractéristiques_mise à jour :



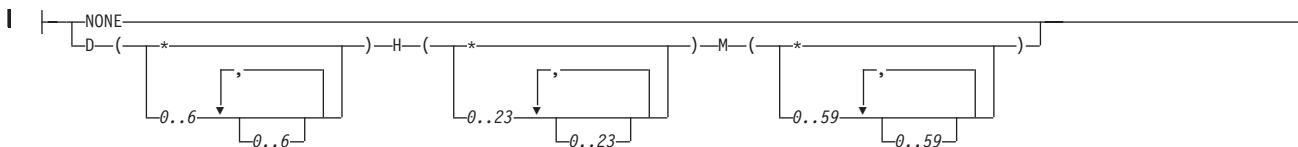
caractéristiques_mise à jour

Indique la fréquence des mises à jour automatiques dans l'index de recherche de texte. Indique aussi le nombre minimum de modifications des documents texte devant intervenir avant que l'index de recherche de texte ne soit mis à jour par incrément au moment spécifié.

UPDATE FREQUENCY *fréquence_mise à jour*

Indique la fréquence des mises à jour automatiques dans l'index de recherche de texte. La valeur par défaut est NONE. Cette option est appropriée pour une colonne de texte ne devant subir aucune modification. L'option fréquence_mise à jour accepte deux formats différents :

fréquence_mise à jour (format 1) :



NONE

Si la valeur NONE est spécifiée, aucune mise à jour de l'index n'intervient. La mise à jour doit être exécutée manuellement. Cette option est appropriée pour une colonne de texte ne devant subir aucune modification.

D

Indique le ou les jours de la semaine où l'index est mis à jour. L'astérisque (*) signifie "tous les jours". 0 signifie dimanche.

H

Indique la ou les heures où l'index est mis à jour. L'astérisque (*) signifie "toutes les heures".

| **M** Indique la ou les minutes où l'index est mis à jour. Vous ne pouvez pas spécifier l'astérisque (*). La fréquence minimum est de 5 minutes.

| **Exemple :** Dans cet exemple, l'index est mis à jour toutes les 30 minutes.

| UPDATE FREQUENCY D(*) H(*) M(0,30)

| **fréquence_mise à jour (format 2, chronologique) :**

| |—< minute>—< heure>—<jour du mois>—<mois de l'année>—<jour de la semaine>—|

| L'option *fréquence_mise à jour (chronologique)* se présente sous la forme d'une liste de cinq valeurs séparées par un espace. Les cinq valeurs représentent respectivement les minutes, les heures, les jours du mois, les mois de l'année et les jours de la semaine à partir du dimanche.

| Si vous indiquez un intervalle de valeurs ou un astérisque (*), vous pouvez spécifier une valeur d'écart en ajoutant une barre oblique (/) à la fin de l'intervalle.

| **Exemple:** Dans cet exemple, l'index est mis à jour à chaque premier quart d'heure (0,15,30,45) des heures paires entre 8h00 et 18h45 (8-18/2 équivaut à 8,10,12,14,16,18), du lundi au vendredi tous les mois de l'année (* * 1-5).

| 0,15,30,45 8-18/2 * * 1-5

| *minute* Indique les minutes de l'heure où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier un intervalle de 5 minutes, ou spécifier un entier compris entre 0 et 59. Vous ne pouvez pas répéter les valeurs. La fréquence minimum est de 5 minutes. Les valeurs 1, 4 ou 8 ne sont pas admises.

| **fréquence_mise à jour (minute) :**



| *heure* Indique les heures de la journée où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier toutes les heures ou spécifier un entier compris entre 0 et 23. Vous ne pouvez pas répéter les valeurs.

| **fréquence_mise à jour (heure) :**



| *jour du mois* Indique les jours du mois où l'index de recherche de texte est mis à jour. Vous pouvez indiquer un astérisque (*) pour spécifier tous les jours ou spécifier un entier compris entre 1 et 31. Vous ne pouvez pas répéter les valeurs.

| **fréquence_mise à jour (jour du mois) :**



| *mois de l'année*

| Indique les mois de l'année où l'index de recherche de texte est mis à jour. Vous
 | pouvez indiquer un astérisque (*) pour spécifier tous les mois ou spécifier un entier
 | compris entre 1 et 12. Vous ne pouvez pas répéter les valeurs.

| **fréquence_mise à jour (mois de l'année) :**



| *jour de la semaine*

| Indique les jours de la semaine où l'index de recherche de texte est mis à jour. Vous
 | pouvez indiquer un astérisque (*) pour spécifier tous les jours ou spécifier un entier
 | compris entre 0 et 7. Vous ne pouvez pas répéter les valeurs.

| **fréquence_mise à jour (jour de a semaine) :**



| **UPDATE MINIMUM** *modification minimum*

| Indique le nombre minimum de modifications des enregistrements devant intervenir dans la
 | table sous-jacente avant que l'index de recherche de texte ne soit mis à jour par incrément au
 | moment spécifié dans l'option *fréquence_mise à jour*. La valeur doit être un entier compris
 | entre 1 et 2147483647.

| **options_configuration_index :**



| **COMMENT**

| Spécifie un commentaire stocké dans la colonne REMARKS de la table d'administration
 | QSYS2.SYSTEXTINDEXES comme description de la collection OmniFind Text Search Server for
 | DB2 for i.

| La valeur de cette option est une chaîne inférieure ou égale à 512 octets.

UPDATEAUTOCOMMIT

Indique la fréquence des opérations de validation lors de l'extraction des documents au cours d'une mise à jour de l'index. La valeur 0 (zéro) signifie qu'une opération de validation intervient à la fin du traitement uniquement.

La valeur doit être un entier compris entre 0 et 2147483647.

Conseil pour les performances : La valeur de UPDATEAUTOCOMMIT peut avoir un impact substantiel sur les performances des mises à jour de l'index. L'opération de validation qui s'exécute à l'intervalle spécifié procure un point de contrôle cohérent à partir duquel vous pouvez redémarrer la mise à jour de l'index en cas d'interruption. Toutefois, cette validation a aussi pour effet de suspendre temporairement le processus de mise à jour. Augmenter la valeur de UPDATEAUTOCOMMIT (ou la mettre à 0) peut améliorer significativement les performances, en particulier lors de la première mise à jour. La valeur que vous indiquez doit procurer un équilibre entre le besoin de performance et le besoin de reprise sur la base de la fréquence des mises à jour de l'index.

Référence associée

«SYSPROC.SYSTS_CREATE», à la page 19

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte. Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

«SYSPROC.SYSTS_UPDATE», à la page 35

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

«Table d'administration QSYS2.SYSTEXTINDEXES», à la page 103

La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de schéma et un nom de collection associée sur le serveur de recherche de texte.

SYSPROC.SYSTS_DROP

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_DROP pour supprimer un index de recherche de texte créé avec la procédure stockée SYSPROC.SYSTS_CREATE.

Il est conseillé de supprimer un index de recherche de texte avec la procédure stockée SYSPROC.SYSTS_DROP avant de supprimer la table.

Le processus de suppression d'une vue représentant l'index de recherche de texte, même avec l'instruction DROP TABLE CASCADE, essaie aussi de supprimer l'index de recherche de texte associé. Toutefois, comme l'index de recherche de texte ne peut pas être supprimé si le contrôle de validation est activé, vous ne pouvez pas supprimer la vue SQL dans ce contexte.

Si le serveur de recherche de texte est inaccessible, la collection présente sur le serveur peut devenir orpheline. Dans ce cas, il faudra supprimer la collection manuellement. Quand le serveur sera à nouveau disponible, vous utiliserez l'outil d'administration d'OmniFind Text Search Server for DB2 for i pour supprimer la collection sur le serveur.

Dans «Outils d'administration», à la page 71, vous pouvez rechercher des informations sur les outils qui permettent d'identifier les index orphelins et sur la procédure stockée à utiliser (STSPROC.SYSTS_REMOVE) pour les supprimer.

Conditions préalables

Vérifiez les prérequis suivants avant d'appeler la procédure stockée SYSPROC.SYSTS_DROP :

- Vous avez démarré les fonctions de recherche de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.
- L'index de recherche de texte a été créé (en appelant la procédure stockée SYSPROC.SYSTS_CREATE).
- Vérifiez que les procédures stockées suivantes ne sont pas en cours d'exécution pour l'index de recherche de texte que vous voulez supprimer : SYSPROC.SYSTS_CREATE, SYSPROC.SYSTS_UPDATE et SYSPROC.SYSTS_DROP.

Autorisation

Les privilèges associés à l'ID de l'émetteur de l'instruction doivent comprendre au moins un des privilèges suivants :

- Les droits système suivants :
 - Droits système *OBJOPR et *OBJEXIST sur l'index de texte à supprimer
 - Droit système *EXECUTE sur la bibliothèque qui contient l'index de texte à supprimer.
- Droits d'administration

Pour plus d'informations sur les droits système correspondant aux privilèges SQL, voir GRANT (privilèges de table ou de vue).

Syntaxe

```

▶▶ SYSTS_DROP ( ( schéma_index , nom_index )
                [ null ]

```

Le qualificateur de schéma est SYSPROC.

Paramètres

schéma_index

Identifie le schéma de l'index de recherche de texte. Si la valeur de ce paramètre est nulle, la valeur du registre spécial CURRENT SCHEMA utilisé pour l'auteur de l'appel est utilisée.

Le type de données de ce paramètre est VARCHAR(128).

nom_index

Identifie le nom de l'index de recherche de texte. Le nom de l'index de recherche de texte associé au schéma d'index identifie de manière unique l'index de recherche de texte dans le sous-système DB2. Vous devez indiquer une valeur non nulle pour ce paramètre.

Le type de données de ce paramètre est VARCHAR(128).

Référence associée

«SYSPROC.SYSTS_CREATE», à la page 19

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte. Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

«SYSPROC.SYSTS_START», à la page 16

Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.

«SYSPROC.SYSTS_UPDATE»

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

«SYSPROC.SYSTS_REMOVE», à la page 73

Vous pouvez supprimer les index orphelins à l'aide de la procédure stockée SQL SYSPROC.SYSTS_REMOVE.

SYSPROC.SYSTS_UPDATE

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

Dans la mesure où la mise à jour de l'index de recherche de texte est une opération extensive, l'index de recherche de texte n'est pas mis à jour de manière synchrone quand la table DB2 est mise à jour. Au lieu de cela, les modifications de la colonne de table DB2 sont capturées par des déclencheurs et placées dans une table de transfert. L'index de recherche de texte est mis à jour lors de l'appel suivant à la procédure stockée SYSPROC.SYSTS_UPDATE ou quand l'option UPDATE FREQUENCY déclenche cette mise à jour. Il en découle que certaines requêtes de recherche ne prennent pas en compte les mises à jour récentes de la table.

Cette procédure stockée renvoie un résultat uniquement quand OmniFind Text Search Server for DB2 for i a terminé le processus de mise à jour de l'index de recherche de texte. Le délai d'attente dépend du nombre d'entrées contenues dans la table de transfert au moment de l'appel de la procédure SYSTS_UPDATE. Vous pouvez continuer d'interroger l'index de recherche de texte pendant le processus de mise à jour.

Si un incident se produit pendant l'indexation d'un document dans la table de base, la valeur de la colonne de la table de transfert TOBEDELETED devient E (error/erreur) ou W (warning/avertissement). Vous pouvez extraire ces enregistrements avec une procédure stockée qui contient la requête suivante :

```
SELECT s.TOBEDLETED, colonnes de base
FROM colonne de base t INNER JOIN QSYS2.tables de transfert
ON (QQQ_TEXTSEARCH_KEY(t.k1, t.k2, t.k3, ...) = s.KEYID)
WHERE s.TOBEDLETED IN('E','W')
```

Dans ce cas, colonnes de base indique la liste des colonnes que vous voulez voir dans la table de base. table de base indique la table en cours d'indexation. table de transfert indique la table de transfert répertoriée dans les catalogues pour l'index de recherche de texte. k1, k2, k3, ... correspond aux colonnes clé dans la clé primaire, l'ID de ligne ou la clé unique à utiliser pour construire l'index de recherche de texte. Une fois que vous avez corrigé les erreurs dans ces documents, exécutez à nouveau la mise à jour.

Si une mise à jour de l'index est demandée alors qu'une mise à jour est déjà en cours pour le même index, un erreur est générée. Il ne peut s'exécuter qu'une seule mise à jour à la fois pour le même index.

| Conditions préalables

| Avant d'appeler la procédure stockée SYSPROC.SYSTS_UPDATE, vérifiez que les prérequis suivants sont satisfaits :

- | • L'index de recherche de texte a été créé (en appelant la procédure stockée SYSPROC.SYSTS_CREATE).
- | • Les procédures stockées suivantes ne sont pas en cours d'exécution pour l'index de recherche de texte que vous voulez mettre à jour : SYSPROC.SYSTS_CREATE, SYSPROC.SYSTS_UPDATE et SYSPROC.SYSTS_DROP.
- | • Le serveur de recherche de texte sur lequel réside l'index a été démarré avec la procédure stockée SYSTS_START. La colonne SERVERSTATUS dans QSYS2.SYSTEXTSERVERS a la valeur '0' (démarré).

| Autorisation

| Les privilèges associés à l'ID de l'émetteur de l'instruction doivent comprendre au moins un des privilèges suivants :

- | • Le privilège de créer le schéma. Pour plus d'informations, voir Autorisation, privilèges et propriété des objets.
- | • Droits d'administration

| Les privilèges associés à l'ID de l'émetteur de l'instruction doivent comprendre au moins un des privilèges suivants :

- | • Les droits système suivants :
 - | – *USE, sur la commande Create Logical File (CRTLF)
 - | – *CHANGE, sur le dictionnaire de données si la bibliothèque dans laquelle l'index de texte est créé est un schéma SQL contenant un dictionnaire de données
- | • Droits d'administration

| Les privilèges associés à l'ID de l'émetteur de l'instruction doivent aussi comprendre au moins un des privilèges suivants :

- | • Pour la table référencée :
 - | – Le privilège INDEX sur la table
 - | – Le droit système *EXECUTE sur la bibliothèque contenant la table
- | • Droits d'administration
- | • Si des noms SQL sont spécifiés, qu'il existe un profil utilisateur portant le même nom que la bibliothèque dans laquelle l'index de texte est créé et que ce nom est différent de l'ID de l'émetteur de l'instruction, les privilèges associés à cet utilisateur doivent inclure au moins l'un des privilèges suivants :
 - | – Droit système *ADD sur le profil utilisateur portant ce nom
 - | – Droits d'administration

| Si un autre type distinct est référencé, les privilèges associés à l'ID de l'émetteur de l'instruction doivent comprendre au moins un des privilèges suivants :

- | • Pour chaque type distinct identifié dans l'instruction :
 - | – le privilège USAGE sur le type distinct, et
 - | – le droit système *EXECUTE sur la bibliothèque qui contient le type distinct
- | • Droits d'administration

| Pour plus d'informations sur les droits système correspondant aux privilèges SQL, voir GRANT (privilèges de table ou de vue).

Syntaxe

SYSTS_UPDATE(schéma_index, nom_index, options)

Diagram showing the syntax of the SYSTS_UPDATE procedure. The first parameter is enclosed in a box labeled 'schéma_index' with a 'null' option below it. The second parameter is labeled 'nom_index' and the third is labeled 'options'. The entire procedure call is enclosed in a box with arrows at both ends.

Le qualificateur de schéma est SYSPROC.

Paramètres

schéma_index

Identifie le schéma de l'index de recherche de texte. Si la valeur de ce paramètre est nulle, la valeur du registre spécial CURRENT SCHEMA utilisé pour l'auteur de l'appel est utilisée.

Le type de données de ce paramètre est VARCHAR(128).

nom_index

Identifie le nom de l'index de recherche de texte. Le nom de l'index de recherche de texte associé au schéma d'index identifie de manière unique l'index de recherche en texte intégral dans le sous-système DB2. Vous devez indiquer une valeur non nulle pour ce paramètre.

Le type de données de ce paramètre est VARCHAR(128).

options

Chaîne de caractères qui indique les options disponibles pour cette procédure stockée.

La seule option disponible est USING UPDATE MINIMUM. Cette option utilise les paramètres de USING UPDATE MINIMUM que vous avez spécifiés pour la procédure stockée SYSPROC.SYSTS_CREATE. Elle démarre une mise à jour incrémentielle uniquement si le nombre de modifications spécifié a été atteint. Par défaut, le processus de mise à jour démarre sans condition.

USING UPDATE MINIMUM :

USING UPDATE MINIMUM

Diagram showing the USING UPDATE MINIMUM option. The text 'USING UPDATE MINIMUM' is enclosed in a box with arrows at both ends.

Concepts associés

«Troncature des documents», à la page 15
OmniFind Text Search Server for DB2 for i limite le nombre de caractères pouvant être indexés pour chaque document texte. Il arrive que cette limite occasionne la troncature des grands documents texte dans l'index de recherche de texte.

Référence associée

«SYSPROC.SYSTS_CREATE», à la page 19
Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte. Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.
«SYSPROC.SYSTS_START», à la page 16
Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.
«SYSPROC.SYSTS_DROP», à la page 33
Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_DROP pour supprimer un index de recherche de texte créé avec la procédure stockée SYSPROC.SYSTS_CREATE.

Recherche avec un index de recherche de texte

Vous pouvez utiliser la fonction CONTAINS et la fonction SCORE dans une instruction SQL avec OMNIFIND. Les fonctions CONTAINS et SCORE permettent de rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. Vous pouvez également indexer et rechercher des documents XML.

CONTAINS

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

►► CONTAINS (—nom_colonne—, —argument_recherche—, —constante chaîne—) (1)

Remarques :

1 constante chaîne doit respecter les règles définies pour les options de argument_recherche.

options_argument_recherche :

(1)

QUERYLANGUAGE	=	—valeur—
RESULTLIMIT	=	—valeur—
SYNONYM	=	OFF ON

Remarques :

1 Vous ne pouvez pas spécifier une même clause plus d'une fois.

Le schéma est QSYS2.

nom_colonne

Indique le nom qualifié ou non qualifié d'une colonne contenant un index de recherche de texte à

rechercher. La colonne doit exister dans la table ou dans la vue identifiée dans la clause FROM de l'instruction. La colonne de la table, ou la colonne de la table de base sous-jacente de la vue, doit contenir un index de recherche de texte associé (SQLSTATE 38H12). L'expression sous-jacente de la colonne d'une vue doit être une simple référence à la colonne d'une table sous-jacente, directement ou via une autre vue imbriquée.

argument_recherche

Indique une expression qui renvoie une valeur de chaîne contenant les termes utilisés dans la recherche. L'expression ne doit pas contenir uniquement des espaces ou une chaîne vide (SQLSTATE 38H14). La longueur de la chaîne ne doit pas dépasser 32704 octets. Cette longueur peut aussi être limitée par les capacités de prise en charge du serveur de recherche de texte (SQLSTATE 38H10). La valeur est convertie dans le format Unicode avant d'être utilisée pour rechercher l'index de recherche de texte.

constante chaîne

Désigne une constante de type chaîne qui indique les options de critères de recherche appliquées à la fonction.

Les options qui peuvent être spécifiées avec la variable *options_argument-recherche* sont les suivantes :

QUERYLANGUAGE=*valeur*

Indique la langue de requête. Cette valeur peut indiquer n'importe lequel des codes de langue pris en charge. Si l'option QUERYLANGUAGE n'est pas spécifiée, la valeur par défaut est la valeur de langue de l'index de recherche de texte utilisé lors de l'appel de cette fonction. Si la valeur de langue de l'index de recherche de texte est AUTO, la valeur par défaut de QUERYLANGUAGE est en_US.

RESULTLIMIT=*valeur*

Indique le nombre maximum de résultats que peut renvoyer le moteur de recherche sous-jacent. La variable *valeur* peut contenir une valeur de type entier comprise entre 1 et 2 147 483 647. Si l'option RESULTLIMIT n'est pas spécifiée, aucune limite de résultat ne s'applique à la requête.

Cette fonction scalaire n'est pas nécessairement appelée pour chaque ligne de la table des résultats, selon le plan choisi par l'optimiseur. Cette fonction peut être appelée une fois pour toute la requête envoyée au moteur de recherche sous-jacent. Un jeu de résultats contenant toutes les clés primaires correspondantes est renvoyé. Ce jeu de résultats est ensuite joint à la table contenant la colonne afin d'identifier les lignes de résultat. Dans ce cas, la valeur RESULTLIMIT agit comme l'expression FETCH FIRST ?? ROWS du moteur de recherche de texte sous-jacent et peut être utilisée comme optimisation. Si le moteur de recherche est appelé pour chaque ligne du résultat (car c'est le meilleur plan), l'option RESULTLIMIT est ignorée.

SYNONYM = OFF ou SYNONYM = ON

Indique s'il faut utiliser un dictionnaire de synonymes associé à l'index de recherche de texte. Vous pouvez ajouter un dictionnaire de synonymes à une collection avec l'utilitaire Synonym.

OFF OFF est la valeur par défaut.

ON Indique d'utiliser le dictionnaire de synonymes associé à l'index de recherche de texte.

La fonction a pour résultat une valeur entière élevée. Si le deuxième argument peut être nul, le résultat peut être nul. Si le deuxième argument est nul, le résultat est une valeur nulle.

Le résultat est égal à 1 si le document contient une entrée correspondant aux critères de recherche spécifiés dans l'argument de recherche. Sinon, le résultat est égal à 0. Le résultat est aussi égal à 0 si la colonne est nulle. Si l'argument de recherche a la valeur Null, le résultat est une valeur nulle.

CONTAINS est une fonction non déterministe.

| Exemple 1

| L'instruction suivante recherche tous les employés dont le descriptif contient la chaîne △COBOL△.

```
| SELECT EMPNO
| FROM EMP_RESUME
| WHERE RESUME FORMAT = 'ascii'
| AND CONTAINS(RESUME, 'COBOL') = 1
```

| Exemple 2

| L'argument de recherche n'est pas obligatoirement une constante de type chaîne. L'argument de recherche peut être toute expression de chaîne SQL, par exemple une chaîne contenue dans une variable hôte. L'instruction suivante recherche exactement la chaîne "ate" dans la colonne COMMENT.

| **Remarque :** La chaîne "ate" doit être délimitée par des doubles guillemets pour être recherchée de manière exacte et que les variations linguistiques ne soient pas prises en compte.

```
| char search_arg[100]; /* input host variable */
| ...
| EXEC SQL DECLARE C3 CURSOR FOR
| SELECT CUSTKEY
| FROM K55ADMIN.CUSTOMERS
| WHERE CONTAINS(COMMENT, :search_arg)= 1
| ORDER BY CUSTKEY;
| strcpy(search_arg, "\"ate\"");
| EXEC SQL OPEN C3;
| ...
```

| Exemple 3

| L'instruction suivante recherche au hasard 10 étudiant ayant rédigé des documents en ligne contenant les termes "carburant fossile" en espagnol, c'est-à-dire "combustible fósil". Ces étudiant seront invités à une émission de radio. Utilisez le dictionnaire de synonymes créé pour l'index de recherche de texte associé. Dans la mesure où nous avons besoin de 10 étudiants seulement, optimisez la requête en utilisant l'option RESULTLIMIT afin de limiter le nombre de résultats renvoyés par le serveur de recherche de texte sous-jacent.

```
| SELECT FIRSTNME, LASTNAME
| FROM STUDENT_ESSAYS
| WHERE CONTAINS(TERM_PAPER, 'combustible fósil',
| 'QUERYLANGUAGE= es_ES RESULTLIMIT = 10 SYNONYM=ON') = 1
```

Tâches associées

«Recherche d'un index de recherche de texte», à la page 15

Vous pouvez rechercher un index de recherche de texte à l'aide d'une instruction SQL avec une fonction CONTAINS ou SCORE. Les critères de l'argument de recherche sont spécifiés avec la fonction.

Référence associée

«SCORE»

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

«Syntaxe des arguments de recherche», à la page 43

Vous pouvez indiquer un argument de recherche comme condition pour rechercher des termes dans des documents texte. Pour cela, vous indiquez des paramètres de recherche et un ou plusieurs termes de recherche. Les fonctions de recherche de texte SQL scalaires qui utilisent des arguments de recherche sont CONTAINS et SCORE.

SCORE

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

(1)

SCORE(—*nom_colonne*—,—*argument_recherche*—,—*constante chaîne*—)

Remarques :

1 *constante chaîne* doit respecter les règles définies pour les options de *argument_recherche*.

options_argument_recherche :

(1)



Remarques :

1 Vous ne pouvez pas spécifier une même clause plus d'une fois.

Le schéma est QSYS2.

nom_colonne

Indique le nom qualifié ou non qualifié d'une colonne contenant un index de recherche de texte à rechercher. La colonne doit exister dans la table ou dans la vue identifiée dans la clause FROM de l'instruction. La colonne de la table, ou la colonne de la table de base sous-jacente de la vue, doit contenir un index de recherche de texte associé (SQLSTATE 38H12). L'expression sous-jacente de la colonne d'une vue doit être une simple référence à la colonne d'une table sous-jacente, directement ou via une autre vue imbriquée.

argument_recherche

Indique une expression qui renvoie une valeur de chaîne contenant les termes utilisés dans la recherche. L'expression ne doit pas contenir uniquement des espaces ou une chaîne vide (SQLSTATE 38H14). La longueur de la chaîne ne doit pas dépasser 32704 octets. Cette longueur peut aussi être limitée par les capacités de prise en charge du serveur de recherche de texte (SQLSTATE 38H10). La

valeur est convertie dans le format Unicode avant d'être utilisée pour rechercher l'index de recherche de texte. Si l'argument de recherche est nul, le résultat est une valeur nulle.

constante chaîne

Désigne une constante de type chaîne qui indique les options de critères de recherche appliquées à la fonction.

Les options qui peuvent être spécifiées avec la variable *options_argument-recherche* sont les suivantes :

QUERYLANGUAGE=*valeur*

Indique la langue de requête. Cette valeur peut indiquer n'importe lequel des codes de langue pris en charge. Si l'option QUERYLANGUAGE n'est pas spécifiée, la valeur par défaut est la valeur de langue de l'index de recherche de texte utilisé lors de l'appel de cette fonction. Si la valeur de langue de l'index de recherche de texte est AUTO, la valeur par défaut de QUERYLANGUAGE est en_US.

RESULTLIMIT=*valeur*

Indique le nombre maximum de résultats que peut renvoyer le moteur de recherche sous-jacent. La variable *valeur* peut contenir une valeur de type entier comprise entre 1 et 2 147 483 647. Si l'option RESULTLIMIT n'est pas spécifiée, aucune limite de résultat ne s'applique à la requête.

Cette fonction scalaire n'est pas nécessairement appelée pour chaque ligne de la table des résultats, selon le plan choisi par l'optimiseur. Cette fonction peut être appelée une fois pour toute la requête envoyée au moteur de recherche sous-jacent. Un jeu de résultats contenant toutes les clés primaires correspondantes est renvoyé. Ce jeu de résultats est ensuite joint à la table contenant la colonne afin d'identifier les lignes de résultat. Dans ce cas, la valeur RESULTLIMIT agit comme l'expression FETCH FIRST ?? ROWS du moteur de recherche de texte sous-jacent et peut être utilisée comme optimisation. Si le moteur de recherche est appelé pour chaque ligne du résultat (car c'est le meilleur plan), l'option RESULTLIMIT est ignorée.

SYNONYM = OFF ou **SYNONYM = ON**

Indique s'il faut utiliser un dictionnaire de synonymes associé à l'index de recherche de texte. Vous pouvez ajouter un dictionnaire de synonymes à une collection avec l'utilitaire Synonym.

OFF OFF est la valeur par défaut.

ON Indique d'utiliser le dictionnaire de synonymes associé à l'index de recherche de texte.

La fonction a pour résultat une valeur double précision à virgule flottante. Si le deuxième argument peut être nul, le résultat peut être nul. Si le deuxième argument est nul, le résultat est une valeur nulle.

Le résultat est supérieur à 0 mais inférieur à 1 si la colonne contient une entrée correspondant aux critères de recherche spécifiés dans l'argument de recherche. Plus l'occurrence est détectée, plus la valeur du résultat augmente. Si la colonne ne contient aucune occurrence, le résultat est 0. Le résultat est aussi égal à 0 si la colonne est nulle.

SCORE est une fonction non déterministe.

Exemple

L'instruction suivante génère une liste des employés classés selon la manière dont leur descriptif contient les termes programmer AND (java OR cobol). De plus, une note de pertinence comprise entre 0 (zéro) et 100 est renvoyée.


```
| SELECT EMPNO, INTEGER(SCORE(RESUME, 'programmer AND
| (java OR cobol)') * 100) AS RELEVANCE
| FROM EMP_RESUME
| WHERE RESUME_FORMAT = 'ascii'
| ORDER BY RELEVANCE DESC
```

| **Tâches associées**

| «Recherche d'un index de recherche de texte», à la page 15
 | Vous pouvez rechercher un index de recherche de texte à l'aide d'une instruction SQL avec une fonction CONTAINS ou SCORE. Les critères de l'argument de recherche sont spécifiés avec la fonction.

| **Référence associée**

| «CONTAINS», à la page 38
 | Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.
 | «SYSPROC.SYSTS_CREATE», à la page 19
 | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte. Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.
 | «Syntaxe des arguments de recherche»
 | Vous pouvez indiquer un argument de recherche comme condition pour rechercher des termes dans des documents texte. Pour cela, vous indiquez des paramètres de recherche et un ou plusieurs termes de recherche. Les fonctions de recherche de texte SQL scalaires qui utilisent des arguments de recherche sont CONTAINS et SCORE.

| **Syntaxe des arguments de recherche**

| Vous pouvez indiquer un argument de recherche comme condition pour rechercher des termes dans des documents texte. Pour cela, vous indiquez des paramètres de recherche et un ou plusieurs termes de recherche. Les fonctions de recherche de texte SQL scalaires qui utilisent des arguments de recherche sont CONTAINS et SCORE.

| Pour tout traitement utilisant une langue spécifique au cours d'une recherche, vous pouvez indiquer une valeur pour le paramètre QUERYLANGUAGE en tant qu'option d'argument de recherche. Cette valeur peut indiquer n'importe lequel des codes de langue pris en charge.

| Si le paramètre QUERYLANGUAGE n'est pas spécifié, la valeur par défaut est la valeur de langue de l'index de recherche de texte utilisé lors de l'appel de cette fonction.

| Si la valeur de langue de l'index de recherche de texte est AUTO, la valeur par défaut de QUERYLANGUAGE est en_US.

| **Limitations**

| Vous ne pouvez pas utiliser les fonctions CONTAINS et SCORE dans une définition d'index ou une contrainte SQL. Vous pouvez les utiliser dans des instructions de requête SQL et afficher les définitions dans les limites suivantes :

- | • Si une vue, une expression de table imbriquée ou une expression de table commune contient une colonne de recherche de texte pour une fonction scalaire CONTAINS ou SCORE et que la vue concernée, l'expression de table imbriquée ou l'expression de table commune contient une clause DISTINCT dans la dernière instruction SELECT, la liste SELECT doit alors contenir tous les champs de clé correspondants de l'index de recherche de texte. Sinon, un message SQL 38H12 est renvoyé.
- | • Si une vue, une expression de table imbriquée ou une expression de table commune contient une colonne de recherche de texte pour une fonction scalaire CONTAINS ou SCORE, la vue concernée,

- | l'expression de table imbriquée ou l'expression de table commune ne pas pas contenir d'instruction UNION, EXCEPT ou INTERSECT au dernier niveau SELECT. Sinon, un message SQL 38H12 est renvoyé.
- | • Si une expression de table commune contient une colonne de recherche de texte pour une fonction scalaire CONTAINS ou SCORE, cette expression de table commune peut être référencée à nouveau dans toute la requête uniquement si la référence ne contient pas de colonne de recherche de texte pour une fonction scalaire CONTAINS ou SCORE. Sinon, un message SQL 38H12 est renvoyé.
- | • Vous ne pouvez pas créer une fonction ayant comme source des fonctions scalaires CONTAINS ou SCORE. Sinon, un message SQL SQL0457 est renvoyé.
- | • La requête peut s'exécuter via le moteur de requête SQL (SQE).

| Recherche Simple

| Pour exécuter une recherche de mot clé simple, entrez un ou plusieurs mots clés dans la requête. Le moteur de recherche renvoie les documents qui contiennent tous les mots clés ou des variantes de ces mots clés.

| Par exemple, si vous tapez king, le moteur de recherche renvoie tous les documents qui contiennent le mot king ou kings. Si vous tapez la requête king lear, le moteur de recherche renvoie les documents qui contiennent les termes king et lear.

| Pour obtenir des résultats plus précis, utilisez des mots clés plus spécifiques. Par exemple, entrez français brûlé café au lieu de simplement café, entrez voyage itinérant Kauai au lieu de vacances Hawaï.

| Si une recherche de mot-clé simple renvoie trop de documents qui ne répondent pas à votre recherche, vous pouvez utiliser des opérateurs pour affiner cette recherche.

| Exclusion de termes dans une recherche

| Utilisez le signe moins (-) pour exclure des termes. Par exemple, si vous recherchez des documents contenant le terme lear mais pas edward, tapez la requête lear -edward.

| Le signe moins (-) s'applique au terme comme à ses variantes. Par exemple, la requête -edward exclue les documents qui contiennent la chaîne edward's.

| Recherche d'expression

| Pour s'assurer que les termes apparaissent exactement dans l'ordre saisi, vous pouvez utiliser des doubles guillemets. Par exemple, pour voir des documents contenant les termes king lear exactement, mais pas des chaînes apparentées comme kingly lear ou king and queen lear, tapez "king lear". La recherche ne distingue pas les majuscules des minuscules mais les variantes des termes saisis ne sont pas considérés comme des occurrences.

| Utilisation de caractère générique dans une recherche

| Le caractère générique (*) vous permet de rechercher des documents dont vous ignorez l'intitulé exact, ou quand vous recherchez des variantes d'un terme. Par exemple, la requête améri* renvoie les documents contenant les termes Amérique, Américain, Américaine, Amérindien et d'autres résultats possibles.

| Vous pouvez également utiliser ce caractère générique dans une recherche d'expression. Par exemple, la requête "John * Kennedy" renvoie les documents contenant les termes John Fitzgerald Kennedy et John F Kennedy mais pas John Kennedy. La requête Mi*I Gorbachev renvoie Mikhail Gorbachev.

| Si vous ajoutez un caractère générique au début de la requête (par exemple, *méricain) il faudra plus de temps au moteur de recherche pour renvoyer les résultats.

| Recherche d'au moins un des termes saisis

| L'opérateur logique OR indique qu'au moins un des termes saisis dans une requête doit apparaître dans le document renvoyé. Par exemple, la requête (othello OR otello) renvoie les documents qui contiennent le terme othell ou otello.

| Vous pouvez également employer les opérateurs logiques AND, OR et NOT de manière combinée en utilisant des parenthèses. Par exemple, la requête cougar OR (jaguar AND NOT voiture) renvoie les documents contenant les termes cougar ou jaguar mais pas voiture.

| Vous devez entrer les opérateurs logiques AND, OR et NOT en majuscules. Utilisez les parenthèses pour les regrouper.

| Concepts associés

| «Recherche XML», à la page 48

| Vous pouvez indexer et rechercher des documents XML. La syntaxe des recherches XML utilise un sous-ensemble du langage W3 XPath avec des extensions développées pour la recherche de texte. Ces extensions permettent de rechercher des plages de valeurs numériques, de dates et d'horodates associées à un élément ou un attribut XML. Les éléments structuraux peuvent s'utiliser séparément ou de manière combinée avec du texte libre dans les requêtes.

| Référence associée

| «CONTAINS», à la page 38

| Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

| «SCORE», à la page 41

| Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

| Exemples de requête simple

| Les requêtes simples exécutées à l'aide des fonctions CONTAINS et SCORE recherchent un ou plusieurs mots dans un index de recherche de texte.

| Le moteur de recherche ignore les blancs présents entre les caractères. La chaîne de recherche ne doit pas être vide ou ne contenir que des blancs (SQLSTATE 38H14).

| Le tableau suivant contient des exemples de recherche simple.

| *Tableau 3. Exemples de requête simple*

Type de recherche de mot	Exemples	Résultats de la requête
Mot unique	king	Renvoie tous les documents qui contiennent le mot king ou kings. Cette requête isole différentes formes du mot recherché et ne distingue pas les majuscules des minuscules.
Plusieurs mots	king lear	Renvoie tous les documents qui contiennent les mots king et lear. L'opérateur par défaut est l'opérateur logique AND.

Les opérateurs **AND** et **+** sont implicites dans toutes les requêtes. Par exemple, la requête King Lear renvoie les mêmes résultats que King AND Lear ou King + Lear.

Vous devez entrer les opérateurs logiques **NOT**, **AND** et **OR** en majuscules.

Référence associée

«CONTAINS», à la page 38

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

«SCORE», à la page 41

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

Opérateurs de recherche avancée

Vous pouvez utiliser des opérateurs de recherche avancée pour affiner les résultats de la recherche avec les fonctions CONTAINS et SCORE.

Dans le tableau suivant, la première colonne contient l'opérateur que vous pouvez utiliser dans une requête de recherche. Vous devez entrer les opérateurs logiques NOT, AND et OR en majuscules. La deuxième colonne contient un exemple de requête que vous pouvez saisir. La troisième colonne indique les types de résultats que l'exemple de requête peut produire.

Tableau 4. Opérateurs de recherche avancée et exemples de requête complexe

Opérateurs	Exemples	Résultats de la requête
AND	"King Lear" AND "Othello" "King Lear" "Othello"	Les deux requêtes renvoient les documents qui contiennent à la fois King Lear et Othello. L'opérateur AND est l'opérateur de conjonction par défaut. En l'absence d'opérateur logique entre les deux termes, l'opérateur AND est utilisée. Par exemple, la requête King Lear est équivalente à la requête King AND Lear.
OU	"King Lear" OR Lear	Renvoie les documents qui contiennent King Lear ou simplement Lear. L'opérateur OR relie les deux termes et trouve une occurrence de document si l'un ou l'autre des deux termes recherchés figure dans le document.
NOT	"King Lear" NOT "Norman Lear"	Renvoie les documents qui contiennent King Lear mais pas Norman Lear.
" " (occurrence exacte)	Première requête : "King Lear" Deuxième requête : "king"	La première requête renvoie exactement la phrase King Lear. La deuxième requête renvoie uniquement le mot king mais pas d'autres formes comme kings ou kingly.
* (caractère générique)	test* te*t	Renvoie les documents pouvant correspondre à des combinaisons possibles, par exemple test, tests et tester, ou test et text.

Tableau 4. Opérateurs de recherche avancée et exemples de requête complexe (suite)

Opérateurs	Exemples	Résultats de la requête
\wedge (facteur de pondération de score) <i>mot ou phrase</i> ^{nombre}	Première requête : "King Lear" ⁴ "Richard III" Deuxième requête : title: (software download) ⁵ pdf viewer -shipping	Avec la première requête, les documents contenant l'expression King Lear apparaissent en haut de la liste des résultats de la recherche. Avec la deuxième requête, un document intitulé software download apparaît en haut de la liste de résultats. Un facteur de pondération est obligatoirement positif mais il peut être inférieur à 1, par exemple 0,2. La valeur du facteur de pondération n'a pas de limite supérieure.
+ (inclure)	+Lear King	Renvoie tous les documents qui contiennent Lear et King (équivalent à la requête Lear AND King).
- (exclure)	"King Lear" -"Lear Jet"	Renvoie les documents qui contiennent King Lear mais pas Lear Jet.
()	(King OR Lear) AND plays	Renvoie les documents qui contiennent King ou Lear et plays. Les parenthèses obligent à trouver plays et à trouver au choix King ou Lear.
\ (caractère d'échappement)	\(1+1\)\:2	Renvoie les documents qui contiennent (1+1):2. Utilisez le symbole "\" pour effacer les caractères spéciaux figurant dans la syntaxe de la requête. Les caractères spéciaux sont les suivants : +, -, &, , !, (,), {, }, [,], ^, ", ~, *, ?, :, et \. Si un caractère spécial est effacé, il est analysé en tant qu'élément de la requête.

Référence associée

«CONTAINS», à la page 38

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

«SCORE», à la page 41

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

Exemple d'utilisation des fonctions CONTAINS et SCORE

Vous pouvez utiliser les fonctions CONTAINS et SCORE dans une même requête. La requête recherche un index de recherche de texte et indique si et à quelle fréquence le document texte correspond aux critères de l'argument de recherche.

Dans l'exemple de la table suivante, les données de la table de base BOOKS sont utilisées avec les colonnes ISBN (VARCHAR(20)), ABSTRACT (VARCHAR(10000)), et PRICE (INTEGER).

Tableau 5. Table de base BOOKS

ISBN	ABSTRACT	PRICE
i1	"a b c"	7
i2	"a b d"	10
i3	"a e a"	8

Vous exécutez la requête suivante :

```
SELECT ISBN, SCORE(ABSTRACT, 'b')
FROM BOOKS
WHERE CONTAINS (ABSTRACT, 'b') = 1
```

Cette requête renvoie les deux lignes suivantes :

```
i1, 0.3
i3, 0.4
```

Les valeurs de notation peuvent différer selon le contenu de la colonne de texte.

Référence associée

«CONTAINS», à la page 38

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

«SCORE», à la page 41

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

Recherche XML

Vous pouvez indexer et rechercher des documents XML. La syntaxe des recherches XML utilise un sous-ensemble du langage W3 XPath avec des extensions développées pour la recherche de texte. Ces extensions permettent de rechercher des plages de valeurs numériques, de dates et d'horodates associées à un élément ou un attribut XML. Les éléments structurels peuvent s'utiliser séparément ou de manière combinée avec du texte libre dans les requêtes.

Les documents doivent être indexés de manière à inclure le marquage XML pour que vous puissiez ensuite interroger l'index avec la syntaxe de recherche xmlxp. L'indexation des documents se fait avec l'option "FORMAT XML" au moment de la création de l'index.

Vous pouvez utiliser les index créés dans une édition précédente du produit pour exécuter les recherches. Toutefois, les documents indexés dans une édition précédente ne contiennent pas les informations requises pour utiliser toutes les fonctions de recherche XML disponibles dans les éditions suivantes. Les documents ajoutés ou mis à jour dans l'index de recherche de texte après la mise à niveau vers la nouvelle édition contiennent de nouvelles informations.

Après la mise à niveau, les documents indexés dans une édition antérieure ne seront pas pris en compte dans certains résultats de recherche. Pour résoudre ce problème, vous pouvez utiliser la procédure stockée SYSPROC.SYSTS_REPRIMEINDEX afin de régénérer l'index.

Pour utiliser les fonctions définies par l'utilisateur d'OMNIFIND CONTAINS et SCORE afin de rechercher des données XML, la chaîne de requête doit commencer par le préfixe de requête @xmlxp:. Le préfixe est suivi d'une expression de requête de recherche XML. Le préfixe de terme opaque @xmlxp indique qu'une recherche doit être effectuée avec l'expression de chemin de requête.

Par exemple : CONTAINS(nom_colonne, '@xmlxp:'expression_requête').

| Les simples guillemets ' ' qui encadrent `expression_requête` doivent être doublés car ils sont contenus dans une chaîne SQL (une chaîne dans une chaîne).

| Le préfixe de terme opaque `@xpath:` qui était utilisé dans les éditions précédentes d'OmniFind Text Search Server for DB2 for i est pris en charge pour assurer la compatibilité avec les versions antérieures. Notez toutefois qu'il est à présent obsolète et que son utilisation est déconseillée pour cette raison.

| La liste suivante décrit les caractéristiques principales de la recherche XML :

| Recherche structurelle XML

| Quand vous ajoutez des termes XML opaques spéciaux dans des requêtes, vous pouvez rechercher dans des document XML des éléments structurels et les textes délimités par ces éléments. Les éléments structurels sont les noms de balise, les noms d'attribut et les valeurs d'attribut. Les noms de balise et d'élément distinguent les majuscules des minuscules.

| Marquage sémantique des requêtes XML

| Le marquage sémantique consiste à analyser la syntaxe des entrées puis à les convertir en jetons. Le texte libre figurant dans les termes de requête XML est segmenté de la même manière que le texte figurant dans des termes de requête non XML. Notez toutefois que les termes opaques imbriqués ne sont pas pris en charge. La recherche de texte libre ne distingue pas les majuscules des minuscules.

| DTD et schéma XML

| Les schémas XML associés aux documents XML ne sont pas téléchargés et les valeurs par défaut ne sont pas indexées.

| Valeurs numériques

| Les prédicats qui servent à comparer les valeurs d'élément ou d'attribut sont pris en charge.

| Valeurs d'élément

| Les prédicats qui servent à comparer les valeurs d'élément à des nombres ou à des dates sont pris en charge. L'élément contenant la date ou le nombre doit être un élément XML contenant uniquement cette date ou ce nombre. Les espaces de début et de fin sont ignorés.

| Valeurs de chaîne

| L'utilisation de l'opérateur `=` comme argument de chaîne dans un prédicat requiert une correspondance totale de tous les mots clé de la chaîne de jetons dans la zone de texte identifiée. L'ordre des jetons n'a pas d'importance lors de la recherche des occurrences.

| Valeurs d'horodate

| Les prédicats qui servent à comparer des éléments ou attributs de date et d'horodate sont pris en charge.

| Expressions de chemin :

| *Tableau 6. Expressions de chemin*

@xmlxp expression	Description
nom_balise	Sélectionne une balise nommée <code>nom_balise</code> et les enfants de cette balise.
@nom_attribut	Sélectionne un attribut nommé <code>@nom_attribut</code> .

Tableau 6. Expressions de chemin (suite)

@xmlxp expression	Description
/	Sélectionne les données à partir du noeud racine.
//	Sélectionne les occurrences de balise et d'attribut situées après la position courante qui correspondent à l'expression.
.	Self: noeud d'élément et de balise courant

Tableau 7. Exemples d'expression de chemin :

Expression @xmlxp	Résultat
/Document	Renvoie tous les documents assortis d'une balise Document de niveau supérieur.
//Document	Renvoie tous les documents assortis d'une balise Document située à n'importe quel niveau.
/Document/Child1	Renvoie tous les documents assortis d'une balise Document de niveau supérieur et d'une balise enfant directe Child1.
/Document//Child1	Renvoie tous les documents assortis d'une balise Document de niveau supérieur et d'une balise descendante Child1 située à n'importe quel niveau.
/Root/@attr1	Renvoie tous les documents assortis d'une balise Root de niveau supérieur et d'un attribut attr1.
/Root//@attr1	Renvoie tous les documents assortis d'une balise Root de niveau supérieur et d'un attribut attr1 dans cette balise racine ou dans n'importe quelle balise descendante.
//@attr1	Renvoie tous les documents assortis d'un attribut @attr1 situé à n'importe quel niveau.

Remarque : L'expression de recherche XML doit contenir une balise réelle ou un nom d'attribut dans l'expression de chemin relatif. Les caractères / et // employés seuls ne constituent pas des requêtes de recherche valides.

Les expressions de chemin ne sont autorisées qu'en aval et exclusivement sur un même axe.

Prise en charge des caractères génériques dans les expressions de chemin

Vous pouvez utiliser le caractère générique spécial * dans une expression de chemin afin d'indiquer exactement une seule balise ayant n'importe quel nom.

Les caractères génériques placés à la fin d'une expression de chemin sont ignorés.

L'utilisation de caractères génériques dans une expression de chemin n'est pas prise en charge et occasionne une erreur quand :

- une expression référence uniquement des caractères génériques et ne spécifie ni éléments ni attributs,
- une expression référence un attribut générique à n'importe quel niveau (par exemple /Tag/@*),
- une expression référence un caractère générique qui précède immédiatement une expression de prédicat (par exemple /Root/*[//anytag]),
- une expression référence un caractère générique dans une comparaison de prédicats (par exemple /Root[* > 5]),

- une expression référence un caractère générique en tant que préfixe d'espace de noms XML (par exemple `//*:nom_balise`),
- une expression référence un caractère générique précédé d'un préfixe d'espace de noms XML (par exemple `//ns:*`),
- une expression référence un caractère générique dans un nom de balise (par exemple `/start*`).

Tableau 8. Exemples de caractères génériques utilisés dans les expressions de chemin

Expression @xmlxp	Résultat
<code>/Root/*/T1</code>	Renvoie tous les documents contenant une balise Root de niveau supérieur et une balise T1 descendante avec un seul niveau intermédiaire.
<code>/Root/*/T1</code>	Renvoie tous les documents contenant une balise Root de niveau supérieur et une balise T1 descendante avec un ou plusieurs niveaux intermédiaires.

Prédicats

Les prédicats servent à indiquer une valeur ou une condition qu'un élément ou un attribut doit satisfaire. Les prédicats sont toujours encadrés par des crochets (`[]`).

Tableau 9. Exemples de prédicat :

Expression @xmlxp	Résultat
<code>/Book[Sentences]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et une balise enfant directe Sentences.
<code>/Book[.//Sentences and .//Author]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et les balises descendantes Sentences et Author. Chaque balise descendante peut résider à n'importe quel niveau au-dessous de Book.

Dans la mesure où les expressions de chemin s'appliquent toujours en aval et sont limitées à un seul accès, les expressions de chemin figurant dans les prédicats doivent être relatives au noeud courant. `/Book[/Root]` et `/Book[/Root]` ne sont pas des expressions de chemin valides car, dans les deux cas, elles commencent avec une balise 'Root' au niveau supérieur au lieu du noeud courant.

Comparaisons de valeurs numériques

OMNIFIND prend en charge les opérateurs `=`, `<=`, `>=`, `>`, `<` et `!=` pour comparer des éléments et des attributs à des entiers et des valeurs à virgule flottante.

Seules les valeurs numériques des éléments sont indexées s'il s'agit d'éléments simples. Ils ne doivent pas contenir d'autres caractères (en dehors des blancs) ou d'éléments descendants. Les éléments complexes sont indexés en tant que texte uniquement.

Tableau 10. Exemples de comparaison numérique

Expression @xmlxp	Résultat
<code>/Book[@id_num = 12345]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et un attribut id_num ayant la valeur 12345.
<code>/Book[Cost <= 100.50]</code>	Renvoie les documents contenant une balise de niveau supérieur Book si Book possède un élément enfant direct Cost dont la valeur est inférieure ou égale à 100.50.

Comparaisons de dates et d'horodates

OMNIFIND prend en charge les opérateurs =, <=, >=, >, < et != pour comparer des éléments et des attributs à des valeurs de date et d'horodate.

Pour les éléments simples, seules les valeurs d'horodate sont indexées. Les éléments ne doivent pas contenir d'autres caractères (en dehors des blancs) ou d'éléments descendants. Les éléments complexes sont indexés en tant que texte uniquement.

Pendant l'indexation, les valeurs d'attribut et le texte contenus dans des balises XML simples sont analysés. Si le texte correspond à un format de date ou d'horodate ISO, il est indexé en tant que date ou horodate pouvant être recherchées dans un prédicat.

Pendant une recherche, une valeur de date ou d'horodate doit être encadrée dans un appel de fonction xs:date() ou xs:dateTime() pour être reconnue comme un type de données valide.

Des données XML de type date ou horodate contenues dans un document XML peuvent indiquer un fuseau horaire. Toutefois, quand une horodate est indexée, le serveur de recherche de texte tronque les valeurs du fuseau horaire au cours de l'indexation. Il en découle que les fuseaux horaire ne sont pas pris en compte lors des recherches XML impliquant des données de type date ou horodate.

De plus, la valeur de l'heure d'une horodate ne peut être 24 que si les minutes et les secondes ont la valeur 0 (zéro). Cette horodate (24:00:00) sera considérée comme une valeur située entre le dernier instant du jour écoulé et le premier instant du jour suivant.

Quand une valeur de date ou d'horodate est indiquée dans un prédicat de recherche XML, une erreur de syntaxe survient si un fuseau horaire est spécifié dans la valeur.

Le type de données horodate prend en charge jusqu'à 12 chiffres pour les fractions de seconde.

Tableau 11. Exemples de comparaison de dates et d'horodates

Expression @xmlxp	Résultat
/Book[@publishDate > xs:date("2000-01-01")]	Renvoie les documents contenant une balise de niveau supérieur Book si Book est assorti d'un attribut publishDate dont la valeur est postérieure à la date 2000-01-01.
/Book[purchaseTime > xs:dateTime("2009-05-20T13:00:00")]	Renvoie les documents contenant une balise de niveau supérieur Book si Book possède un enfant direct purchaseTime dont la valeur est une expression d'horodate postérieure à l'horodate 2009-05-20T13:00:00.000000.

Utilisation de *contains* et *excludes* dans le marquage XML

Les fonctions *contains* et *excludes* permettent d'effectuer des recherches en texte intégral dans le marquage XML. La fonction *contains* renvoie la valeur "true" si la chaîne de requête est présente dans le noeud cible. La fonction *excludes* renvoie la valeur "true" si la chaîne de requête est absente dans le noeud cible.

Par exemple, pour rechercher tous les documents assortis d'une balise email au niveau supérieur et d'un descendant direct nommé body qui contient des variantes de la chaîne "Department budget", entrez :

```
@xnkxo: '/email[body contains ("department budget")]'
```

Le texte libre communiqué à la fonction *contains* ou *excludes* est géré de la même manière que pour les autres recherches de texte libre. La recherche ne distingue pas les majuscules des minuscules et les

variantes linguistiques des termes saisis sont prises en compte. Ainsi, la requête de l'exemple précédent isole la chaîne "departments budgets" mais aussi "budget for the department".

La recherche peut être limitée aux occurrences exactes en utilisant les guillemets traditionnels, par exemple `@xmlxp: '/email[body contains("department budget")] '`. Les guillemets indiquant une recherche d'occurrence exacte doivent être doublés pour ne pas être interprétés comme faisant partie de la chaîne de texte libre passée à la fonction *contains*.

Tableau 12. Exemples d'utilisation de *contains* et *excludes*

Expression @xmlxp	Résultat
<code>/Book[abstract contains("cat AND dog")]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et une balise enfant abstract, qui contiennent des variantes linguistiques des mots cat et dog.
<code>/Book[abstract contains("cat AND dog")] /Book/@title[. contains("cat OR dog")]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et un attribut Title, qui contiennent des variantes linguistiques des mots cat ou dog.
<code>/Book/Title[. contains("All good dogs go to heaven")]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et un enfant direct Title, qui contiennent la chaîne all good dogs go to heaven dans cet ordre, sans prendre en compte les variantes linguistiques.
<code>/Book[abstract excludes("cat AND dog")]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et une balise enfant abstract, qui ne contiennent pas de variantes linguistiques des mots cat et dog.

Opérateur de recherche de chaîne complète

L'utilisation de l'opérateur = avec un argument de chaîne dans un prédicat requiert une correspondance totale entre tous les jetons de la chaîne et tous les jetons dans la zone de texte identifiée. Les équivalences linguistiques ne sont pas prises en compte. L'ordre des termes recherchés n'a pas d'importance. Il n'est pas obligatoire que l'élément ou l'attribut contienne uniquement le texte recherché.

Tableau 13. Exemples d'utilisation d'un opérateur de recherche de chaîne complète

Expression @xmlxp	Résultat
<code>/Book[@author = "Nicholas Lawrence"]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et un attribut author si author contient les termes Nicholas Lawrence. Les variantes linguistiques de ces termes ne sont pas considérées comme des occurrences.
<code>/Book[author = "Nicholas Lawrence"]</code>	Renvoie les documents contenant une balise de niveau supérieur Book et un descendant direct author si author contient les termes Nicholas Lawrence dans cet ordre. Les variantes linguistiques de ces termes ne sont pas considérées comme des occurrences.

Opérateurs logiques

Vous pouvez utiliser les opérateurs logiques *and* et *or* dans des prédicats.

Tableau 14. Exemples d'utilisation des opérateurs logiques

Expression @xmlxp	Résultat
/Book[@author = ""Nicholas Lawrence""]/Price[. < 1000 and @unit = "dollars"]	Renvoie les documents contenant une balise de niveau supérieur Book et un attribut author si author contient les termes Nicholas Lawrence dans cet ordre. Les variantes linguistiques de ces termes ne sont pas considérées comme des occurrences. Book doit posséder un enfant direct Price dont la valeur est < 1000. Le noeud Price doit posséder un attribut @unit dont la valeur est dollars.

Priorité des opérateurs

Dans les prédicats de recherche XML, les opérateurs de contenu et les opérateurs de comparaison ont priorité sur les opérateurs logiques et tous les opérateurs logiques ont le même niveau de priorité.

- Les opérateurs de contenu sont *contains* et *excludes*.
- Les opérateurs de comparaison sont =, !=, <, >, <= et >=.
- Les opérateurs logiques sont *and* et *or*.

Vous pouvez utiliser des parenthèses pour établir les priorités désirées.

Référence associée

«Syntaxe des arguments de recherche», à la page 43

Vous pouvez indiquer un argument de recherche comme condition pour rechercher des termes dans des documents texte. Pour cela, vous indiquez des paramètres de recherche et un ou plusieurs termes de recherche. Les fonctions de recherche de texte SQL scalaires qui utilisent des arguments de recherche sont CONTAINS et SCORE.

«SYSPROC.SYSTS_REPRIMEINDEX», à la page 75

La procédure stockée SYSPROC.SYSTS_REPRIMEINDEX permet de réorganiser l'index et de démarrer une mise à jour initiale. Vous pouvez utiliser cette procédure stockée quand vous souhaitez restaurer des données de la table de base.

«CONTAINS», à la page 38

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

«SCORE», à la page 41

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

Prise en charge de l'espace de noms de recherche XML

Vous pouvez utiliser un espace de nom pour délimiter les éléments et les attributs dans un document. Les espaces de noms permettent de limiter la requête de recherche aux éléments pertinents d'un document.

Présentation

Dans le langage XML, les noms des éléments et des attributs sont choisis par le développeur. Ces noms peuvent engendrer des conflits quand des documents XML issus de différentes applications sont mélangés.

| Il est donc utile de limiter la requête de recherche aux éléments pertinents dans le document,
| spécialement quand différents types de documents peuvent être indexés. Pour limiter ces recherches, vous
| pouvez utiliser des espaces de nom.

| Les espaces de noms permettent de délimiter la portée la éléments et des attributs du document pour
| garantir une interprétation appropriée des valeurs. Les espaces de noms sont définis avec un nom long
| (URI) et, éventuellement, un nom abrégé appelé le QName (nom qualifié).

```
| <?xml version='1.0'?>  
|   <doc xmlns:x="http://example.com/ns/abc">  
|     <x:p/>  
|   </doc>
```

| http://example.com/ns/abc est le nom d'espace de noms long et x est le préfixe du QName. Le préfixe du
| QName est utilisé comme forme abrégée de l'espace de noms de chaque référence d'élément.

| L'élément p est qualifié par l'espace de noms http://example.com/ns/abc.

| Espace de nom par défaut

| Vous pouvez indiquer un espace de noms par défaut pour les éléments XML. L'espace de noms par
| défaut s'applique à la balise courante et à ses balises descendantes. Une balise non qualifiée dans l'espace
| de noms hérite de l'espace de noms par défaut.

```
| <?xml version='1.0'?>  
|   <doc xmlns="http://example.com/ns/abc">  
|     <p/>  
|   </doc>
```

| Dans ce cas, les éléments doc et p se trouvent tous les deux dans l'espace de noms http://example.com/
| ns/abc.

| Espaces de noms des attributs

| Un attribut peut avoir un espace de noms différent de celui de son élément associé.

| Élément et attribut, qualifié :

```
| <dog xmlns:an="http://example.org/animals" xmlns:sz="http://example.org/sizes">  
|   <an:breed sz:size="Medium">Mutt</an:breed>  
| </dog>
```

| Il existe une différence dans la manière dont les éléments et les attributs héritent d'un espace de noms
| quand cela n'est pas explicitement spécifié. Les éléments non qualifiés héritent de l'espace de noms par
| défaut de leur portée. Les attributs non qualifiés n'ont pas d'espace de nom.

| Élément et attribut, non qualifié :

```
| <dog xmlns:an="http://example.org/animals">  
|   <breed size="Medium">Mutt</an:breed>  
| </dog>
```

| Dans cet exemple, l'espace de noms de l'élément breed est http://example.org/animals. En revanche,
| l'attribut size n'est associé à aucun espace de nom.

| Pour plus d'informations sur les espaces de noms XML, consultez la documentation du W3C
| "Recommendation for Namespaces in XML" disponible sur le site du World Wide Web Consortium

| (W3C) (<http://www.w3.org>) .

| Préfixes de QName réservés

| Les préfixes de QName suivants sont réservés et ne doivent pas être utilisés pour qualifier des attributs ou des éléments définis par l'utilisateur : **xml**, **xs**, **xsi**, **fn**, **local**.

| **Utilisation des espaces de noms dans les recherches**

| Les espaces de noms des éléments par défaut et des préfixes de QName doivent être définis dans le prologue de la requête @xmlxp du terme de recherche.

| Dans cet exemple, un prologue mappe l'espace de noms *ns1* à l'URI "http://mycompany.com"
| `declare namespace ns1 = "http://mycompany.com";`

| Dans cet exemple, un prologue spécifie que tous les éléments non qualifiés sont qualifiés par l'URI "http://mycompany.com" :
| `declare default element namespace "http://mycompany.com"`

| Si une requête ne déclare aucun préfixe de QName d'espace de noms ou aucun espace de noms d'élément par défaut, les espaces de noms ne sont pas pris en compte dans la requête. Un nom d'attribut ou d'élément est considéré comme une occurrence s'il existe dans un espace de noms quelconque.

| Si un espace de noms par défaut ou un préfixe de QName est déclaré, les noms d'attribut ou d'élément sont considérés comme des occurrences uniquement s'ils existent dans l'espace de noms spécifié.

| La syntaxe
| `declare default element namespace "";`

| peut être utilisée pour indiquer que les balises non qualifiées n'existent dans aucun espace de noms.

| Les préfixes de QName utilisés dans la chaîne de recherche XML ne doivent PAS obligatoirement être identique aux préfixes de QName utilisés dans le document XML. Les correspondances reposent uniquement sur l'URI du nom long.

| **Exemples :**

| Limitez la recherche à l'attribut `attr` de l'élément `test` quand l'élément `test` est mappé à l'espace de noms "http://posample.org", et que `attr` n'existe dans aucun espace de noms. Utilisez l'espace de noms par défaut pour simplifier la syntaxe.

| `CONTAINS(myxmlcol, '@xmlxp:''declare default element namespace "http://myexample.org";
| /test[@attr > xs:date("2005-01-01")]''')`

| Limitez la recherche à l'attribut `attr` de l'élément `test` quand l'élément `test` est mappé à l'espace de noms "http://myexample.org". Utilisez une syntaxe d'espace de noms explicite avec le préfixe de QName `abc`.

| `CONTAINS(myxmlcol, '@xmlxp:''declare namespace abc = "http://myexample.org";
| /abc:test[@attr < xs:date("2009-01-01")]''')`

| Limitez la recherche aux éléments enfants `shipTo name` et `billTo name` de l'élément `purchaseOrder`, qui est explicitement mappé à l'espace de noms "http://myexample.org" avec le préfixe de QName `ns1`. Un espace de noms par défaut également défini ("http://mastsample.org"), s'applique aux éléments `shipTo`, `name` et `billTo`.

| `CONTAINS(myxmlcol, '@xmlxp:'' declare default namespace "http://mastsample.org";
| declare namespace ns1 = "http://posample.org"; /ns1:purchaseOrder[shipTo/name = "Jane"
| and billTo/name = "Jason"]''')`

| Limitez la recherche à l'attribut `name` (explicitement défini dans l'espace de noms "http://posample.org") de l'élément `shipTo` (dans l'espace de noms par défaut "http://mastsample.org"), qui est un enfant de l'élément `purchaseOrder` (explicitement défini dans l'espace de noms "http://posample.org"). L'espace de noms par défaut "http://mastsample.org" s'applique aux éléments `shipTo`, `billTo` et `name`.

```
| CONTAINS(myxmlcol, '@xmlxp:' declare default namespace "http://mastsample.org";
| declare namespace ns1 = "http://posample.org"; /ns1:purchaseOrder/shipTo[@ns1:name =
| "Jane" and billTo/name = "Jason"]''')
```

| Exemple de recherche XML

- | • Créez une table nommée XML_DOCUMENTS dans le schéma XMLTEST afin de stocker les documents XML :

```
| CREATE TABLE XMLTEST.XML_DOCUMENTS (ID INT, XML_DATA XML, PRIMARY KEY (ID));
```

- | • Créez un index de recherche de texte appelé XML_INDEX sur la colonne XML :

```
| call SYSPROC.SYSTS_CREATE('XMLTEST', 'XML_INDEX', 'XMLTEST.XML_DOCUMENTS(XML_DATA)', '');
```

- | • Insérez des documents XML :

```
| INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
| VALUES(1,
| ' <BOOK publication_date="2009-01-01">' ||
| ' <TITLE> OmniFind Text Search Server for DB2 </TITLE>' ||
| ' <ID_NUMBER> 1 ></ID_NUMBER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 1 </NUMBER>' ||
| ' <TITLE> Introduction </TITLE>' ||
| ' <ABSTRACT> Ce chapitre présente les fonctionnalités d'OmniFind
| for DB2 for IBM i </ABSTRACT>' ||
| ' </CHAPTER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 2 </NUMBER>' ||
| ' <TITLE> Création d'un index de recherche de texte </TITLE>' ||
| ' <ABSTRACT> Ce chapitre décrit comment créer un index de recherche de texte </ABSTRACT>' ||
| ' </CHAPTER>' ||
| ' </BOOK>');
```

```
| INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
| VALUES(2,
| ' <BOOK publication_date="2010-02-01">' ||
| ' <TITLE> Utilisation du type de données XML pour DB2 for IBM i </TITLE>' ||
| ' <ID_NUMBER> 2 ></ID_NUMBER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 1 </NUMBER>' ||
| ' <TITLE> Introduction </TITLE>' ||
| ' <ABSTRACT> Ce chapitre présente le type de données XML DB2 </ABSTRACT>' ||
| ' </CHAPTER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 2 </NUMBER>' ||
| ' <TITLE> Insertion de données XML dans une table DB2
| </TITLE>' ||
| ' <ABSTRACT> Ce chapitre décrit comment insérer des données XML dans une table DB2
| </ABSTRACT>' ||
| ' </CHAPTER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 3 </NUMBER>' ||
| ' <TITLE> Recherche de données XML </TITLE>' ||
| ' <ABSTRACT> Ce chapitre décrit comment interroger les données des colonnes XML
| avec les fonctions CONTAINS et SCORE </ABSTRACT>' ||
| ' </CHAPTER>' ||
| ' </BOOK>');
```

```
| INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
| VALUES(3,
| ' <BOOK xmlns="http://www.ibm.com/digital_media_library" |
| publication_date="2010-02-01">' ||
| ' <TITLE> Utilisation d'espaces de noms avec OmniFind Text Search Server for DB2 for IBM i
| </TITLE>' ||
| ' <ID_NUMBER> 2 </ID_NUMBER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 1 </NUMBER>' ||
| ' <TITLE> Introduction </TITLE>' ||
| ' <ABSTRACT> Ce chapitre décrit les espaces de noms XML </ABSTRACT>' ||
| ' </CHAPTER>' ||
```



```
|      ' <CHAPTER>' ||
|      ' <NUMBER> 2 </NUMBER>' ||
|      ' <TITLE> Utilisation des espaces de noms par défaut </TITLE>' ||
|      ' <ABSTRACT> Ce chapitre décrit comment utiliser un espace de noms dans une recherche XML
| </ABSTRACT>' ||
|      ' </CHAPTER>' ||
|      '</BOOK>');
|
| • Mettez à jour l'index :
| CALL SYSPROC.SYSTS_UPDATE('XMLTEST', 'XML_INDEX', '');
```

| **Exemples de requête**

| **Exemple 1:**

| Recherchez tous les documents contenant un élément racine BOOK avec un descendant direct TITLE qui contient le terme DB2.

```
| SELECT ID
| FROM XMLTEST.XML_DOCUMENTS
| WHERE CONTAINS(XML_DATA, '@xmlns:''/BOOK/TITLE[. contains("DB2")]'' ') = 1;
```

| Dans la mesure où aucun prologue d'espace de noms n'est spécifié dans le terme de recherche, aucun espace de noms n'est pris en compte dans la recherche.

| *Tableau 15. Résultat*

ID
1
2
3

| **Exemple 2:**

| Recherchez tous les documents contenant un élément racine BOOK avec un descendant direct TITLE qui contient le terme DB2. Utilisez un espace de noms d'élément par défaut pour indiquer que BOOK et TITLE doivent figurer dans l'espace de nom "http://www.ibm.com/digital_media_library".

```
| SELECT ID
| FROM XMLTEST.XML_DOCUMENTS
| WHERE CONTAINS(XML_DATA, '@xmlns:''declare default element namespace
| "http://www.ibm.com/digital_media_library";
| /BOOK/TITLE[. contains("DB2")]'' ') = 1;
```

| *Tableau 16. Résultat*

ID
3

| **Exemple 3:**

| Recherchez tous les documents contenant un élément racine BOOK dont l'attribut publication_date a une valeur postérieure à "2010-01-01" et un élément enfant TITLE contenant le terme DB2. Limitez la recherche aux documents dont les balises BOOK et TITLE n'existent dans aucun espace de nom.

```
| SELECT ID
| FROM XMLTEST.XML_DOCUMENTS
| WHERE CONTAINS(XML_DATA, '@xmlns:''declare default element namespace "";
| /BOOK[@publication_date > xs:date("2010-01-01")]/TITLE[. contains("DB2")]'' ') = 1;
```


Tableau 17. Résultat

ID
2

Exemple 4 :

Recherchez tous les documents contenant un élément racine BOOK (dans aucun espace de nom) possédant un descendant direct CHAPTER (également dans aucun espace de nom) contenant des informations sur l'insertion de données dans une table XML.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlns:''declare default element namespace "";
/BOOK/CHAPTER[. contains("inserting XML data into a table")]'' ') = 1;
```

Remarque :

- Le texte contenu dans CHAPTER inclut le texte contenu dans les éléments ABSTRACT et TITLE qui sont des descendants de CHAPTER.
- La chaîne de recherche ne distingue pas les majuscules des minuscules et les variantes linguistiques des mots recherchés sont prises en compte.

Tableau 18. Résultat

ID
2

Exemple 5 :

Recherchez tous les documents contenant un élément racine BOOK (dans l'espace de noms "http://www.ibm.com/digital_media_library") et possédant un descendant direct CHAPTER (aussi dans l'espace de noms "http://www.ibm.com/digital_media_library"). CHAPTER doit posséder un descendant direct NUMBER (dans l'espace de noms "http://www.ibm.com/digital_media_library") dont la valeur est 1 et contenir également des informations sur la recherche dans un espace de noms XML.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlns:''declare namespace ns1 = "http://www.ibm.com/digital_media_library";
/ns1:BOOK/ns1:CHAPTER[. contains("search XML using a namespace") and NUMBER = 1]'' ') = 1;
```

Le document n°3 est le seul qui contient des balises dans l'espace de noms demandé mais il contient des occurrences des mots clé uniquement dans un chapitre dont la valeur est 2 (et pas 1).

Aucune ligne ne correspond à la requête.

Tableau 19. Résultat

ID

Exemple 6 :

Recherchez tous les documents contenant un élément racine BOOK (dans l'espace de noms "http://www.ibm.com/digital_media_library") et possédant un descendant direct CHAPTER (dans l'espace de noms "http://www.ibm.com/digital_media_library"). CHAPTER doit posséder un descendant direct NUMBER (dans l'espace de noms "http://www.ibm.com/digital_media_library") ayant la valeur 1. BOOK

| doit posséder un descendant CHAPTER (pas nécessairement avec un descendant NUMBER) qui contient des informations sur la recherche dans un espace de noms XML.

```
| SELECT ID
| FROM XMLTEST.XML_DOCUMENTS
| WHERE CONTAINS(XML_DATA, '@xmlns:'||declare namespace ns1 = "http://www.ibm.com/digital_media_library";
| /ns1:BOOK[ns1:CHAPTER contains("search XML using a namespace")]/ns1:CHAPTER[ns1:NUMBER = 1]'^') = 1;
```

| Le document n°3 contient un élément CHAPTER qui répond aux critères de la requête CONTAINS et un élément CHAPTER possédant un descendant NUMBER ayant la valeur 1. Le document n°3 répond bien à cette requête.

| *Tableau 20. Résultat*

ID
3

| **Référence associée**

- | «SYSPROC.SYSTS_CREATE», à la page 19
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte.
- | Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.
- | «SYSPROC.SYSTS_UPDATE», à la page 35
- | Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.
- | «CONTAINS», à la page 38
- | Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

| **Exemple de recherche XML**

| Cet exemple comprend une table de documents XML, un index de recherche de texte basé sur une colonne de données XML de la table et six requêtes de recherche de texte SQL utilisant la fonction CONTAINS.

| **Remarque :** En utilisant les exemples de codes, vous acceptez les termes des «Licence du code et informations de limitation de responsabilité», à la page 111.

- | • Créez une table nommée XML_DOCUMENTS dans le schéma XMLTEST afin de stocker les documents XML :
| CREATE TABLE XMLTEST.XML_DOCUMENTS (ID INT, XML_DATA XML, PRIMARY KEY (ID));
- | • Créez un index de recherche de texte appelé XML_INDEX sur la colonne XML :
| call SYSPROC.SYSTS_CREATE('XMLTEST', 'XML_INDEX', 'XMLTEST.XML_DOCUMENTS(XML_DATA)', '');
- | • Insérez des documents XML :
| INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
| VALUES(1,
| '<BOOK publication_date="2009-01-01">' ||
| ' <TITLE> OmniFind Text Search Server for DB2 </TITLE>' ||
| ' <ID_NUMBER> 1 ></ID_NUMBER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 1 </NUMBER>' ||
| ' <TITLE> Introduction </TITLE>' ||
| ' <ABSTRACT> Ce chapitre présente les fonctionnalités d'OmniFind
| for DB2 for IBM i </ABSTRACT>' ||
| ' </CHAPTER>' ||
| ' <CHAPTER>' ||
| ' <NUMBER> 2 </NUMBER>' ||

```

|      ' <TITLE> Création d'un index de recherche de texte </TITLE>' ||
|      ' <ABSTRACT> Ce chapitre décrit comment créer un index de recherche de texte </ABSTRACT>' ||
|      ' </CHAPTER>' ||
|      '</BOOK>');
|
| INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
| VALUES(2,
|      '<BOOK publication_date="2010-02-01">' ||
|      ' <TITLE> Utilisation du type de données XML pour DB2 for IBM i </TITLE>' ||
|      ' <ID_NUMBER> 2 </ID_NUMBER>' ||
|      ' <CHAPTER>' ||
|      ' <NUMBER> 1 </NUMBER>' ||
|      ' <TITLE> Introduction </TITLE>' ||
|      ' <ABSTRACT> Ce chapitre présente le type de données XML DB2 </ABSTRACT>' ||
|      ' </CHAPTER>' ||
|      ' <CHAPTER>' ||
|      ' <NUMBER> 2 </NUMBER>' ||
|      ' <TITLE> Insertion de données XML dans une table DB2 </TITLE>' ||
|      ' <ABSTRACT> Ce chapitre décrit comment insérer des données XML dans une table DB2
| </ABSTRACT>' ||
|      ' </CHAPTER>' ||
|      ' <CHAPTER>' ||
|      ' <NUMBER> 3 </NUMBER>' ||
|      ' <TITLE> Recherche de données XML </TITLE>' ||
|      ' <ABSTRACT> Ce chapitre décrit comment interroger les données des colonnes XML
|      avec les fonctions CONTAINS et SCORE </ABSTRACT>' ||
|      ' </CHAPTER>' ||
|      '</BOOK>');
|
| INSERT INTO XMLTEST.XML_DOCUMENTS (ID, XML_DATA)
| VALUES(3,
|      '<BOOK xmlns="http://www.ibm.com/digital_media_library"' ||
|      ' publication_date="2010-02-01">' ||
|      ' <TITLE> Utilisation d'espaces de noms avec OmniFind Text Search Server for DB2 for IBM i
| </TITLE>' ||
|      ' <ID_NUMBER> 2 </ID_NUMBER>' ||
|      ' <CHAPTER>' ||
|      ' <NUMBER> 1 </NUMBER>' ||
|      ' <TITLE> Introduction </TITLE>' ||
|      ' <ABSTRACT> Ce chapitre décrit les espaces de noms XML </ABSTRACT>' ||
|      ' </CHAPTER>' ||
|      ' <CHAPTER>' ||
|      ' <NUMBER> 2 </NUMBER>' ||
|      ' <TITLE> Utilisation des espaces de noms par défaut </TITLE>' ||
|      ' <ABSTRACT> Ce chapitre décrit comment utiliser un espace de noms dans une recherche XML
| </ABSTRACT>' ||
|      ' </CHAPTER>' ||
|      '</BOOK>');
|
| • Mettez à jour l'index :
| CALL SYSPROC.SYSTS_UPDATE('XMLTEST', 'XML_INDEX', '');

```

Exemples de requête

Recherche 1 :

Recherchez tous les documents contenant un élément racine BOOK avec un descendant direct TITLE qui contient le terme DB2.

```

| SELECT ID
| FROM XMLTEST.XML_DOCUMENTS
| WHERE CONTAINS(XML_DATA, '@xmlxp:''/BOOK/TITLE[. contains("DB2")]'' ') = 1;

```

Dans la mesure où aucun prologue d'espace de noms n'est spécifié dans le terme de recherche, aucun espace de noms n'est pris en compte dans la recherche.

Tableau 21. Résultat

ID
1
2
3

Recherche 2 :

Recherchez tous les documents contenant un élément racine BOOK avec un descendant direct TITLE qui contient le terme DB2. Utilisez un espace de noms d'élément par défaut pour indiquer que BOOK et TITLE doivent figurer dans l'espace de nom "http://www.ibm.com/digital_media_library".

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlns:''declare default element namespace
"http://www.ibm.com/digital_media_library";
/BOOK/TITLE[. contains("DB2")]'' ') = 1;
```

Tableau 22. Résultat

ID
3

Recherche 3 :

Recherchez tous les documents contenant un élément racine BOOK dont l'attribut publication_date a une valeur postérieure à "2010-01-01" et un élément enfant TITLE contenant le terme DB2. Limitez la recherche aux documents dont les balises BOOK et TITLE n'existent dans aucun espace de nom.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlns:''declare default element namespace "";
/BOOK[@publication_date > xs:date("2010-01-01")]/TITLE[. contains("DB2")]'' ') = 1;
```

Tableau 23. Résultat

ID
2

Recherche 4 :

Recherchez tous les documents contenant un élément racine BOOK (dans aucun espace de nom) possédant un descendant direct CHAPTER (également dans aucun espace de nom) contenant des informations sur l'insertion de données dans une table XML.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlns:''declare default element namespace "";
/BOOK/CHAPTER[. contains("inserting XML data into a table")]'' ') = 1;
```

Remarque :

- Le texte contenu dans CHAPTER inclut le texte contenu dans les éléments ABSTRACT et TITLE qui sont des descendants de CHAPTER.
- La chaîne de recherche ne distingue pas les majuscules des minuscules et les variantes linguistiques des mots recherchés sont prises en compte.

Tableau 24. Résultat

ID
2

Recherche 5 :

Recherchez tous les documents contenant un élément racine BOOK (dans l'espace de noms "http://www.ibm.com/digital_media_library") et possédant un descendant direct CHAPTER (aussi dans l'espace de noms "http://www.ibm.com/digital_media_library"). CHAPTER doit posséder un descendant direct NUMBER (dans l'espace de noms "http://www.ibm.com/digital_media_library") dont la valeur est 1 et contenir également des informations sur la recherche dans un espace de noms XML.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlns:''declare namespace ns1 = "http://www.ibm.com/digital_media_library";
/ns1:BOOK/ns1:CHAPTER[. contains("search XML using a namespace") and NUMBER = 1]'' ') = 1;
```

Le document n°3 est le seul qui contient des balises dans l'espace de noms demandé mais il contient des occurrences des mots clé uniquement dans un chapitre dont la valeur est 2 (et pas 1).

Aucune ligne ne correspond à la requête.

Tableau 25. Résultat

ID

Recherche 6 :

Recherchez tous les documents contenant un élément racine BOOK (dans l'espace de noms "http://www.ibm.com/digital_media_library") et possédant un descendant direct CHAPTER (dans l'espace de noms "http://www.ibm.com/digital_media_library"). CHAPTER doit posséder un descendant direct NUMBER (dans l'espace de noms "http://www.ibm.com/digital_media_library") ayant la valeur 1. BOOK doit posséder un descendant CHAPTER (pas nécessairement avec un descendant NUMBER) qui contient des informations sur la recherche dans un espace de noms XML.

```
SELECT ID
FROM XMLTEST.XML_DOCUMENTS
WHERE CONTAINS(XML_DATA, '@xmlns:''declare namespace ns1 = "http://www.ibm.com/digital_media_library";
/ns1:BOOK[ns1:CHAPTER contains("search XML using a namespace")]/ns1:CHAPTER[ns1:NUMBER = 1]'' ') = 1;
```

Le document n°3 contient un élément CHAPTER qui répond aux critères de la requête CONTAINS et un élément CHAPTER possédant un descendant NUMBER ayant la valeur 1. Le document n°3 répond bien à cette requête.

Tableau 26. Résultat

ID
3

Règles syntaxiques des requêtes de recherche XML

Les règles syntaxiques des requêtes de recherche XML reposent sur un sous-ensemble du langage XPath, qui applique les normes de syntaxe Extended Backus-Naur Form (EBNF). Les requêtes qui ne respectent pas les règles syntaxiques prises en charge sont refusées par l'analyseur syntaxique.

Les règles syntaxiques EBNF ont été simplifiées comme suit :

- Interdiction des noms de chemin absolus dans les expressions de prédicat

- Reconnaissance limitée à un seul axe (balise) et uniquement en aval
- Application de limites sémantiques supplémentaires pour l'utilisation des caractères génériques (voir la section "Prise en charge des caractères génériques dans les expressions de chemin" dans «Recherche XML», à la page 48).
- Obligation de spécifier la déclaration d'espace de noms dans la chaîne de recherche avant toute utilisation, implicite ou explicite, de l'espace de nom. Si la déclaration d'espace de noms est omise, les espaces de noms ne sont pas pris en compte dans la recherche.
- Obligation d'inclure un nom d'attribut ou de balise dans les expressions de chemin relatif. Il est interdit d'insérer dans les expressions le caractère '/' pour sélectionner le noeud racine et le caractère '/' pour sélectionner tous les noeuds.

Le tableau suivant illustre les règles syntaxiques prises en charge par la notation EBNF.

Tableau 27. Règles syntaxiques prises en charge par la notation EBNF

Symbole	Production
XMLQuery ::=	QueryPrefix NamespaceDeclaration QueryString QueryPrefix QueryString
QueryPrefix ::=	@xmlxp:
QueryString ::=	"" PathExpr ""
PathExpr ::=	RelativePathExpr "/" RelativePathExpr? "/" RelativePathExpr
RelativePathExpr ::=	StepExpr (("/" "/") StepExpr)*
StepExpr ::=	("." AbbrevForwardStep) Predicate?
AbbrevForwardStep ::=	"@"? (QName "*")
Predicate ::=	"[" PredicateExpr "]"
PredicateExpr ::=	Expr PredicateExpr ("and" "or") "(" PredicateExpr ")"
Expr ::=	ComparisonExpr ContainmentExpr
ComparisonExpr ::=	PathExpr ComparisonOp Literal
ComparisonOp ::=	"=" "<" ">" "!=" "<=" ">="
Literal ::=	StringLiteral NumericLiteral DateLiteral
ContainmentExpr ::=	PathExpr "contains" "(" StringLiteral ")" PathExpr "excludes" "(" StringLiteral ")"
StringLiteral ::=	"\" [^]* \"" "" [^]* ""
DateLiteral ::=	"xs:date(\"" xmlDate ")" "xs:dateTime(\"" xmlDateTime ")"
xmlDate ::=	yyyy"-mm"-dd
xmlDateTime ::=	yyyy"-mm"-dd [T] hh":mm":ss".uuuuuu
NamespaceDeclaration ::=	defaultNamespace (NamespacePrefixDeclaration)*
defaultNamespace ::=	"declare default element namespace " StringLiteral ";"
NamespacePrefixDeclaration ::=	"declare namespace" NamespacePrefix "=" StringLiteral ";"
NamespacePrefix ::=	[^:]+

Administration d'un serveur de recherche de texte OmniFind(r) pour DB2(r) for i

Vous pouvez administrer OmniFind Text Search Server for DB2 for i à l'aide des techniques et des outils suivants.

Démarrage de OmniFind Text Search Server for DB2 for i

Vous pouvez démarrer OmniFind Text Search Server for DB2 for i en appelant la procédure SYSPROC.SYSTS_START.

OMNIFIND démarre automatiquement le serveur de recherche de texte quand l'option SERVERSTATUS dans QSYS2.SYSTEXTSERVERS a la valeur 0. Cette option permet de démarrer automatiquement le serveur de recherche de texte au démarrage du système hôte. Toutefois, vous pouvez aussi démarrer le serveur manuellement si nécessaire.

Pour démarrer le serveur, procédez comme suit :

CALL SYSPROC.SYSTS_START(ID de serveur)

Si l'opération aboutit, l'option SERVERSTATUS dans QSYS2.SYSTEXTSERVERS prend la valeur 0 après l'appel de la procédure. Si le serveur est local, les travaux suivants s'exécutent en arrière-plan :

- QJVAEXEC *userx* BCI 0.0 JVM-com.ibm.es
- QJVAEXEC *userx* BCI 0.0 PGM-OutsideInP
- QJVAEXEC *userx* BCI 0.0 PGM-OutsideInP
- QJVAEXEC *userx* BCI 0.0 PGM-OutsideInP
- QJVAEXEC *userx* BCI 0.0 PGM-OutsideInP

où *userx* est l'ID utilisateur de l'administrateur ayant appelé la procédure stockée.

Plusieurs minutes peuvent s'écouler avant que les travaux ne s'activent et que le serveur de texte ne devienne utilisable.

Référence associée

«SYSPROC.SYSTS_START», à la page 16

Vous pouvez activer les fonctions de recherche de texte de DB2 en appelant la procédure stockée SYSPROC.SYSTS_START.

«Table d'administration QSYS2.SYSTEXTSERVERS», à la page 106

La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.

Arrêt de OmniFind Text Search Server for DB2 for i

Vous pouvez arrêter OmniFind Text Search Server for DB2 for i manuellement en exécutant le script d'arrêt fourni.

Si vous avez installé OmniFind Text Search Server for DB2 for i en tant que service, le serveur de recherche de texte s'arrête automatiquement chaque fois que le système hôte s'arrête. Toutefois, vous pouvez arrêter le serveur manuellement même si vous avez installé OmniFind Text Search Server for DB2 for i en tant que service.

Pour arrêter OmniFind Text Search Server for DB2 for i, procédez comme suit :

1. Indiquez dans le catalogue SYSTEXTSERVER que le serveur est arrêté en appelant la procédure SYSPROC.SYSTS_STOP.
 - Pour arrêter tous les serveurs : CALL SYSPROC.SYSTS_STOP().

- Pour arrêter un serveur particulier :

- a. Interrogez le catalogue des serveurs pour obtenir l'ID de serveur (*serverid*) du serveur que vous voulez arrêter :

```
SELECT SERVERID,SERVERPORT,SERVERSTATUS,SERVERPATH
FROM QSYS2.SYSTEXTSERVERS
```

Remarque : SERVERPATH identifie le serveur. SERVERSTATUS indique si le serveur est actif (0) ou inactif (1).

- b. Appelez la procédure SYSPROC.SYSTS_STOP, en indiquant le *serverid* numérique ou le nom d'alias du serveur que vous voulez arrêter :

```
CALL SYSPROC.SYSTS_STOP(serverid).
```

2. (facultatif) Arrêtez le serveur désiré en appelant le script d'arrêt. L'arrêt du serveur a pour effet d'arrêter tous les travaux du serveur de recherche de texte sur le système hôte. Arrêtez le serveur dans l'environnement Qshell.

Pour arrêter le serveur local, entrez la commande suivante sur la ligne de commande :

```
QSH CMD('cd /QOpenSys/QIBM/ProdData/TextSearch/server1/bin; shutdown.sh')
```

Si le serveur que vous voulez arrêter n'est pas le serveur local par défaut créé par le processus d'installation, vous devez obtenir la valeur de la variable SERVERPATH à partir du catalogue QSYS2.SYSTEXTSERVERS. Utilisez le chemin du serveur (SERVERPATH) au lieu de /QOpenSys/QIBM/ProdData/TextSearch/server1/bin.

Si vous arrêtez le serveur avec le script d'arrêt, la valeur de SERVERSTATUS ne devient pas 1 (statut inactif). Quand les procédures stockées SYSTS_CREATE, SYSTS_UPDATE et SYSTS_DROP sont appelées par la suite, ou qu'une fonction définie par l'utilisateur CONTAINS ou SCORE est appelée dans une requête SQL, le serveur démarre automatiquement.

Référence associée

«SYSPROC.SYSTS_STOP», à la page 17

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_STOP pour arrêter les fonctions de recherche de texte de DB2. Cette procédure stockée affecte la valeur 1 (arrêté) au paramètre SERVERSTATUS dans le catalogue QSYS2.SYSTEXTSERVERS.

«Table d'administration QSYS2.SYSTEXTSERVERS», à la page 106

La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.

Sauvegarde et restauration des index de recherche de texte

Vous pouvez sauvegarder et restaurer vos index de recherche de texte avec ou sans données.

Sauvegarde et restauration d'un index de recherche de texte sans données

Vous pouvez sauvegarder et restaurer la structure d'un index de recherche de texte sans ses données.

Vous pouvez exécuter la procédure de sauvegarde et de restauration à l'aide des commandes de langage de contrôle (CL) SAVOBJ et SAVLIB.

Quand vous créez un index de recherche de texte avec la procédure SYSTS_CREATE, une vue DB2 est créée à l'aide du schéma d'index. La vue porte le même nom que l'index. La vue permet de sauvegarder et de restaurer la structure de l'index.

L'utilisateur peut sauvegarder la vue à l'aide des méthodes employées pour sauvegarder les vues et les tables de base de données. (Voir les commandes CL SAVOBJ et SAVLIB). Le processus de sauvegarde de la vue enregistre automatiquement les informations supplémentaires requises pour recréer l'index pendant la restauration.

| Vous pouvez ensuite restaurer la vue à l'aide des commandes CL RSTOBJ ou RSTLIB. DB2 for i reconnaît
| que la vue représente un index de recherche de texte et recrée l'index. Une fois la structure de l'index
| recrée, une mise à jour est soumise à un travail d'arrière-plan pour remplir à nouveau l'index avec des
| données.

| Le processus de restauration appelle les observations suivantes :

- | 1. Si le serveur de recherche de texte ne peut pas démarrer, ou qu'un produit obligatoire n'est pas
| installé sur le système, la restauration échoue. Voir la section Configuration logicielle requise pour
| consulter la liste des produits obligatoires.
- | 2. Si l'index de recherche de texte existe déjà sur le système, les actions suivantes sont exécutées :
 - | a. Si les données de l'index existant sont identiques à celles de l'index en cours de restauration, la
| restauration aboutit. L'index n'est toutefois pas régénéré.
 - | b. Si les données de l'index existant sont différentes de celles de l'index en cours de restauration et
| ne peuvent pas être modifiées sans recréer l'index, la restauration échoue.
 - | c. Si les données de l'index existant sont différentes de celles de l'index en cours de restauration mais
| peuvent pas être modifiées avec la procédure SYSTS_ALTER, l'index existant est modifié pour
| refléter l'index sauvegardé. L'index n'est toutefois pas régénéré.
- | 3. L'index est restauré de manière à utiliser le serveur de recherche de texte utilisé au moment de la
| sauvegarde. Si le serveur utilisé au moment de la sauvegarde n'est pas défini, un serveur disponible
| est sélectionné. Si le serveur utilisé pour la sauvegarde est défini mais indisponible, la restauration
| échoue.
- | 4. S'il est impossible de recréer l'index de recherche de texte pour une autre raison quelconque, par
| exemple une colonne incompatible dans la table de base, la restauration échoue.
- | 5. Le nom de la table de transfert dans QSYS2, les noms des déclencheurs ajoutés dans la table de base
| et le nom de la collection peuvent changer sur le serveur de recherche de texte. Les droits spéciaux
| attachés à ces objets ne sont pas conservés.
- | 6. Les synonymes ajoutés au dictionnaire de synonymes de l'index de recherche de texte ne sont pas
| conservés.
- | 7. Si l'index existe dans les catalogues système au moment de la restauration, et que la vue n'existe pas
| actuellement sur le système, seule la vue est restaurée. La table de transfert, la collection du serveur
| de recherche de texte et les déclencheurs qui étaient définis sur la table de base ne sont pas recréés.
| Dans ce cas, le programme suppose que l'index de recherche de texte fait partie d'une restauration
| plus importante dans laquelle les éléments individuels de l'index ont été sauvegardés explicitement
| par l'utilisateur et sont à présent en cours de restauration (une restauration globale du système par
| exemple).
| Tous les éléments obligatoires de l'index doivent être restaurés pour que l'index fonctionne. Il
| incombe à l'utilisateur de vérifier que tous les éléments de l'index sont synchronisés.

| Les index de recherche de texte sont pris en charge par la commande Restore Deferred Objects
| (RSTDFROBJ). Il est recommandé d'utiliser le paramètre DFRID dans les commandes CL RSTOBJ et
| RSTLIB. Ce paramètre permet de restaurer les index de recherche de texte avec la commande
| RSTDFROBJ après avoir corrigé les erreurs courantes qui peuvent empêcher de créer l'index.

| Exemples de situations pouvant empêcher la création de l'index :

- | • Un produit obligatoire n'est pas installé.
- | • Un serveur de recherche de texte n'est pas défini ou est indisponible.
- | • La table de base n'existe plus.

| **Référence associée**

| «SYSPROC.SYSTS_CREATE», à la page 19

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte. Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

| «SYSPROC.SYSTS_ALTER», à la page 28

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_ALTER pour modifier les attributs d'un index créé par la procédure SYSPROC.SYSTS_CREATE. Seuls les attributs spécifiés explicitement dans cette procédure sont modifiés. Les autres attributs de l'index demeurent inchangés.

| **Information associée**

| SAVOBJ (Sauvegarder un objet)

| SAVLIB (Sauvegarder une bibliothèque)

| RSTOBJ (Restaurer objet)

| RSTLIB (Restaurer bibliothèque)

| RSTDFROBJ (Restaurer objets différés)

| **Sauvegarde et restauration d'un index de recherche de texte avec des données**

| La sauvegarde et la restauration d'un index de recherche de texte comprenant des données est une opération plus complexe qu'en l'absence de données.

| Vous devez sauvegarder les objets suivants :

- | • L'index de recherche de texte (stocké dans le système de fichiers intégré)
- | • La table de transfert utilisée comme fichier journal pour suivre les modifications des enregistrements dans la table de base (sur laquelle l'index est construit). La table de transfert se trouve dans la bibliothèque QSYS2. Son nom commence par QDBTS, par exemple QDBTS00001.
- | • La vue, c'est-à-dire l'objet de base de données qui représente l'index de texte. La vue porte le même nom que l'index de texte.
- | • La table de base sur laquelle l'index est construit.
- | • Les catalogues SQL qui contiennent les informations requises pour suivre l'index.

| Pour sauvegarder les index de recherche de texte, procédez comme suit :

- | 1. Recommandé : Actualisez les index en commençant par les opérations de mise à jour (SYSTS_UPDATE) applicables aux index de recherche de texte.
- | 2. Enregistrez la table de base et la vue à l'aide des techniques de sauvegarde standard comme la commande SAVOBJ.
- | 3. Enregistrez les tables de transfert contenues dans QSYS2 à l'aide des techniques de sauvegarde standard. Par exemple, entrez SAVOBJ LIB(QSYS2) OBJ(QDBTS*).
- | 4. Enregistrez les catalogues d'index de recherche de texte contenus dans QSYS2 :
| Les noms de catalogue commencent tous par SYSTXT, par exemple SYSTXTSRVR. Comme pour les autres catalogues SQL de QSYS2, vous devez vérifier qu'une copie de sauvegarde a été enregistrée et est disponible.
| Vous pouvez effectuer cette copie de sauvegarde de deux manières :
 - | a. Vous pouvez enregistrer toute la bibliothèque avec la commande SAVLIB en spécifiant le paramètre LIB avec la valeur *ALLUSR ou *IBM.
 - | b. Vous pouvez enregistrer des catalogues de recherche de texte spécifiques avec la commande SAVOBJ en spécifiant LIB(QSYS2) et OBJ((SYSTXT*)).
- | 5. Enregistrez les données des index de recherche de texte contenus dans le système de fichiers intégré. Ces données comprennent tout le contenu du répertoire *config* situé dans le chemin du serveur de texte.

| Pour déterminer le chemin du serveur de texte, interrogez la colonne SERVERPATH du catalogue SYSTXTSRVR du serveur concerné. Le répertoire bin est ajouté au chemin du serveur et il suffit de le remplacer par le répertoire config.

| Une technique de sauvegarde commune consiste à utiliser la commande SAV via un travail de compression de sauvegarde quelconque.

| **Remarque :** Ces données de sauvegarde ne s'appliquent qu'aux serveurs de texte exécutés sur IBM i.

| Exemple:

| Imaginez que vous voulez sauvegarder tous les index de texte associés au serveur de texte par défaut créé par OMNIFIND. Il existe une table nommée QGPL/MYDOCS sur laquelle repose l'index de texte QGPL/MYDOCIX. Dans cet exemple, les données sont sauvegardées dans des fichiers.

| Procédez comme suit :

| 1. Enregistrez tous les tables de transfert et les catalogues OMNIFIND depuis QSYS2 :

```
| SAVOBJ OBJ(QDBTS* SYSTXT*) LIB(QSYS2) DEV(*SAVF) SAVF(QGPL/SAVFQSYS2)
```

| 2. Enregistrez la table de base et la vue :

```
| SAVOBJ OBJ(MYDOCS MYDOCIX) LIB(QGPL) DEV(*SAVF) SAVF(QGPL/SAVFMYFILE)
```

| 3. Entrez une instruction SQL pour obtenir le nom de chemin du serveur de texte. Dans cet exemple, serverid = 2 :

```
| SELECT SERVERPATH FROM systxtsrvr WHERE serverid=2
```

| La valeur renvoyée pour le paramètre SERVERPATH est /QOpenSys/QIBM/ProdData/TextSearch/server1/bin/.

| **Remarque :** Vérifiez que la requête porte sur le serveur approprié.

| 4. Remplacez bin/ par config et enregistrez les index de texte :

```
| SAV DEV('/QSYS.LIB/QGPL.LIB/SAVIFS.FILE') OBJ('/QOpenSys/QIBM/ProdData/TextSearch/server1/config')
```

| Les index de texte sont à présent enregistrés dans les fichiers de sauvegarde QGPL/SAVFMYFILE, QGPL/SAVFQSYS2 et QGPL/SAVIFS.

| La restauration des index de texte doit être exécutée dans le même ordre que la sauvegarde. Les catalogues de QSYS2 doivent OBLIGATOIREMENT être restaurés en premier.

| **Référence associée**

| «SYSPROC.SYSTS_UPDATE», à la page 35

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

| **Information associée**

| SAVOBJ (Sauvegarder un objet)

| SAVLIB (Sauvegarder une bibliothèque)

| **Identification des problèmes**

| Vous pouvez utiliser les messages de trace et les messages du système journalisés pour déterminer la source des problèmes qui se produisent.

| Les journaux du serveur OmniFind Text Search Server for DB2 for i résident dans le répertoire <INSTALL_HOME>/log. Le serveur par défaut créé à l'installation réside dans le répertoire /QOpenSys/QIBM/ProdData/TextSearch/server1/log.

| Par défaut, le journal de trace est désactivé et le journal système est réglé sur le niveau informationnel.
| Vous pouvez utiliser l'outil de configuration pour changer les niveaux du journal de trace et du journal système.

| La rotation des journaux du serveur s'effectue sur la base de leur taille. Les cinq dernières copies des journaux de serveur inférieurs à 8 Mo sont stockés. Vous pouvez visualiser et enregistrer les journaux de serveur à l'aide du script fourni.

| Sur IBM i ou un serveur Linux, ce script est `logformatter.sh`. Sur un serveur Windows, ce script est `logformatter.bat`.

| Options

| Le script accepte les options suivantes :

- | `-f logfile`
| Indique le fichier journal de serveur que vous voulez formater.
- | `-l locale`
| Indique l'environnement local à utiliser pour écrire les messages reformatés. Par exemple, indiquez `en_US` pour l'anglais américain, ou `ja_JP` pour le japonais. Cette valeur est facultative. La valeur par défaut est `en_US`.
- | `-o outputfile`
| Indique le fichier de sortie dans lequel les messages de journal reformatés sont écrits avec le codage UTF-8. Cette valeur est facultative. Si vous n'indiquez pas de valeur, un fichier de sortie standard est utilisé.
- | `-?` Affiche le message d'aide. Cette valeur est facultative.
- | `-v` Indique le mode à utiliser pour le débogage des messages. Cette valeur est facultative.

| Affichage et enregistrement des journaux serveur

| Vous pouvez utiliser les scripts fournis pour afficher et enregistrer les journaux de serveur. Ces fichiers journaux facilitent l'identification de la cause des problèmes.

| Pour visualiser et enregistrer un journal serveur, procédez comme suit :

- | 1. Pour visualiser le journal serveur, exécutez l'une des commandes suivantes :

Option	Description
Sur IBM i (dans l'environnement QSH)	<code>bin/logformatter.sh -f log/System.0.log</code>
Sur un serveur Linux	<code>bin/logformatter.sh -f log/System.0.log</code>
Sur un serveur Windows	<code>bin/logformatter.bat -f log/System.0.log</code>

- | 2. Pour enregistrer le journal serveur dans un fichier que vous pourrez lire dans un programme d'édition, exécutez l'une des commandes suivantes :

Option	Description
Sur IBM i (dans l'environnement QSH)	<code>bin/logformatter.sh -f log/System.0.log -o <nom de fichier de sortie></code>
Sur un serveur Linux	<code>bin/logformatter.sh -f log/System.0.log -o <nom de fichier de sortie></code>

Option	Description
Sur un serveur Windows	bin/logformatter.bat -f log/System.0.log -o <nom de fichier de sortie>

Outils d'administration

OMNIFIND fournit des outils qui permettent d'exécuter des tâches communes comme, par exemple, configurer et administrer un serveur de recherche de texte supplémentaire ou ajouter un dictionnaire de synonymes à une collection.

Ces outils sont davantage des scripts shell que des commandes CL. Ils peuvent s'appeler depuis l'environnement de script que vous démarrez à l'aide des commandes CL Start QSH (STRQSH) ou QSH (QSH).

Notez que ces outils n'authentifient pas les ID utilisateur. Toutefois, ils ne sont utilisables que par un utilisateur doté de droits d'accès valides au serveur de recherche de texte.

Information associée

STRQSH (Démarrer QSH)

QSH (Démarrer QSH)

Outil de configuration

L'outil de configuration vous permet de personnaliser les paramètres de configuration après l'installation de OmniFind Text Search Server for DB2 for i.

Pour personnaliser la plupart des paramètres de configuration, vous devez arrêter le serveur de recherche de texte avant d'exécuter l'outil de configuration.

Toutefois, quand le serveur est en cours d'exécution, vous pouvez afficher les options suivantes :

- jeton d'authentification actif
- port du serveur
- propriétés actives du système

Outil configServerAndDB2

L'outil configServerAndDB2 (configServerAndDB2.sh) réside dans le répertoire du système de fichiers intégré /QOpenSys/QIBM/ProdData/TextSearch. Vous pouvez utiliser cet outil pour créer ou modifier des entrées dans le fichier de catalogue SYSTEXTSERVERS de DB2.

Vous pouvez aussi l'utiliser pour configurer le jeton d'authentification ou le numéro de port du serveur concerné. Cet outil modifie ou définit les valeurs des paramètres SERVERAUTHTOKEN et SERVERPORT dans le fichier de catalogue SYSTEXTSERVERS de DB2.

Si vous désirez créer un autre serveur exécuté localement sur votre système, utilisez plutôt l'outil «Utilitaire ServerInstance», à la page 81.

L'outil configServerAndDB2 (configServerAndDB2.sh) est appelé avec cinq paramètres :

1. Le premier paramètre est au choix **generateToken** ou **configureHTTPListener**.
2. Le second paramètre est **-serverPath**.
3. Le troisième paramètre indique le chemin du noeud racine du système de fichiers intégré dans lequel les informations associées au serveur sont stockées. Exemple : /QOpenSys/QIBM/ProdData/TextSearch/server2.

- 4. Le quatrième et le cinquième paramètres varient selon la valeur du premier paramètre.
 - Si le premier paramètre a la valeur **generateToken**, la valeur du quatrième paramètre est **-seed** et celle du cinquième paramètre est un entier (par exemple 1).
 - Si le premier paramètre a la valeur **configureHTTPListener**, la valeur du quatrième paramètre est **-adminHTTPPort**. Celle du cinquième paramètre est alors un entier qui désigne le port de socket du serveur.

Voici deux exemples :

- STRQSH
`cd /QOpenSys/QIBM/ProdData/TextSearch`
`configServerAndDB2.sh generateToken -serverPath /QOpenSys/QIBM/ProdData/TextSearch/server2 -seed 1`
- STRQSH
`cd /QOpenSys/QIBM/ProdData/TextSearch`
`configServerAndDB2.sh configureHTTPListener -serverPath /QOpenSys/QIBM/ProdData/TextSearch/server2`
`-adminHTTPPort 9997`

Script configTool

Le script configTool.sh est disponible pour chaque serveur local. Il est déconseillé de l'utiliser pour modifier des entrées de serveur. Vous pouvez l'utiliser pour afficher les informations du serveur (par exemple printAll et printToken).

Tableau 28. Commandes à exécuter pour démarrer l'outil de configuration

Sur IBM i :
<code>configTool.sh</code> <code><option de commande obligatoire></code> <code><arguments communs obligatoires></code> <code><arguments communs facultatifs></code> <code><options de commande facultatives></code>
Sur un serveur Linux :
<code>configTool.sh</code> <code><option de commande obligatoire></code> <code><arguments communs obligatoires></code> <code><arguments communs facultatifs></code> <code><options de commande facultatives></code>
Sur un serveur Windows :
<code>configTool.bat</code> <code><option de commande obligatoire></code> <code><arguments communs obligatoires></code> <code><arguments communs facultatifs></code> <code><options de commande facultatives></code>

Options de commande

L'outil de configuration prend en charge les options de commande suivantes :

configureParams

Indique les paramètres système que vous pouvez configurer. Vous pouvez configurer les paramètres suivants :

-configPath

Indique le chemin d'accès absolu du dossier de configuration qui contient le fichier config.xml.

-adminHTTPPort

Indique le numéro du port HTTP d'administration. Si une erreur se produit, le code d'erreur -3 est renvoyé.

| *-logPath*
 | Indique le chemin d'accès absolu du répertoire des journaux.

| *-temDirPath*
 | Indique le chemin d'accès absolu du répertoire temporaire.

| *-numberOfIndexers*
 | Indique le nombre de sous-systèmes simultanés utilisés pour l'indexation de recherche de texte.

| *-numberOfTokenizers*
 | Indique le nombre de sous-systèmes simultanés utilisés pour analyser les entrées et les convertir en jetons.

| *-maxDocumentSize*
 | Indique le nombre maximum de caractères à indexer dans un document. Si une erreur se produit, le code d'erreur -3 est renvoyé.

| *-logLevel*
 | Indique le niveau de journalisation des messages système dans le fichier journal. Le niveau par défaut est `informational`. Les autres options sont `warning` et `severe`.

| *-maxHeapSize*
 | Démarre et termine la taille de pile dans un format accepté par la machine virtuelle Java. Si une erreur se produit, le code d'erreur -5 est renvoyé.

| *printToken*
 | Affiche le jeton d'authentification actif et une clé de chiffrement.

| *printAll*
 | Affiche toutes les valeurs actives pour les options que vous pouvez configurer avec cet outil.

| *printAdminHTTPPort*
 | Affiche la valeur actuelle du port HTTP d'administration.

| *generateToken*
 | Génère le jeton d'authentification.

| **Arguments communs**

| *-configPath*
 | Indique le chemin d'accès absolu du dossier de configuration qui contient le fichier `config.xml`. Cet argument commun est obligatoire.

| *-locale*
 | Indique la valeur du paramètre d'environnement local à utiliser pour rédiger les messages dans le fichier trace. Si vous n'indiquez pas de valeur, la valeur par défaut `en_US` est utilisée.

| **Exemple**

| Sur un serveur Linux, entrez la commande suivante pour afficher le jeton d'authentification actif :

```
| configTool.sh printToken -configPath <path> <arguments communs facultatifs>
```

| **Information associée**

| QSH (Démarrer QSH)

| **SYSPROC.SYSTS_REMOVE**

| Vous pouvez supprimer les index orphelins à l'aide de la procédure stockée SQL `SYSPROC.SYSTS_REMOVE`.

| Autorisation

| Vous pouvez identifier le nom de collection des index potentiellement orphelins à l'aide de la fonction de table définie par l'utilisateur (UDTF) QDBTS_LISTINXSTS.

| Les privilèges associés à l'ID de l'émetteur de l'instruction doivent comprendre au moins un des privilèges suivants :

- | • autorisation *JOBCTL
- | • utilisation de la fonction spéciale de sécurité QIBM_DB_SQLADM

| Syntaxe

| >>-SYSPROC.SYSTS_REMOVE (nom_collection) -><

| Paramètre

| *nom_collection*
| Littéral chaîne qui indique le nom de la collection à supprimer.

| **Remarque :** Cette procédure utilise le script shell `adminTool.sh` pour supprimer le répertoire de la collection. Pour utiliser ce script shell, le serveur doit être en cours d'exécution. Si le serveur n'est pas démarré, cette procédure renvoie un message d'erreur.

| Code SQL pour SYSTS_REMOVE

```
| CREATE PROCEDURE SYSPROC.SYSTS_REMOVE(  
|     IN COLLECTIONNAME VARCHAR(255) CCSID 1208)  
|     EXTERNAL NAME QDBTSLIB.DSN5RMCOLL  
|     DYNAMIC RESULT SETS 0  
|     LANGUAGE C++  
|     PARAMETER STYLE SQL  
|     PROGRAM TYPE MAIN  
|     COMMIT ON RETURN NO  
|     INHERIT SPECIAL REGISTERS;
```

| Exemples

- | • Pour supprimer un index orphelin portant le nom de collection 0_65_2815_2008_06_02_11_58_22_901726 dans le groupe d'ASP *SYSBASE, entrez la commande suivante dans n'importe quelle interface SQL :

| `CALL SYSPROC.SYSTS_REMOVE('0_65_2815_2008_06_02_11_58_22_901726')`

| La procédure stockée SYSTS_REMOVE contrôle si les données d'index sont dans la table du catalogue QSYS2.SYSTEXTINDEXES. Si oui, le message d'erreur DSX_INDEX_EXISTS est renvoyé. Sinon, la procédure analyse le répertoire `config/collections` du serveur 65.

| Si la collection n'existe pas, un message d'erreur DSX_COLLECTION_NOT_FOUND est renvoyé. Si elle existe, la procédure appelle le script `adminTool.sh` pour supprimer la collection.

| Ensuite, la procédure contrôle à nouveau le répertoire pour vérifier que la collection a été supprimée. Si la collection est toujours là, l'utilisateur reçoit le message d'erreur DSX_REMOVE_COLLECTION_FAILED.

| **Remarque :** Quand la collection réside dans un groupe d'ASP indépendants sur le serveur de recherche de texte, l'unité d'exécution qui appelle la procédure stockée SYSTS_REMOVE doit s'exécuter dans l'espace de noms de l'ASP indépendant concerné. Utilisez la commande `Set Auxiliary Storage Pool Group (SETASPGRP)`.

- | • Pour supprimer un index orphelin portant le nom de collection 33_7_26_2008_06_18_21_28_39_407824 dans un ASP indépendant nommé `iaspXXX`, vous pouvez utiliser les commandes suivantes :
| CL :


```
| SETASPGRP(isapXXX)

| SQL :
| CALL SYSPROC.SYSTS_REMOVE(' 33_7_26_2008_06_18_21_28_39_407824')
```

| **Remarque :** Si vous utilisez System i Navigator, cliquez avec le bouton droit sur le nom de la base de données de l'ASP indépendant, puis exécutez les scripts SQL.

| **Référence associée**

| «Table d'administration QSYS2.SYSTEXTINDEXES», à la page 103
| La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de schéma et un nom de collection associée sur le serveur de recherche de texte.
| «Recherche des index orphelins et manquants», à la page 79
| Vous pouvez rechercher les index orphelins et manquants à l'aide d'une fonction de table définie par l'utilisateur (UDTF-User Defined Table Function) SQL nommée QDBTS_LISTINXSTS.

| **SYSPROC.SYSTS_REPRIMEINDEX**

| La procédure stockée SYSPROC.SYSTS_REPRIMEINDEX permet de réorganiser l'index et de démarrer une mise à jour initiale. Vous pouvez utiliser cette procédure stockée quand vous souhaitez restaurer des données de la table de base.

| Si les données de la table de base sont restaurées, le contenu mis à jour de la table de base ne peut pas être indexé pendant l'appel de la procédure stockée SYSTS_UPDATE. Dans ce cas, vous pouvez appeler la procédure stockée SYSPROC.SYSTS_REPRIMEINDEX afin de réorganiser l'index.

| **Remarque :** Si un dictionnaire de synonymes a été créé pour l'index de recherche de texte, le processus supprime ce dictionnaire.

| **Syntaxe**

```
| >>-SYSPROC.SYSTS_REPRIMEINDEX( schéma_index, nom_index, options) -><
```

| Le qualificateur de schéma est SYSPROC.

| **Paramètres**

| *schéma_index*

| Identifie le schéma de l'index de recherche de texte. Si la valeur de ce paramètre est nulle, la valeur du registre spécial CURRENT_SCHEMA utilisé pour l'auteur de l'appel est utilisée.

| Le type de données de ce paramètre est VARCHAR(128).

| *nom_index*

| Identifie le nom de l'index de recherche de texte. Le nom de l'index de recherche de texte associé au schéma d'index identifie de manière unique l'index de recherche en texte intégral dans le sous-système DB2. Vous devez indiquer une valeur non nulle pour ce paramètre.

| Le type de données de ce paramètre est VARCHAR(128).

| *options*

| Chaîne de caractères qui indique les options pouvant être ajoutées par la suite pour cette procédure stockée.

| **Important :** Vous devez indiquer une valeur nulle pour le paramètre *options*. A défaut, des erreurs seront générées. Pour savoir comment spécifier le paramètre *options*, reportez-vous à l'exemple ci-après.

| **SQL pour SYSTS_REPRIMEINDEX**

```
| CREATE PROCEDURE SYSPROC.SYSTS_REPRIMEINDEX(  
|         IN INDEXSCHEMA VARCHAR(128) CCSID 1208,  
|         IN INDEXNAME  VARCHAR(128) CCSID 1208,  
|         IN OPTIONS    VARCHAR(32000) CCSID 1208)  
|     EXTERNAL NAME QDBTSLIB.DSN5RPMIDX  
|     DYNAMIC RESULT SETS 0  
|     LANGUAGE C  
|     PARAMETER STYLE SQL  
|     MODIFIES SQL DATA  
|     PROGRAM TYPE MAIN  
|     COMMIT ON RETURN NO  
|     INHERIT SPECIAL REGISTERS
```

| **Exemple**

- Pour réorganiser un index à partir de n'importe quelle interface SQL, entrez la commande suivante dans l'interface :

```
| CALL SYSPROC.SYSTS_REPRIMEINDEX('indexSchema1','indexName1','')
```

| **Référence associée**

| «SYSPROC.SYSTS_UPDATE», à la page 35

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de recherche de texte avec le contenu actuel de la colonne de texte.

| **SYSPROC.SYSTS_VALIDITYCHECK**

| Vous pouvez utiliser la procédure stockée SQL SYSPROC.SYSTS_VALIDITYCHECK afin de vérifier la validité des éléments d'un index.

| **Syntaxe**

| Cette procédure stockée peut corriger des éléments non valides quand le paramètre *autoFix* est spécifié.

```
| >>-SYSPROC.SYSTS_VALIDITYCHECK( schéma_index, nom_index, autoFix) -><
```

| Le qualificateur de schéma est SYSPROC.

| **Paramètres**

| *schéma_index*

| Identifie le schéma de l'index de recherche de texte. Si la valeur de ce paramètre est nulle, la valeur du registre spécial CURRENT SCHEMA utilisé pour l'auteur de l'appel est utilisée.

| Le type de données de ce paramètre est VARCHAR(128).

| *nom_index*

| Identifie le nom de l'index de recherche de texte. Le nom de l'index de recherche de texte associé au schéma d'index identifie de manière unique l'index de recherche en texte intégral dans le sous-système DB2. Vous devez indiquer une valeur non nulle pour ce paramètre.

| Le type de données de ce paramètre est VARCHAR(128).

| *autoFix*

| Indique si la correction automatique est demandée. La valeur de ce paramètre est obligatoirement 0 ou 1. La signification de ces valeurs est la suivante :

| **0** Seule la validité de l'index est vérifiée.

| **1** La validité est vérifiée et les éléments non valides sont corrigés.

| **Remarque :**

Si d'autres valeurs que 0 ou 1 sont spécifiées, elles sont interprétées comme égales à 0.

Le type de données de ce paramètre est INTEGER.

Restrictions : Si les paramètres schéma_index et nom_index sont spécifiés tous les deux avec la valeur *NONE, la procédure stockée vérifie uniquement la validité des parties communes du produit.

SQL for SYSTS_VALIDITYCHECK

```
CREATE PROCEDURE SYSPROC.SYSTS_VALIDITYCHECK
  (IN INDEXSCHEMA VARCHAR(128) CCSID 1208,
   IN INDEXNAME VARCHAR(128) CCSID 1208,
   IN AUTOFIX INTEGER)
  EXTERNAL NAME QDBTSLIB.DSN5VALCHK
  DYNAMIC RESULT SETS 0
  LANGUAGE C
  PARAMETER STYLE SQL
  MODIFIES SQL DATA
  PROGRAM TYPE MAIN
  COMMIT ON RETURN NO
  INHERIT SPECIAL REGISTERS
```

Exemples

- Pour vérifier la validité d'un index, entrez la commande suivante dans n'importe quelle interface SQL :
`CALL SYSPROC.SYSTS_VALIDITYCHECK('indexSchema1','indexName1',0)`
- Pour vérifier un index et le corriger automatiquement, entrez :
`CALL SYSPROC.SYSTS_VALIDITYCHECK('indexSchema1','indexName1',1)`

Dictionnaires de synonymes

Un dictionnaire de synonymes peut améliorer la qualité des résultats de la recherche.

Vous pouvez ajouter un dictionnaire de synonymes à une collection à tout instant.

Un dictionnaire de synonymes comprend des groupes de synonymes que vous définissez dans un fichier XML. Par exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups version="1.0">
  <synonymgroup>
    <synonym>Paixão</synonym>
    <synonym>amor</synonym>
    <synonym>flor</synonym>
    <synonym>linda</synonym>
  </synonymgroup>
  <synonymgroup>
    <synonym>worldwide patent tracking system</synonym>
    <synonym>wpts</synonym>
  </synonymgroup>
</synonymgroups>
```

Ajout d'un dictionnaire de synonymes dans une collection :

Vous pouvez ajouter un dictionnaire de synonymes à une collection avec l'utilitaire Synonym.

La spécification d'un groupe de synonymes dans un dictionnaire de synonymes améliore la qualité des résultats de la recherche de texte. L'administrateur OMNIFIND a l'autorité et les privilèges requis pour exécuter l'utilitaire Synonym.

Pour ajouter un dictionnaire de synonymes à une collection, procédez comme suit :

1. Créez un fichier XML de synonymes en indiquant le groupe de synonymes, comme dans l'exemple ci-après :

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups version="1.0">
  <synonymgroup>
    <synonym>Paixão</synonym>
    <synonym>amor</synonym>
    <synonym>flor</synonym>
    <synonym>linda</synonym>
  </synonymgroup>
  <synonymgroup>
    <synonym>worldwide patent tracking system</synonym>
    <synonym>wpts</synonym>
  </synonymgroup>
</synonymgroups>
```

2. Copiez le fichier XML de synonymes dans un répertoire quelconque sur le serveur de recherche de texte.

3. Utilisez l'utilitaire Synonym pour ajouter le dictionnaire de synonymes à une collection.

Vous pouvez ajouter un dictionnaire de synonymes dans le mode append ou dans le mode replace. Si vous ajoutez un dictionnaire de synonymes dans le mode append, les nouveaux synonymes sont ajoutés au dictionnaire de synonymes existant. Si vous ajoutez un dictionnaire de synonymes dans le mode replace, les synonymes existants sont remplacés par les nouveaux synonymes que vous avez définis pour l'index de recherche de texte.

Option	Description
Sur IBM i, entrez la commande suivante (dans l'interface QSH) :	<pre>synonymTool.sh importSynonym -synonymFile <chemin absolu du fichier XML de synonymes> -collectionName <nom de la collection> -replace <[true false]> -configPath <chemin absolu du dossier de configuration></pre>
Sur un serveur Linux, entrez la commande suivante :	<pre>synonymTool.sh importSynonym -synonymFile <chemin absolu du fichier XML de synonymes> -collectionName <nom de la collection> -replace <[true false]> -configPath <chemin absolu du dossier de configuration></pre>
Sur un serveur Windows, entrez la commande suivante :	<pre>synonymTool.bat importSynonym -synonymFile <chemin absolu du fichier XML de synonymes> -collectionName <nom de la collection> -replace <[true false]> -configPath <chemin absolu du dossier de configuration></pre>

Si le format du fichier XML est incorrect, ou que le fichier XML est vide, un code d'erreur est renvoyé.

Suppression d'un dictionnaire de synonymes dans une collection :

Vous pouvez utiliser le script fourni pour supprimer un dictionnaire de synonymes dans une collection.

L'administrateur OMNIFIND doit récupérer le nom de la collection dans laquelle vous voulez supprimer un dictionnaire de synonymes.

Exécutez le script pour supprimer le dictionnaire de synonymes dans une collection.

Option	Description
Sur IBM i, entrez la commande suivante (dans l'interface QSH) :	removeSynonym.sh -collectionName <nom de la collection> -configPath <chemin absolu du dossier de configuration>
Sur un serveur Linux, entrez la commande suivante :	removeSynonym.sh -collectionName <nom de la collection> -configPath <chemin absolu du dossier de configuration>
Sur un serveur Windows, entrez la commande suivante :	removeSynonym.bat -collectionName <nom de la collection> -configPath <chemin absolu du dossier de configuration>

Si une base de données contient plusieurs index de recherche de texte, vous devez réaliser cette tâche pour chacune des collections correspondantes.

Recherche des index orphelins et manquants

Vous pouvez rechercher les index orphelins et manquants à l'aide d'une fonction de table définie par l'utilisateur (UDTF-User Defined Table Function) SQL nommée QDBTS_LISTINXSTS.

Un index peut devenir orphelin si une procédure stockée SYSTS_DROP est appelée et que le serveur est arrêté pendant que la procédure s'exécute.

La fonction QDBTS_LISTINXSTS combine l'ensemble des collections et index de catalogue du système de fichiers intégré dans l'espace de noms actif en une seule table. La fonction détermine quel pool de mémoire secondaire (ASP - auxiliary storage pool) indépendant ou *SYSBASE est défini. Elle analyse ensuite le répertoire des collections de chaque serveur dans l'ASP indépendant ou *SYSBASE.

Pour *SYSBASE, chaque répertoire de serveur placé sous /QOpenSys/QIBM/ProdData/TextSearch est vérifié. Pour les pools de mémoire secondaires (ASP) indépendants, chaque répertoire de serveur placé sous /numéro ASP/QOpenSys/QIBM/ProdData/TextSearch est vérifié. Par exemple, si l'ASP indépendant a le numéro 67, chaque répertoire de serveur placé sous /67/QOpenSys/QIBM/ProdData/TextSearch est vérifié.

Les données d'index de catalogue sont obtenues à partir de la table de catalogue QSYS2.SYSTEXTINDEXES. Si vous désirez vérifier les serveurs sur un ASP indépendant, entrez la commande SETASPGRP (Set Auxiliary Storage Pool Group) avant l'appel de cette fonction.

Si vous souhaitez supprimer les index orphelins éventuels du système de fichiers intégré après leur identification, utilisez la procédure stockée SYSPROC.SYSTS_REMOVE ou l'«Administration avancée», à la page 80 (adminTool.sh).

Termes

Index orphelin

Il existe une collection (un index) dans le répertoire du système de fichiers intégré du serveur mais aucun index correspondant n'est enregistré dans le catalogue QSYS2.SYSTEXTINDEXES.

Index manquant

Il existe des groupes d'entrées d'index dans le catalogue QSYS2.SYSTEXTINDEXES mais le répertoire de collection correspondant n'existe pas.

| Syntaxe

| >>-QDBTS_LISTINXSTS(--null--)->><<

| Format de retour

| La fonction QDBTS_LISTINXSTS renvoie des informations sur les index détectés dans une table.
| Reportez-vous à la commande suivante qui permet de créer la fonction UDTF.

| SQL for LISTINXSTS UDTF

```
| CREATE FUNCTION QDBTSLIB.QDBTS_LISTINXSTS()  
|     RETURNS TABLE(COLLECTIONNAME VARCHAR(255),  
|                   INDEXID INTEGER,  
|                   INDEXSCHEMA VARCHAR(128),  
|                   INDEXNAME VARCHAR(128),  
|                   SERVERID INTEGER)  
|     SPECIFIC qdbts_listinxsts  
|     SCRATCHPAD  
|     NO FINAL CALL  
|     LANGUAGE C++  
|     PARAMETER STYLE DB2SQL  
|     EXTERNAL NAME 'QDBTSLIB/QDBTSSP(checkIndex)';
```

| Exemples

| • Détecter tous les index orphelins :

```
| SELECT COLLECTIONNAME, SERVERID  
|       FROM TABLE(QDBTSLIB.QDBTS_LISTINXSTS()) AS T  
|       WHERE T.INDEXSCHEMA IS NULL AND T.INDEXNAME IS NULL
```

| • Détecter tous les index manquants :

```
| SELECT INDEXSCHEMA, INDEXNAME  
|       FROM TABLE(QDBTSLIB.QDBTS_LISTINXSTS()) AS T  
|       WHERE T.COLLECTIONNAME IS NULL
```

| • Détecter les index orphelins sur le serveur serverid = 2 sur l'ASP indépendant iaspXXX :

```
| CONNECT TO iaspXXX  
| SQL:  
| SELECT T.COLLECTIONNAME, S.SERVERPATH  
|       FROM TABLE(QDBTSLIB.QDBTS_LISTINXSTS())  
|           AS T LEFT OUTER JOIN QSYS2.SYSTEXTSERVERS S ON (T.SERVERID = S.SERVERID)  
|       WHERE T.INDEXSCHEMA IS NULL AND T.INDEXNAME IS NULL AND T.SERVERID = 2
```

| Référence associée

| «SYSPROC.SYSTS_DROP», à la page 33
| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_DROP pour supprimer un index de
| recherche de texte créé avec la procédure stockée SYSPROC.SYSTS_CREATE.
| «SYSPROC.SYSTS_REMOVE», à la page 73
| Vous pouvez supprimer les index orphelins à l'aide de la procédure stockée SQL
| SYSPROC.SYSTS_REMOVE.
| «Table d'administration QSYS2.SYSTEXTINDEXES», à la page 103
| La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur
| chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de
| schéma et un nom de collection associée sur le serveur de recherche de texte.

| Administration avancée

| Vous pouvez utiliser l'outil d'administration pour exécuter les tâches d'administration avancée.

| OmniFind Text Search Server for DB2 for i peut être en cours d'exécution quand vous utilisez l'outil
| d'administration.

- Vous pouvez utiliser l'outil d'administration pour exécuter les tâches suivantes :
- Vérification de l'état des collections, par exemple pour identifier le nombre de documents présents
- Suppression des collections orphelines
- Identification de la version du serveur
- Identification des collections présentes sur le serveur de recherche de texte

Commandes

La commande que vous devez exécuter pour démarrer l'outil d'administration varie selon le système d'exploitation du système sur lequel le serveur de recherche de texte est installé. La commande dépend également de la tâche que vous envisagez d'exécuter.

Tableau 29. Commandes à exécuter pour vérifier l'état des collections et pour supprimer les collections orphelines

Sur IBM i (dans l'interface QSH)	Sur un serveur Linux	Sur un serveur Windows
<code>adminTool.sh -[delete status]</code> <code>-collectionName</code> <code><nom de la collection></code> <code>-configPath</code> <code><chemin absolu du dossier de configuration></code>	<code>adminTool.sh -[delete status]</code> <code>-collectionName</code> <code><nom de la collection></code> <code>-configPath</code> <code><chemin absolu du dossier de configuration></code>	<code>adminTool.bat -[delete status]</code> <code>-collectionName</code> <code><nom de la collection></code> <code>-configPath</code> <code><chemin absolu du dossier de configuration></code>

Tableau 30. Commandes à exécuter pour afficher la version du serveur et pour identifier toutes les collections

Sur IBM i (dans l'interface QSH)	Sur un serveur Linux	Sur un serveur Windows
<code>adminTool.sh -[version]</code> <code>-configPath</code> <code><chemin absolu du dossier de configuration></code>	<code>adminTool.sh -[version reportAll]</code> <code>-configPath</code> <code><chemin absolu du dossier de configuration></code>	<code>adminTool.bat -[version reportAll]</code> <code>-configPath</code> <code><chemin absolu du dossier de configuration></code>

Options

status

Vérifie l'état de la collection.

delete

Supprime la collection orpheline.

version

Affiche la version du serveur.

reportAll

Indique toutes les collections présentes sur le serveur de recherche de texte.

Exemple

Pour rechercher la version d'un serveur Linux, entrez la commande suivante :

```
adminTool.sh -version -s <chemin absolu du fichier config.xml du serveur>
```

Si vous utilisez un serveur Windows, un script .bat correspondant est fourni.

Utilitaire ServerInstance

Vous pouvez utiliser l'utilitaire ServerInstance pour créer ou supprimer des serveurs sur *SYSBASE ou sur un pool de mémoire secondaire (ASP) indépendant. Vous pouvez également utiliser l'utilitaire ServerInstance pour lier des fichiers depuis un serveur vers le serveur où est installé OmniFind Text Search Server for DB2 for i.

| Par défaut, OmniFind Text Search Server for DB2 for i est installé dans le répertoire /QOpenSys/QIBM/
| ProdData/TextSearch/server1.

| Vous pouvez utiliser l'utilitaire ServerInstance pour exécuter les tâches suivantes avant de l'utiliser pour
| arrêter le serveur server1 sur *SYSBASE :

- | • Création d'un serveur sur *SYSBASE ou sur des ASP indépendants
- | • Suppression d'un serveur sur *SYSBASE ou sur des ASP indépendants
- | • Liaison de fichiers entre un serveur et server1

| Syntaxe

| ServerInstance.sh -[create|delete|relink]
| -servernum <numéro_serveur>
| (-port <port>)
| (-device <nom_unité>)

| Options de commande

| *create*

| Crée un serveur.

| *delete*

| Supprime un serveur.

| *relink*

| Lie des fichiers entre un serveur et server1.

| **Remarque :** Vous n'avez pas besoin de cette option si vous avez installé le PTF SI31548 sur votre
| système. Le système exécute automatiquement la liaison si ce PTF est installé.

| Paramètres

| *numéro_serveur*

| Indique le numéro du serveur. Par exemple, quand un serveur nommé server3 est créé, le répertoire
| de ce serveur est /QOpenSys/QIBM/ProdData/TextSearch/server3.

| *port*

| Indique le port du serveur. Ce paramètre est nécessaire uniquement quand vous créez un serveur.

| *nom_unité*

| Indique le nom de l'ASP indépendant. Ce paramètre est nécessaire uniquement quand l'opération
| s'exécute sur un ASP indépendant.

| Exemples

- | • Pour créer un serveur sous le numéro de serveur 2 avec le numéro de port 50000 sur *SYSBASE,
| procédez comme suit :

| ServerInstance.sh -create -servernum 2 -port 50000

- | • Pour créer un serveur sous le numéro de serveur 3 avec le numéro de port 50001 sur l'ASP
| indépendant iasp1, procédez comme suit :

| ServerInstance.sh -create -servernum 3 -port 50001 -device iasp1

- | • Pour supprimer le serveur numéro 2 sur *SYSBASE, procédez comme suit :

| ServerInstance.sh -delete -servernum 2

- | • Pour supprimer le serveur numéro 3 sur l'ASP indépendant iasp1, procédez comme suit :

| ServerInstance.sh -delete -servernum 3 -device iasp1

- | • Pour lier les fichiers d'un serveur avec le serveur numéro 2 sur *SYSBASE, procédez comme suit :

| ServerInstance.sh -relink -servernum 2

- Pour lier les fichiers d'un serveur avec le serveur numéro 3 sur l'ASP indépendant iasp1, procédez comme suit :

```
ServerInstance.sh -relink -servernum 3 -device iasp1
```

Remarques relatives aux ASP indépendants pour OmniFind Text Search Server for DB2 for i

Vous pouvez administrer un index de recherche de texte sur un pool de mémoire secondaire (ASP - auxiliary storage pool) indépendant. L'ASP peut être basculé entre plusieurs systèmes, ce qui amène différentes remarques.

Un serveur de recherche de texte local est créé lors de l'installation de OMNIFIND. Pour les ASP indépendants, un serveur de recherche de texte local est créé par un administrateur à l'aide de l'utilitaire ServerInstance (ServerInstance.sh) après la création du groupe d'ASP indépendants.

Une fois le serveur de recherche de texte local créé sur l'ASP indépendant, les données d'index résident sur le système de fichiers de l'ASP indépendant. Les données sont disponibles si l'ASP indépendant bascule sur un autre système. L'administrateur doit créer un serveur de recherche de texte local une fois seulement pour chaque groupe d'ASP indépendants.

Les index de recherche de texte présents dans l'ASP indépendant doivent être contenus dans des serveurs de recherche de texte définis dans l'ASP indépendant. Vous ne pouvez pas afficher un serveur de recherche de texte défini dans un autre groupe d'ASP indépendants ou dans l'ASP indépendant système quand le travail est connecté à l'ASP indépendant.

Pour créer un serveur de recherche de texte sur un ASP indépendant nommé *myiasp*, procédez comme suit :

1. Activez l'ASP indépendant avec la commande CL WRKCFGSTS (Work with Configuration Status) ou avec System i Navigator.
2. Connectez-vous à l'espace de noms du groupe d'ASP indépendants avec la commande CL SETASPGRP (Set Auxiliary Pool Group).
3. Utilisez le script ServerInstance.sh pour créer le serveur de recherche de texte.

Voici un exemple de commande QSH :

```
/QOpenSys/QIBM/ProdData/TextSearch/ServerInstance.sh -create  
-servernum 2 -port nnnnn -device myiasp
```

Dans cette commande, nnnnn indique un numéro de port disponible pour le serveur. Ce port doit être utilisable sur tous les systèmes sur lesquels le groupe d'ASP indépendants peut être basculé.

Une fois le serveur de recherche de texte défini pour le groupe d'ASP indépendants, vous pouvez utiliser les procédures stockées d'administration pour le démarrer et l'arrêter. Vous pouvez aussi utiliser ces procédures stockées pour créer, supprimer et mettre à jour les index de recherche de texte.

Remarque : Les entrées du planificateur de travaux sont ajoutées quand l'ASP indépendant est activé pour tous les index ayant des mises à jour planifiées qui sont présents sur l'ASP indépendant. Les entrées du planificateur de travaux permettent aux mises à jour planifiées de continuer, même quand l'ASP indépendant bascule entre les systèmes.

Limites d'utilisation des index de recherche de texte et des ASP indépendants

- Tous les systèmes sur lesquels l'ASP indépendant peut basculer doivent posséder une installation de OmniFind Text Search Server for DB2 for i et des correctifs de programme provisoire (PTF) du même niveau.
- Ne créez pas d'index de recherche de texte sur un pool de mémoire secondaire (ASP) autre que celui sur lequel l'index de table a été construit.

- Les catalogues système SYSTEXTSERVERS, SYSTEXTINDEXES, SYSTEXTDEFAULTS, SYSTEXTCOLUMNS et SYSTEXTCONFIGURATION ne contiennent pas d'enregistrements pour les index et les serveurs qui sont définis dans un autre groupe d'ASP, en particulier l'ASP système. Les catalogues contiennent uniquement des lignes pour les index et les serveurs qui sont définis pour le groupe d'ASP indépendants auquel le travail est connecté.
- Vous pouvez utiliser les procédures stockées d'administration pour exécuter des fonctions exclusivement sur les index et les serveurs de recherche de texte qui sont définis dans le groupe d'ASP indépendants auquel le travail est connecté.

Remarque : Vous pouvez utiliser les instructions SQL CONTAINS et SCORE quand un travail est connecté à un groupe d'ASP indépendants, même si la colonne est basée sur une table présente dans l'ASP système.

Accessibilité avancée

Vous pouvez implémenter une solution d'accessibilité avancée comprenant des index de recherche de texte à l'aide des API et des commandes existantes. DB2 for IBM i peut reconnaître les index de recherche de texte et exécuter des actions spéciales pendant les opérations de DB2 portant sur ces index.

Remarques sur les index de recherche de texte, l'accessibilité avancée et l'administration de base de données

- Une commande de suppression de fichier (commande CL DLTF) ou une instruction SQL DROP VIEW exécutée sur la vue représentant l'index de recherche de texte a pour effet de supprimer l'index de recherche de texte. La suppression échoue si un contrôle de validation est appliqué.
- La restauration d'une table ou d'un fichier physique enregistré avec un index de recherche de texte sur une colonne n'active pas les déclencheurs qui avaient été créés pour l'index de recherche de texte avant la sauvegarde. Si l'index de recherche de texte est restauré ou créé ensuite, les déclencheurs sont ajoutés pendant la création de l'index. Cette méthode permet aux applications de fonctionner, même si l'index de recherche de texte construit à l'origine sur la table n'est ni restauré ni créé.

Relecture des entrées de journal pour les procédures stockées d'administration de OmniFind

- La relecture d'une entrée de journal (API QDBRPLAY) générée pour la création d'une vue représentant l'index de recherche de texte a pour effet de créer l'index de recherche de texte.
- L'ajout ou la suppression des déclencheurs sur la table de base pendant les appels de procédure SYSTS_CREATE ou SYSTS_DROP n'entraîne pas la journalisation des entrées de journal.
- L'appel de la procédure stockée SYSTS_UPDATE ou SYSTS_REPRIMEINDEX entraîne la journalisation d'une entrée de journal sur la vue de l'index de recherche de texte. La relecture de cette entrée de journal à l'aide de l'API QDBRPLAY a pour effet de relire l'appel de procédure associé.
- Les mises à jour planifiées automatiquement ne génèrent pas d'entrées de journal pour l'index et ne peuvent pas être relues.
- L'appel de la procédure SYSTS_ALTER génère une entrée de journal sur la vue. La relecture de l'entrée de journal à l'aide de l'API QDBRPLAY a pour effet de relire l'appel de procédure SYSTS_ALTER.

Recommandations relatives à l'accessibilité avancée

Si vous voulez implémenter une solution d'accessibilité avancée, prenez en compte les recommandations suivantes :

- Une fois l'index de recherche de texte créé sur le système principal, vous pouvez utiliser l'API QDBRPLAY pour relire la création sur le système de secours.
- Le système de secours contiendra alors un double de l'index dont les mises à jour planifiées s'exécuteront à la même fréquence que sur le système principal.
- A mesure que les modifications d'enregistrement sont relues sur le système de secours basé sur la table, elles sont journalisées dans la table de transfert du système de secours.

- Les appels des procédures SYSTS_UPDATE, SYSTS_REPRIMEINDEX et SYSTS_ALTER sur le système principal génèrent des entrées de journal pouvant être relues sur le système de secours.
- Il n'est PAS nécessaire pour les clients de journaliser et de répliquer les fichiers IFS pour les index de recherche de texte.
- Il *est* nécessaire de vérifier que les déclencheurs ajoutés pendant la procédure SYSTS_CREATE peuvent enregistrer les modifications des enregistrements dans la table de base. Pour identifier ces déclencheurs, vous pouvez appliquer le préfixe spécial "QDBTS" au nom de chaque déclencheur.

Commande CL APYJRNCHG

Vous pouvez utiliser la commande CL Apply Journal Changes (APYJRNCHG) pour relire les événements OmniFind. Si vous voulez utiliser cette commande, regardez soigneusement dans quel ordre les entrées de journal sont relues. La table de transfert QSYS2 doit être synchronisée de manière exacte avec la table de base quand une procédure SYSTS_UPDATE est relue ou appelée.

Les modifications d'enregistrement appliquées à la table de base ne sont PAS journalisées dans la table de transfert pendant le processus APYJRNCHG. Dans certains cas, il peut être nécessaire d'appeler la procédure stockée SYSTS_REPRIMEINDEX pour régénérer l'index après l'exécution de cette commande.

Référence associée

«SYSPROC.SYSTS_REPRIMEINDEX», à la page 75
 La procédure stockée SYSPROC.SYSTS_REPRIMEINDEX permet de réorganiser l'index et de démarrer une mise à jour initiale. Vous pouvez utiliser cette procédure stockée quand vous souhaitez restaurer des données de la table de base.

Analyse de la performance

L'analyse de la performance de OMNIFIND requiert de choisir une définition d'index appropriée, de gérer les documents de manière efficace et de spécifier une recherche sélective.

Informations générales

Le traitement de OmniFind combine un travail de client frontal et un travail de serveur d'arrière-plan, avec des communications entre ces deux travaux. Les communications utilisent des connexions de socket standard.

Le travail client lit les enregistrements depuis la table de base de données appropriée pendant la maintenance et la génération des index. Il traite le fichier journal des modifications de table, envoie les documents pour traitement ultérieur et gère les requêtes de recherche de texte à l'aide des fonctions SQL CONTAINS ou SCORE. Le travail client est celui pendant lequel s'exécute la procédure, par exemple SYSTS_CREATE ou SYSTS_UPDATE, ou la requête contenant CONTAINS ou SCORE. Les performances du travail client varient selon que le traitement des actions de base de données est rapide ou lent et que les documents texte sont extraits et transportés efficacement ou non jusqu'au travail serveur.

Le travail serveur analyse la syntaxe des documents envoyés par le travail client, gère l'index de texte associé avec des insertions et des suppressions, et administre les requêtes de recherche dans l'index. Les performances du travail serveur varient selon sa capacité à communiquer efficacement avec les travaux client et à gérer efficacement les documents. Il existe en général un seul travail serveur pour différents travaux client. Pour cette raison, le travail serveur utilise plusieurs unités d'exécution de manière à pouvoir gérer plusieurs clients.

La table de base de données sous-jacente ne gère pas les modifications avec un index de texte de manière immédiate. Les modifications d'enregistrement effectuées dans la table sont journalisées à l'aide d'un déclencheur de base de données combiné à une table de transfert. La table de transfert enregistre le type

| de modification (insertion, mise à jour, suppression) avec une indication de l'enregistrement modifié dans la table de base. Le texte n'est pas capturé dans le fichier journal. Seules les informations identifiant l'enregistrement modifié sont capturées.

| Les modifications des enregistrements de table ne sont prises en compte dans l'index de texte et, par conséquent, dans les recherches exécutées avec CONTAINS ou SCORE, qu'après l'exécution de la procédure SYSTS_UPDATE suivante.

| Les mises à jour d'un index de texte qui suivent la première mise à jour sont appelées des mises à jour incrémentielles. Ces mises à jour ajoutent ou suppriment des documents dans l'index de texte sur la base des modifications d'enregistrement réalisées dans la table sous-jacente depuis la dernière mise à jour.

| Les modifications réalisées dans la table de base sont enregistrées dans une table de transfert. Lors d'une mise à jour incrémentielle, la table de transfert sert à déterminer les enregistrements de la table de base qui ont changé. Ces enregistrements sont ensuite lus depuis la table de base et leur contenu mis à jour est répercuté dans l'index.

| La mise à jour initiale est un processus plus efficace, par ligne, qu'une mise à jour incrémentielle. La table de base est traitée sans qu'il soit nécessaire de traiter aussi la table de transfert. Pour optimiser les performances, il est donc conseillé d'exécuter la mise à jour initiale sur l'index de texte après le remplissage initial de la table sous-jacente. Cette technique réduit le temps requis pour remplir les documents dans l'index.

| L'option UPDATEAUTOCOMMIT est une option de configuration importante pour les performances. Sa valeur définit à quelle fréquence le travail client de base de données interrompt le traitement des documents. Le client attend que le travail serveur confirme qu'il a traité tous les documents qui lui ont été envoyés. L'option UPDATEAUTOCOMMIT sert de méthode de point de contrôle pour permettre à la base de données de définir les limites des travaux terminés.

| Si la mise à jour de l'index est interrompue puis reprise ensuite, le processus redémarre au niveau du point de contrôle. Comme pour toute interruption, une limite de point de contrôle force le flux de documents à s'arrêter et vide le pipeline existant entre le client et le serveur. Ce processus de démarrage et d'arrêt peut avoir un impact négatif considérable sur les performances.

| La valeur par défaut de l'option UPDATEAUTOCOMMIT est 100. Elle déclenche des points de contrôle fréquents. Si vous entrez une valeur plus élevée, les appels de la procédure SYSTS_UPDATE s'exécutent en général plus rapidement. Toutefois, une valeur plus élevée n'induit pas forcément un délai de reprise plus long si la mise à jour est annulée et redémarrée.

| Si vous affectez à l'option UPDATEAUTOCOMMIT une valeur élevée (ou zéro pour n'avoir aucun point de contrôle), le temps de réponse est raccourci. Toutefois, si la mise à jour initiale est annulée, OmniFind doit reprendre depuis le début de la génération d'index puisqu'il n'y a aucun point de contrôle.

| Pour certains clients, une valeur comprise entre 5000 et 20000 procure un équilibre raisonnable entre les performances et la capacité de reprise sur point de point de contrôle.

| **Choix de la définition d'index appropriée**

| Vous pouvez spécifier un index de texte avec un type de configuration FORMAT. Il existe quatre types disponibles : TEXT, HTML, XML et INSO.

- | 1. TEXT est habituellement le format le plus efficace. Le texte est lu dans l'enregistrement de base de données puis envoyé au serveur de texte qui le traite directement.
- | 2. HTML s'utilise quand les documents sont reconnus comme étant au format HyperText Markup Language (HTML). Pendant le traitement du texte, les valeurs de contrôle de marquage présentes dans le document sont ignorées.

- | 3. XML s'utilise quand les documents sont reconnus comme étant au format eXtensible Markup Language (XML). La structure du contenu du document est analysée, les éléments sont suivis ainsi que les attributs et la hiérarchie interne du document. La marquage d'un index en tant que balise XML permet d'exécuter des recherches XML à l'aide du langage de recherche xmlxp (xpath) sur les fonctions CONTAINS et SCORE.
- | 4. Le format INSO est utilisé quand le document doit être traité dans le mode INSide Out. Dans ce format, le contenu de chaque document est réputé être plus complexe que du simple texte. Chaque document est préanalysé afin de déterminer de quel type de document il s'agit. Les documents INSO sont en général des documents de texte enrichi générés par des programmes de traitement de texte.

| Du point de vue des performances, le travail d'indexation d'un document est plus important quand vous passez du format TEXT au format INSO. Le format INSO gérant le texte simple, il occasionne plus de charge que le format TEXT, car une phase d'interprétation est nécessaire. Utilisez le format TEXT quand la colonne de base de données contient un document de texte simple.

| L'option de configuration LANGUAGE n'est pas strictement obligatoire car OmniFind détermine la langue du document sur la base d'un examen. Toutefois, si la langue du document est connue, vous pouvez la spécifier dans l'option LANGUAGE pour augmenter les performances. Ceci limite le travail d'interprétation requis.

| **Gestion efficace des documents**

| Le travail client et le travail serveur communiquent les données des documents dans le format UTF (Unicode Transformation Format). Pour obtenir le document dans ce format, le client lit le document depuis la base de données et convertit les données lues dans le format UTF-8, également appelé CCSID 1208. Tout le texte est converti en UTF-8 avant d'être transmis au serveur.

| Pour améliorer les performances, affectez à la colonne de texte à indexer la valeur CCSID 1208 pour éviter cette conversion et améliorer l'efficacité du processus de gestion de document.

| **Spécification d'une recherche sélective**

| Les recherches de texte se font à l'aide des fonctions CONTAINS ou SCORE à l'intérieur d'une instruction de requête SQL. Ces recherches comparent les critères de recherche de la fonction aux documents associés à la colonne analysée. Les enregistrements correspondants sont identifiés et sélectionnés. Le travail client envoie la requête de recherche au serveur et reçoit une réponse indiquant si une occurrence a été trouvée.

| Comme pour tous les critères de recherche, plus la recherche est sélective, plus elle est efficace. Rechercher des termes usuels, comme l'article "le" en français, produit de nombreux résultats de recherche mais peut affecter négativement les performances. En pratique, ce type de recherche fournit des informations peu significatives. Si vous indiquez des termes de recherche plus spécifiques, vous obtenez moins de résultats mais ils sont plus pertinents.

| Quand vous utilisez CONTAINS dans la clause WHERE d'une instruction SQL, cela fonctionne mieux si vous l'associez à d'autres critères avec l'opérateur AND. Par exemple :

```
| SELECT bn, pubdate, description  
| FROM myBooks  
| WHERE CONTAINS(description,'Alladin') = 1  
| AND Pubdate > '2004-01-01'
```

| **Utilisation de CONTAINS et SCORE**

| L'optimiseur peut améliorer les performances des fonctions CONTAINS et SCORE en combinant de manière interne et en remplaçant ces fonctions définies par l'utilisateur (UDF) par une fonction de table définie par l'utilisateur (UDTF). L'UDTF renvoie la liste des documents correspondants sous la forme

| d'un résultat unique. Dans de nombreux cas, ce traitement à base d'UDTF fonctionne mieux que l'autre
| processus qui consiste à appeler l'UDF pour chaque enregistrement afin d'identifier une occurrence.

| Pour activer l'optimiseur afin d'exécuter une réécriture de l'UDTF, la fonction CONTAINS doit :

- | • résider dans la clause WHERE de l'instruction SQL,
- | • pouvoir être isolée (avec l'opérateur AND) de tout autre prédicat dans la clause WHERE,
- | • être égale à la valeur 1.

| Par exemple, la clause suivante :

| WHERE CONTAINS(MyDocuments, 'java performance') = 1

| peut être réécrite par l'optimiseur sous la forme d'une UDTF. Par contre, la clause suivante :

| WHERE CONTAINS(MyDocuments, 'java performance') = 1 OR price >100

| ne peut pas être réécrite car la fonction CONTAINS est associée aux autres prédicats avec l'opérateur OR.

| **Remarque :** Quand la réécriture d'une UDTF est activée, l'optimiseur utilise quand même un
| comparateur de coût pour choisir le meilleur plan.

| **Référence associée**

| «CONTAINS», à la page 38

| Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de
| critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si
| une occurrence a été trouvée.

| «SCORE», à la page 41

| Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de
| critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence
| qui mesure à quel degré un document correspond à la requête.

| «SYSPROC.SYSTS_UPDATE», à la page 35

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_UPDATE pour mettre à jour l'index de
| recherche de texte avec le contenu actuel de la colonne de texte.

| «SYSPROC.SYSTS_CREATE», à la page 19

| Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche
| de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte.

| Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions
| CONTAINS ou SCORE.

| **Remarques relatives aux transactions**

| Prenez en compte votre environnement quand vous décidez de la fréquence des mises à jour des index
| de texte à partir des données sous-jacentes des tables de base de données.

| Une notion importante pour les utilisateurs familiers des bases de données est le concept de limite et de
| traitement des transactions. A titre d'exemple, prenons une transaction bancaire dans laquelle une somme
| d'argent est transférée d'un compte bancaire vers un autre. Le transfert est considéré comme ne formant
| qu'une seule transaction, qu'il soit effectif ou non. Le client n'appréciera pas de voir son argent
| disparaître du premier compte sans qu'il apparaisse aussitôt sur l'autre compte. Inversement, la banque
| ne souhaite que la somme apparaisse sur les deux comptes avant la fin de la transaction. Le principe est
| que, si une modification a lieu, elle doit être répercutée sur les deux comptes de manière immédiate.

| Il existe de nombreux exemples en matière de gestion de l'information où les délais sont plus acceptables,
| voire attendus. Dans la conception d'un entrepôt de données traditionnel, le contenu de l'entrepôt de
| données est souvent différent des données opérationnelles pendant plusieurs heures ou davantage. Cette
| acceptation des délais découle d'une combinaison d'attentes réalistes à l'égard d'un entrepôt de données
| et du souhait de disposer de données cohérentes et prévisibles.

| Les recherches de texte non structurées se situent entre ces limites. En pratique, l'indexation des documents texte est un processus intensif qui analyse et décompose la signification sous-jacente des mots présents dans les documents. Par exemple, la recherche de texte permet de rechercher le terme "cheval" et de trouver des documents contenant le terme "chevaux". On peut obtenir ces résultats grâce à des techniques d'indexation qui permettent de décomposer les mots en fonction de leur sens. Cette analyse a lieu au moment où le document est indexé de manière à rendre les recherches les plus rapides possible.

| Quand vous utilisez un index de base de données classique, vous escomptez que cet index reflète l'état courant des données de la table de base de données. Cette attente n'est pas satisfaite avec un index de texte. Le contenu de l'index de texte reflète l'état de la table telle qu'elle était au moment de la dernière mise à jour (SYSTS_UPDATE).

| Dans un environnement fréquemment modifié, il est peu probable que l'index de texte reflète l'état courant de la table à chaque instant. En revanche, dans un environnement plus prévisible où la table de base de données est mise à jour moins souvent, ou en mode de traitement par lots, les mises à jour de l'index de texte peuvent être planifiées pour s'exécuter après la mise à jour de la table afin de refléter son état de manière exacte.

| Il est important d'avoir les attentes appropriées pour utiliser un index de texte. Appliquez l'option UPDATE FREQUENCY à l'index de texte «SYSPROC.SYSTS_CREATE», à la page 19 ou «SYSPROC.SYSTS_UPDATE», à la page 35, ou appelez explicitement la procédure SYSTS_UPDATE pour mettre à jour le contenu de l'index de texte de manière appropriée.

| Pour les environnements plus statiques, utilisant des chargements de données en bloc par exemple, il est pertinent de planifier la mise à jour de l'index de texte de manière qu'elle s'exécute après le chargement en bloc. Pour les environnements utilisant plutôt des transactions, la valeur de UPDATE FREQUENCY peut être une durée courte, ou vous pouvez utiliser la procédure SYSTS_UPDATE de manière fréquente. Il est habituellement exact que plus les mises à jour sont fréquentes, plus la charge de l'ordinateur est importante.

| **Référence associée**

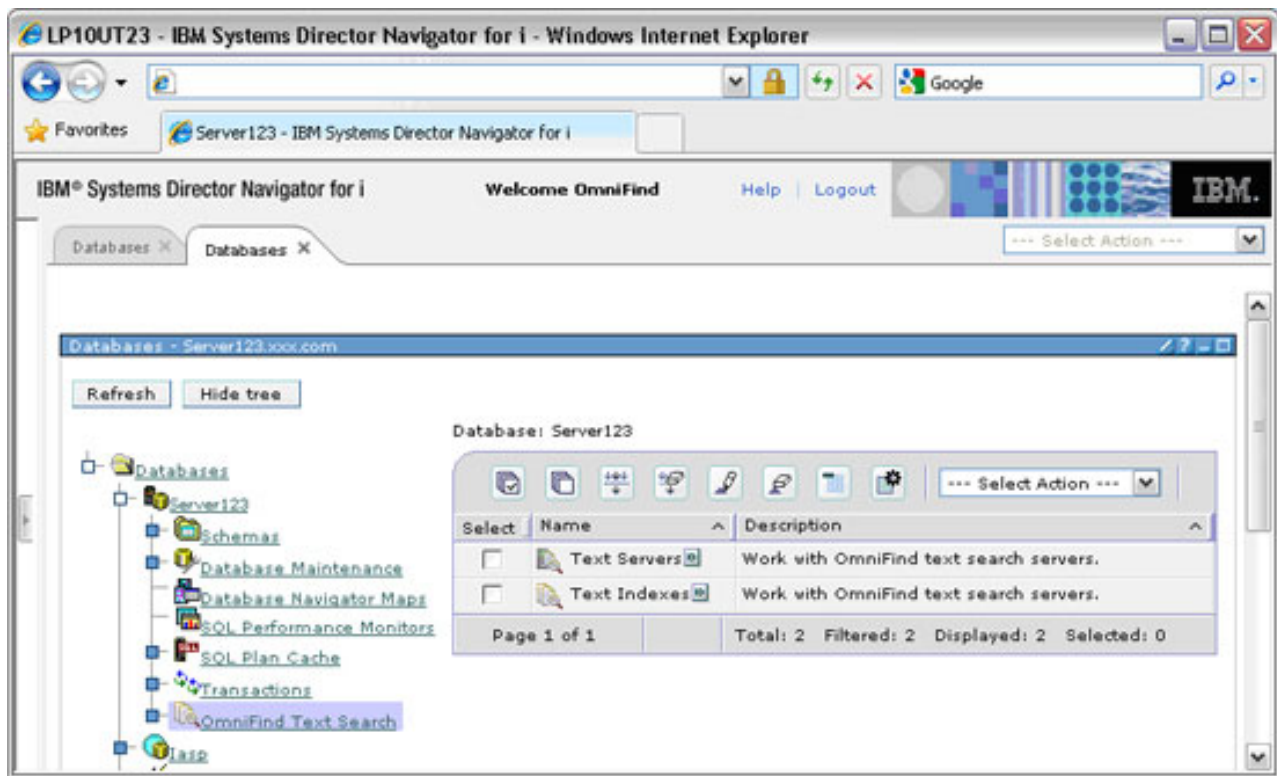
| «Table d'administration QSYS2.SYSTEXTINDEXES», à la page 103

| La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de schéma et un nom de collection associée sur le serveur de recherche de texte.

| **Utilisation d'IBM Systems Director Navigator for i**

| Vous pouvez administrer vos serveurs de recherche de texte OmniFind et les index de recherche de texte à l'aide du programme IBM Systems Director Navigator.

- | 1. Dans la fenêtre **IBM Systems Director Navigator**, développez le système que vous voulez utiliser.
- | 2. Développez le noeud **Databases**.
- | 3. Développez la base de données que vous voulez utiliser.
- | 4. Sélectionnez **OmniFind Text Search**.

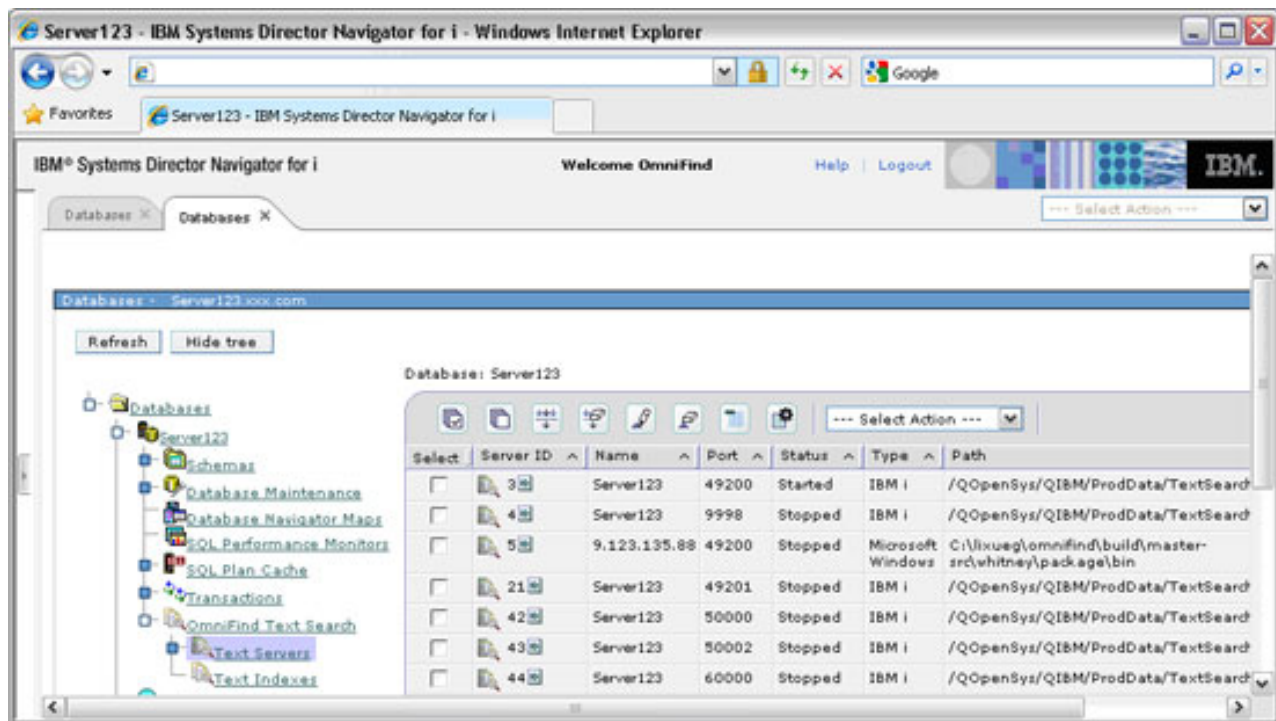


Utilisation des serveurs de recherche de texte

Vous pouvez démarrer et arrêter vos serveurs de recherche de texte OmniFind et créer un index de recherche de texte à l'aide du programme IBM Systems Director Navigator.

Affichage du statut des serveurs de texte OmniFind :

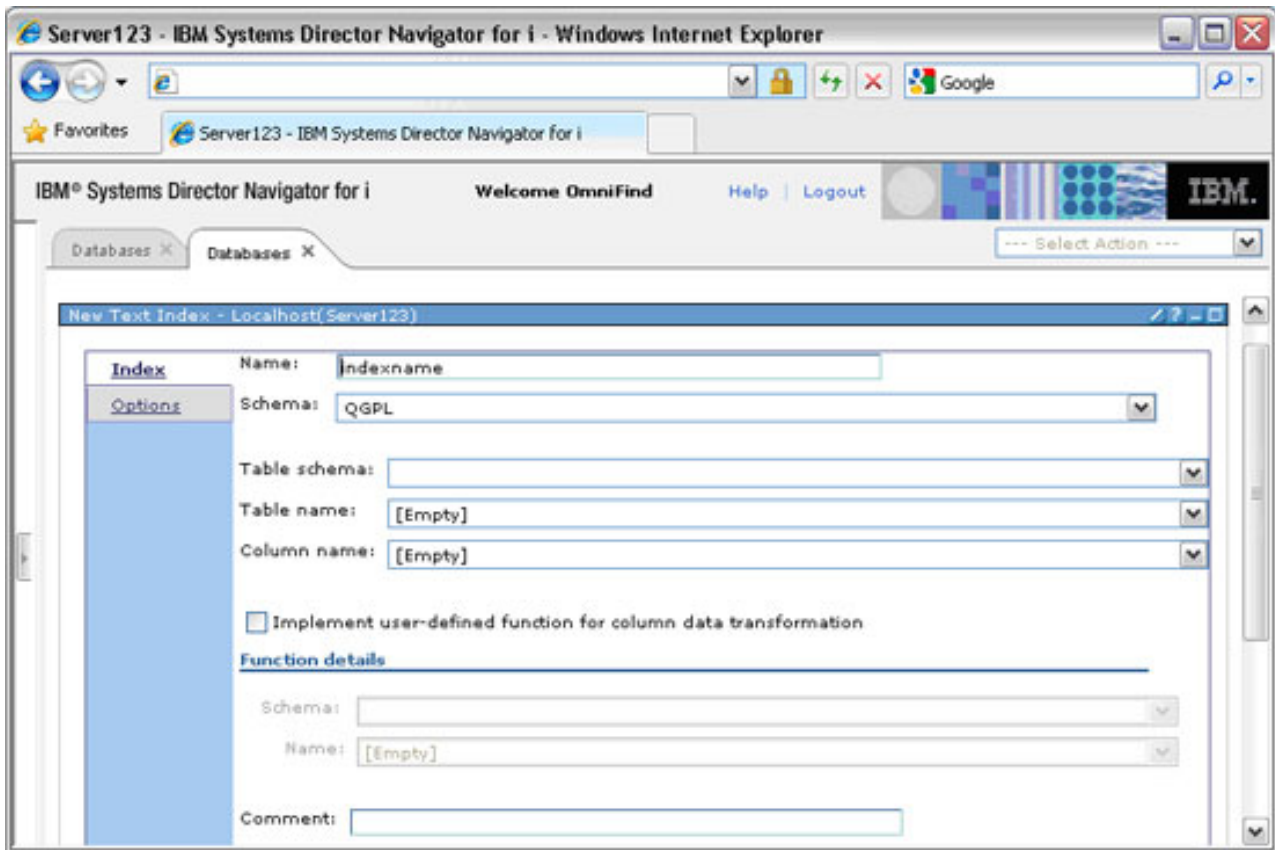
1. Sélectionnez le dossier Text Servers dans la sous-fenêtre de droite.
2. Dans la sous-fenêtre de droite, regardez le statut des serveurs de recherche de texte actuellement configurés sur le système.
3. Sélectionnez **Refresh** pour régénérer la liste des serveurs.



Démarrage et arrêt des serveurs de texte OmniFind :

1. Sélectionnez la case située devant l'ID du serveur sélectionné dans la sous-fenêtre de droite.
2. Dans le menu **Actions**, sélectionnez au choix :
 - Démarrage du serveur
 - Arrêt du serveur
 - Création d'un index de recherche de texte

Création d'un index de recherche de texte :

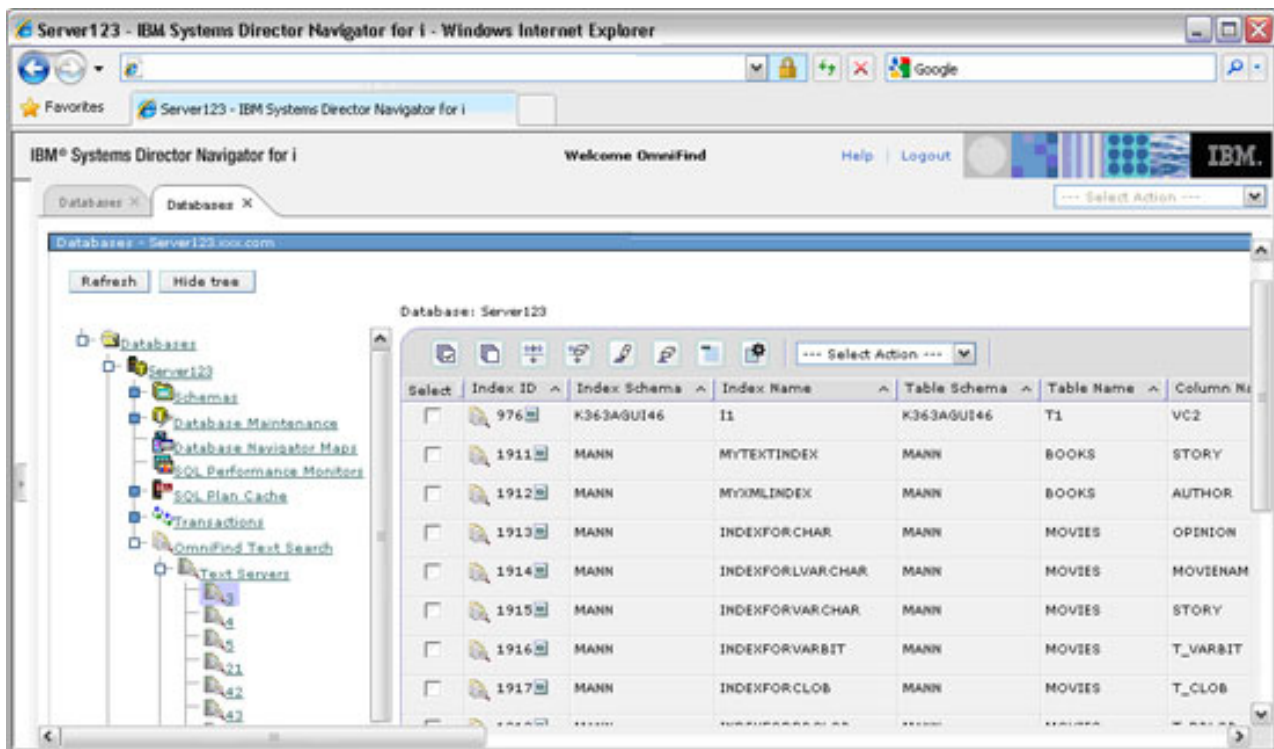


Utilisation des index de recherche de texte

Vous pouvez exécuter des opérations sur un index de recherche de texte sur un système à l'aide du programme IBM Systems Director Navigator.

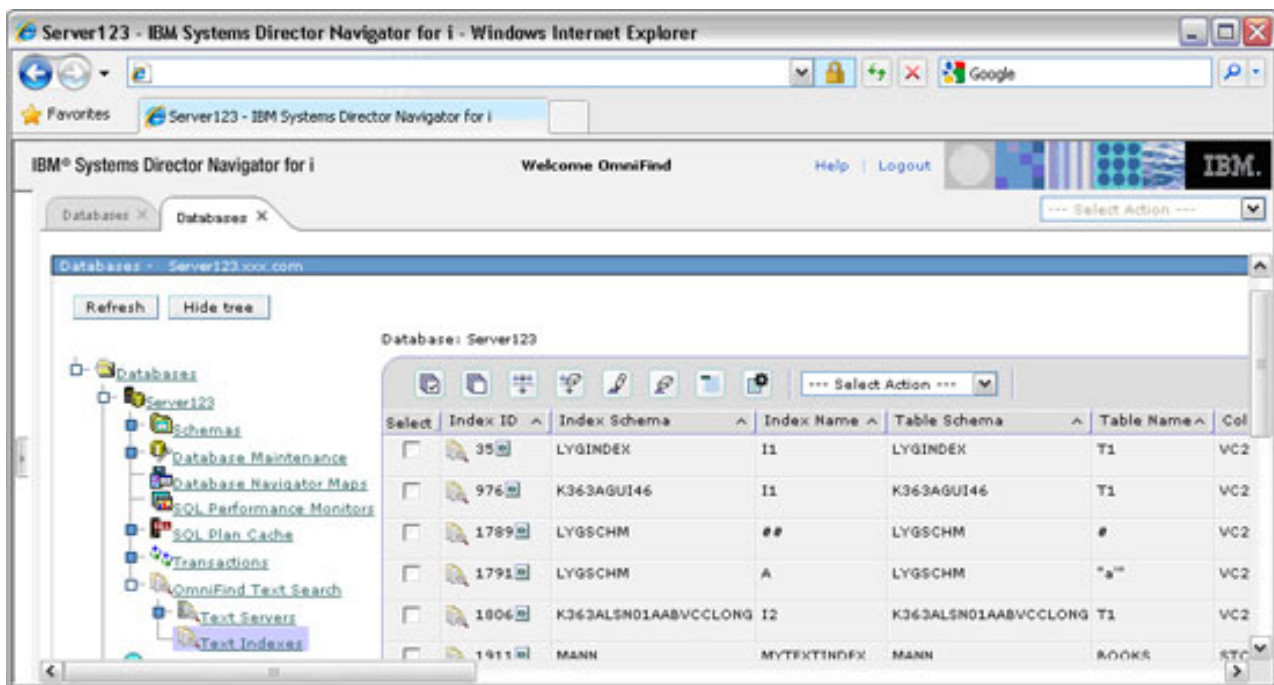
Affichage des index de texte OmniFind pour un serveur de texte :

1. Dans la fenêtre **IBM Systems Director Navigator**, développez le système que vous voulez utiliser.
2. Développez le nœud **Databases**.
3. Développez la base de données que vous voulez utiliser.
4. Sélectionnez **OmniFind Text Search**.
5. Sélectionnez **Text Servers**.
6. Sélectionnez le serveur de texte que vous voulez utiliser. Les index associés à ce serveur apparaissent dans la sous-fenêtre de droite.



Affichage des index de texte OmniFind associés au système :

1. Sélectionnez **OmniFind Text Search**.
2. Sélectionnez **Text Indexes**. Les index affichés représentent les index de recherche de texte contenus dans la partition active du système associé.

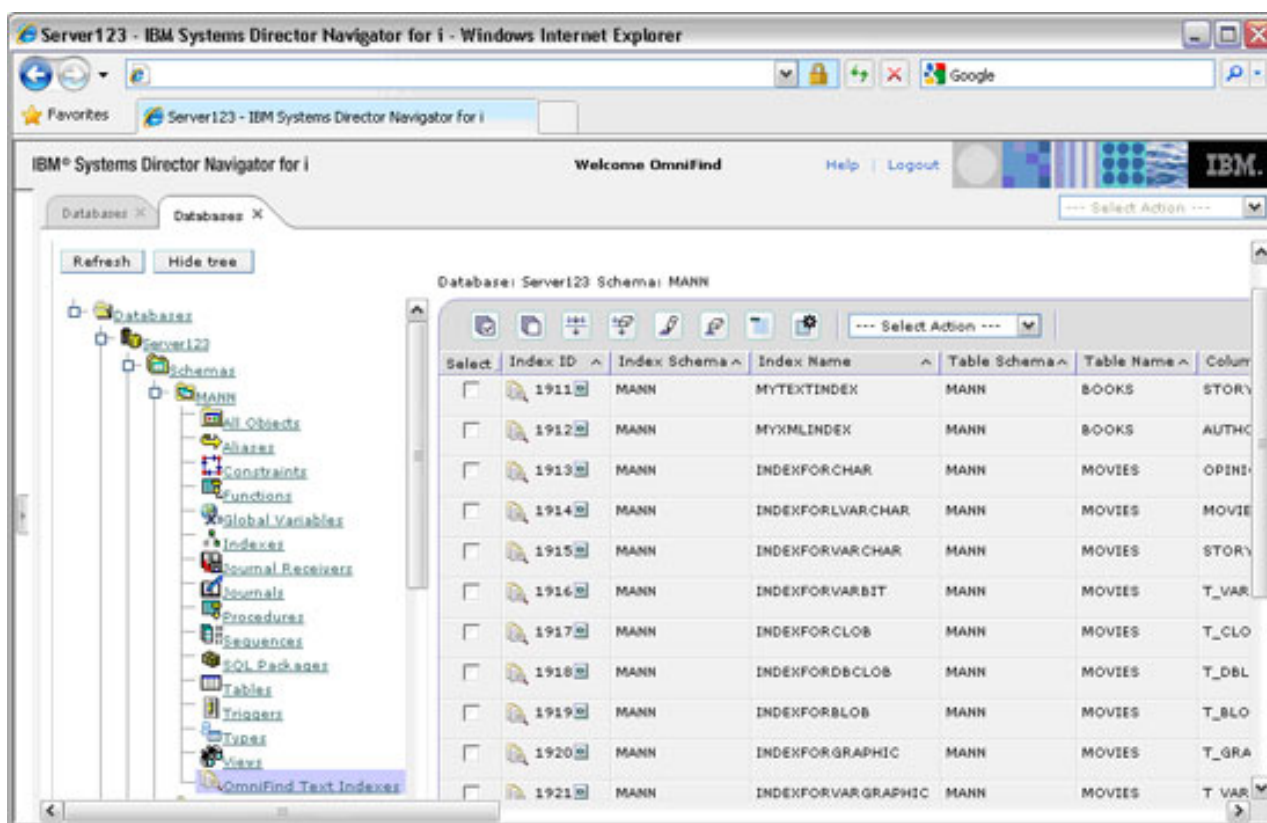


Exécutez une opération sur un index de texte :

1. Sélectionnez dans la sous-fenêtre de droite la case de l'index que vous voulez utiliser.
2. Dans la zone située en haut du panneau, sélectionnez une **Action**.
 - mettre à jour
 - réorganiser
 - suppression
 - voir la description
 - voir la définition
 - modifier la définition

Affichage des index de texte OmniFind pour un schéma :

1. Dans la fenêtre **IBM Systems Director Navigator**, développez le système que vous voulez utiliser.
2. Développez le noeud **Databases**.
3. Développez la base de données que vous voulez utiliser.
4. Développez **Schemas**.
5. Développez le schéma que vous voulez utiliser.
6. Sélectionnez **OmniFind Text Indexes**. Les index affichés représentent les index de recherche de texte associé au schéma sélectionné.

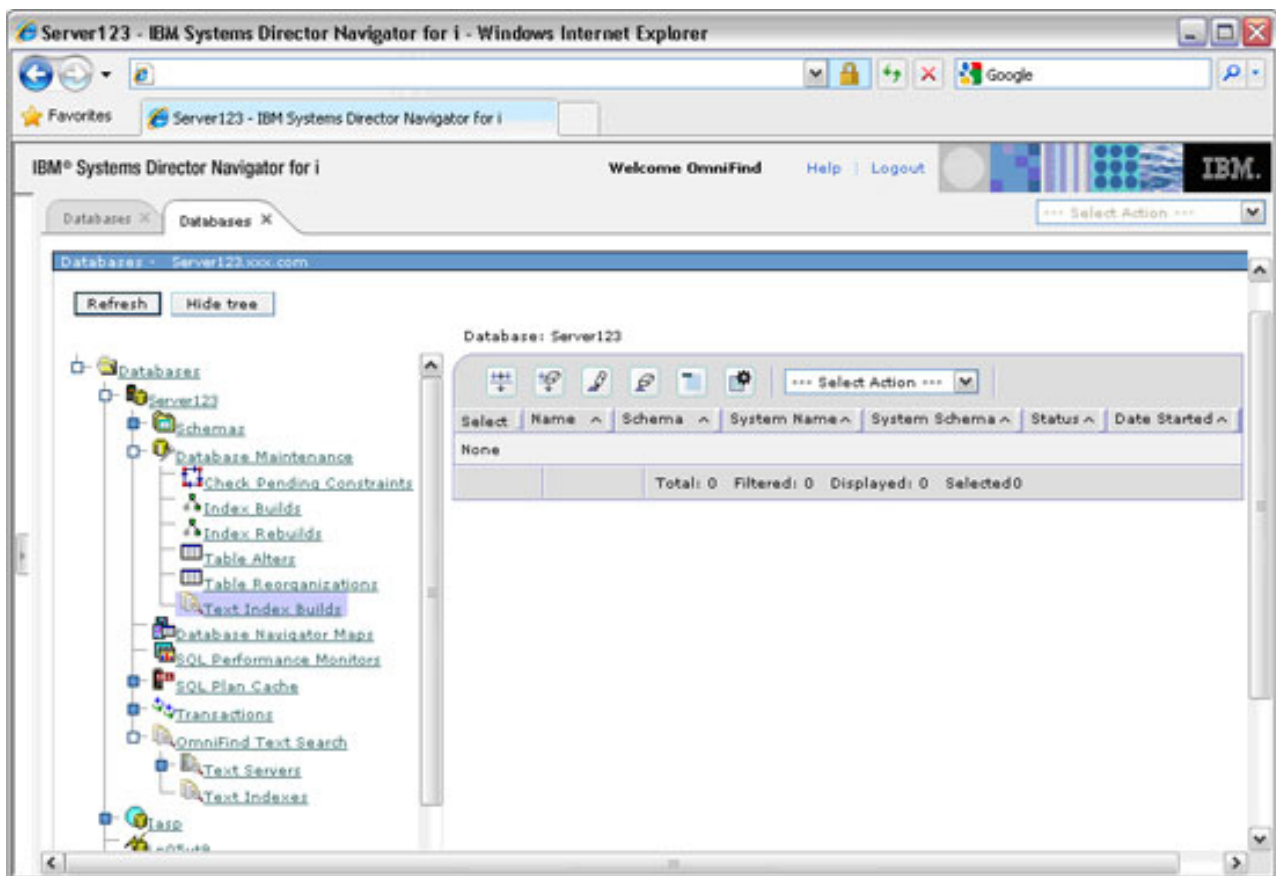


Affichage des générations d'index de recherche de texte :

- Vous pouvez visualiser les index de texte générés par la base de données à l'aide du programme IBM Systems Director Navigator. Cette vue permet de déterminer quand les index de recherche de texte deviennent disponibles pour vos applications.
- Pour afficher les index de recherche de texte générés, procédez comme suit :

1. Dans la fenêtre **IBM Systems Director Navigator**, développez le système que vous voulez utiliser.
2. Développez le noeud **Databases**.
3. Développez la base de données que vous voulez utiliser.
4. Sélectionnez **Database Maintenance**.
5. Sélectionnez **Text Index Builds**.

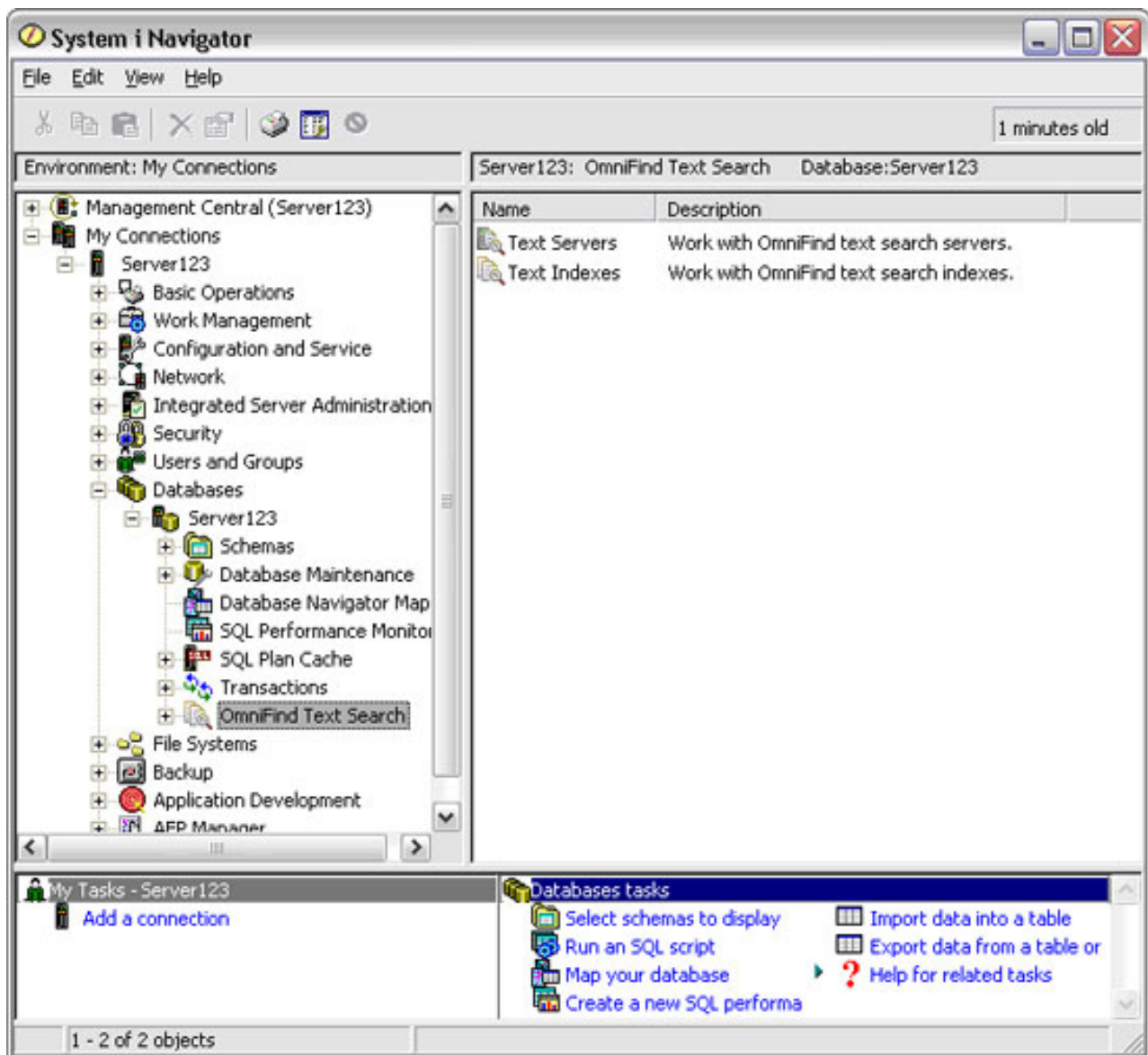
Ce panneau affiche uniquement les mises à jour d'index de recherche de texte en cours d'exécution. Il est vide s'il n'existe pas de mises à jour ou de réorganisations en cours sur le système.



Utilisation de System i Navigator

Vous pouvez administrer vos serveurs de recherche de texte OmniFind et les index de recherche de texte à l'aide du programme System i Navigator.

1. Dans la fenêtre **System i Navigator**, développez le système que vous voulez utiliser.
2. Développez le noeud **Databases**.
3. Développez la base de données que vous voulez utiliser.
4. Sélectionnez **OmniFind Text Search**.

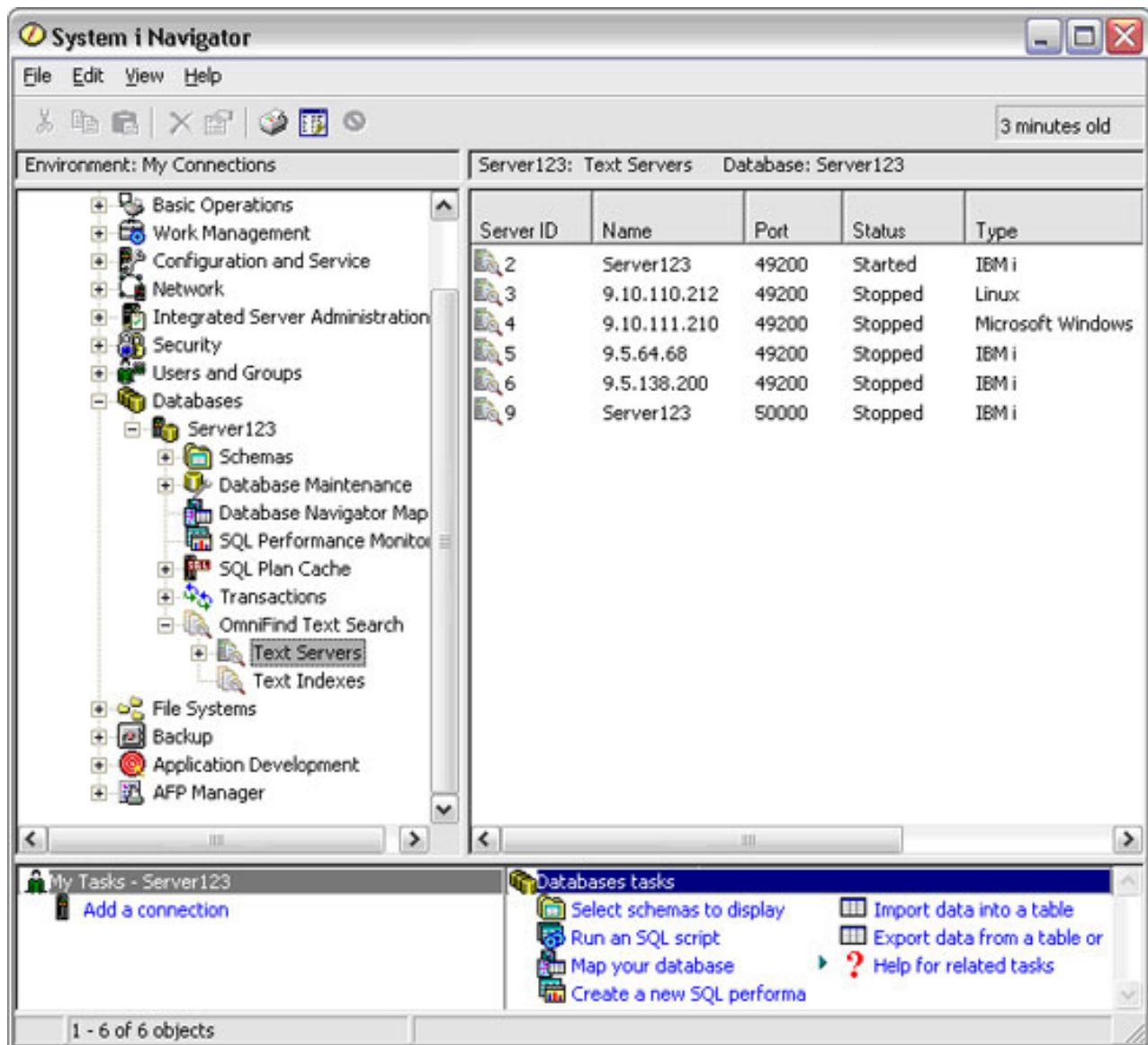


Utilisation des serveurs de recherche de texte

Vous pouvez démarrer et arrêter vos serveurs de recherche de texte OmniFind et créer un index de recherche de texte à l'aide du programme System i Navigator.

Affichage du statut des serveurs de texte OmniFind :

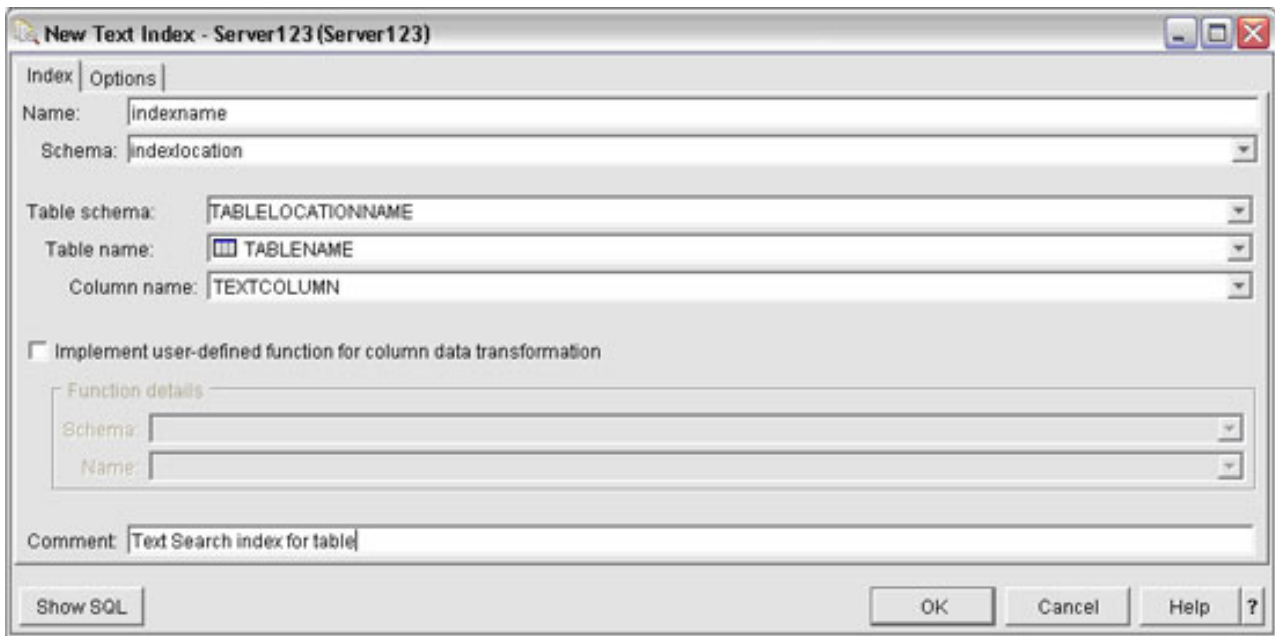
1. Sélectionnez le dossier Text Servers dans la sous-fenêtre de droite.
2. Dans la sous-fenêtre de droite, regardez le statut des serveurs de recherche de texte actuellement configurés sur le système.
3. Appuyez sur F5 pour régénérer la liste des serveurs.



Démarrage et arrêt des serveurs de texte OmniFind :

1. Sélectionnez l'ID du serveur sélectionné dans la sous-fenêtre de droite.
2. Cliquez avec le bouton droit pour afficher les options :
 - Démarrage du serveur
 - Arrêt du serveur
 - Création d'un index de recherche de texte

Création d'un index de recherche de texte :

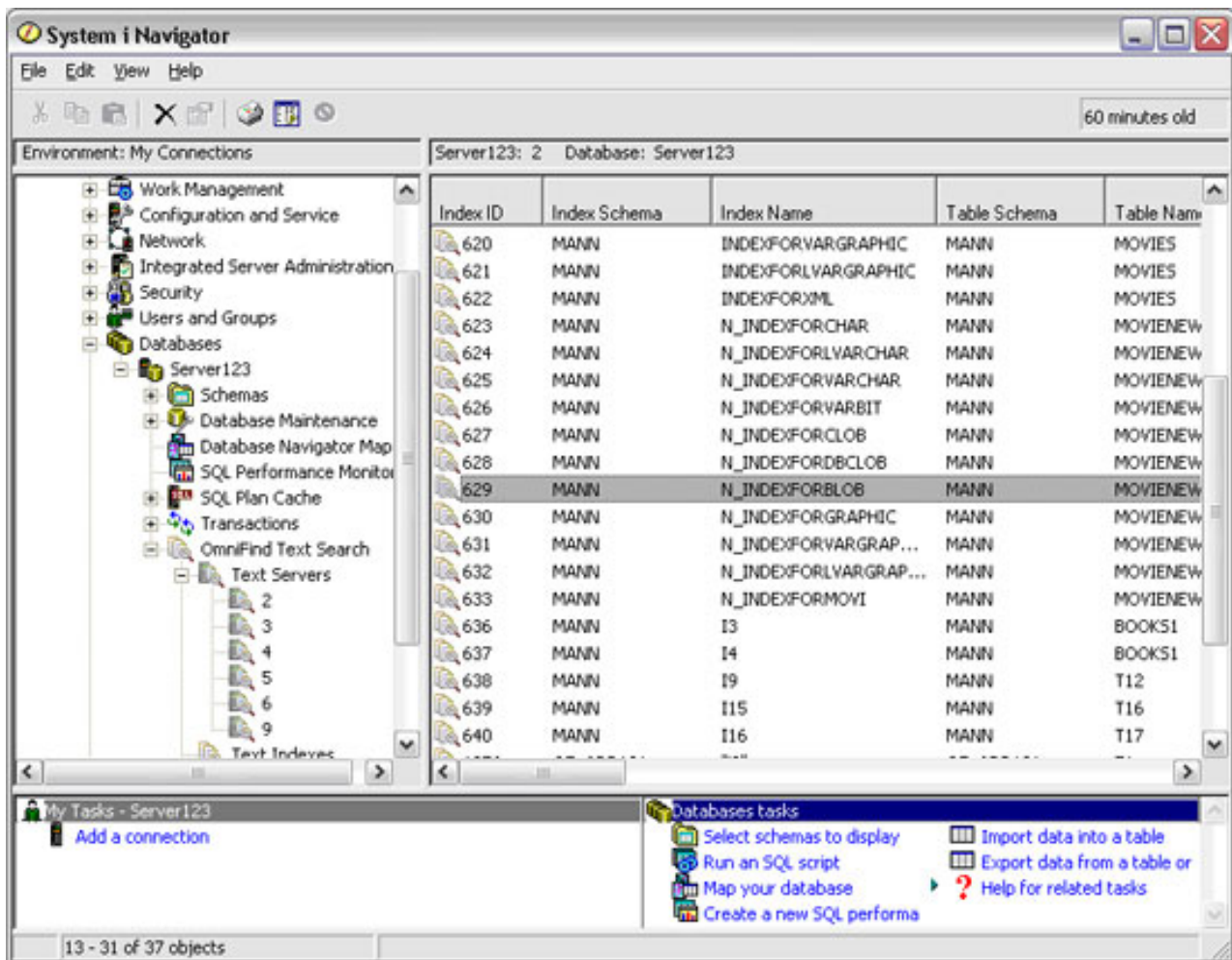


Utilisation des index de recherche de texte

Vous pouvez exécuter des opérations sur un index de recherche de texte sur un système à l'aide du programme System i Navigator.

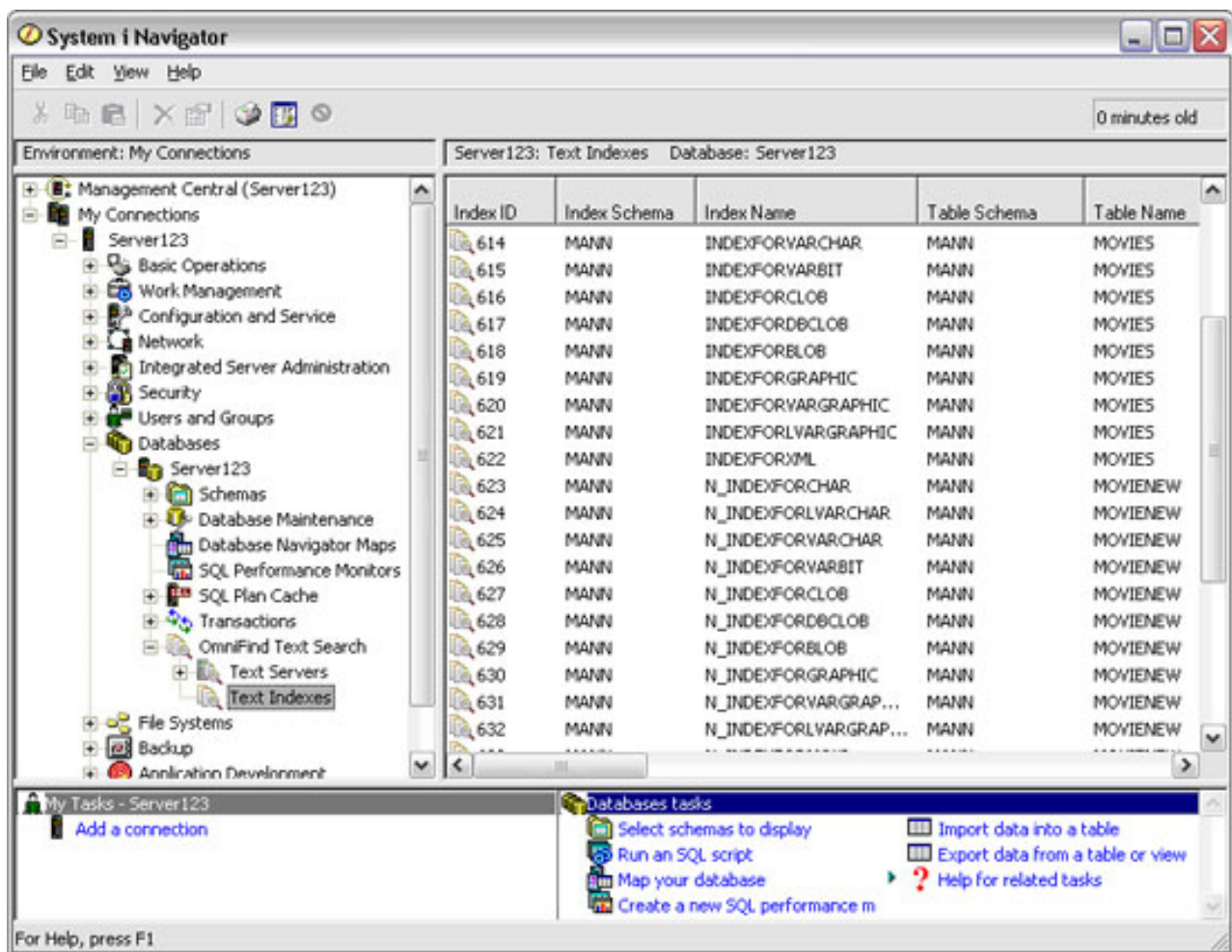
Affichage des index de texte OmniFind pour un serveur de texte :

1. Dans la fenêtre **System i Navigator**, développez le système que vous voulez utiliser.
2. Développez le noeud **Databases**.
3. Développez la base de données que vous voulez utiliser.
4. Sélectionnez **OmniFind Text Search**.
5. Sélectionnez **Text Servers**.
6. Sélectionnez le serveur de texte que vous voulez utiliser. Les index associés à ce serveur apparaissent dans la sous-fenêtre de droite.



Affichage des index de texte OmniFind associés au système :

1. Sélectionnez **OmniFind Text Search**.
2. Sélectionnez **Text Indexes**. Les index affichés représentent les index de recherche de texte contenus dans la partition active du système associé.

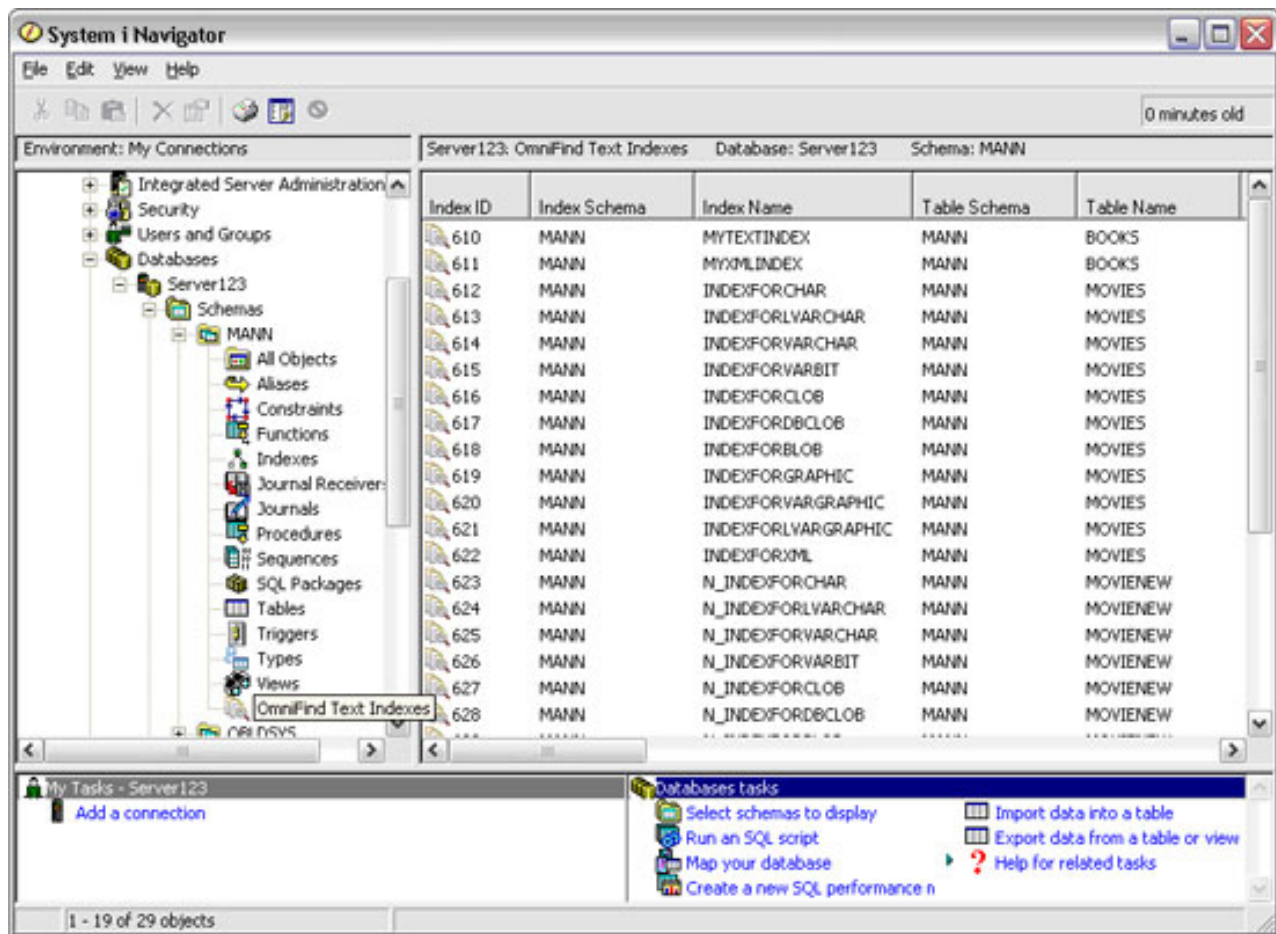


Exécutez une opération sur un index de texte :

1. Sélectionnez dans la sous-fenêtre de droite l'index que vous voulez utiliser.
2. Cliquez avec le bouton droit sur l'index.
3. Sélectionnez une opération :
 - mettre à jour
 - réorganiser
 - supprimer
 - voir la description
 - voir la définition
 - modifier la définition

Affichage des index de texte OmniFind pour un schéma :

1. Dans la fenêtre **System i Navigator**, développez le système que vous voulez utiliser.
2. Développez le noeud **Databases**.
3. Développez la base de données que vous voulez utiliser.
4. Développez **Schemas**.
5. Développez le schéma que vous voulez utiliser.
6. Sélectionnez **OmniFind Text Indexes**. Les index affichés représentent les index de recherche de texte associé au schéma sélectionné.



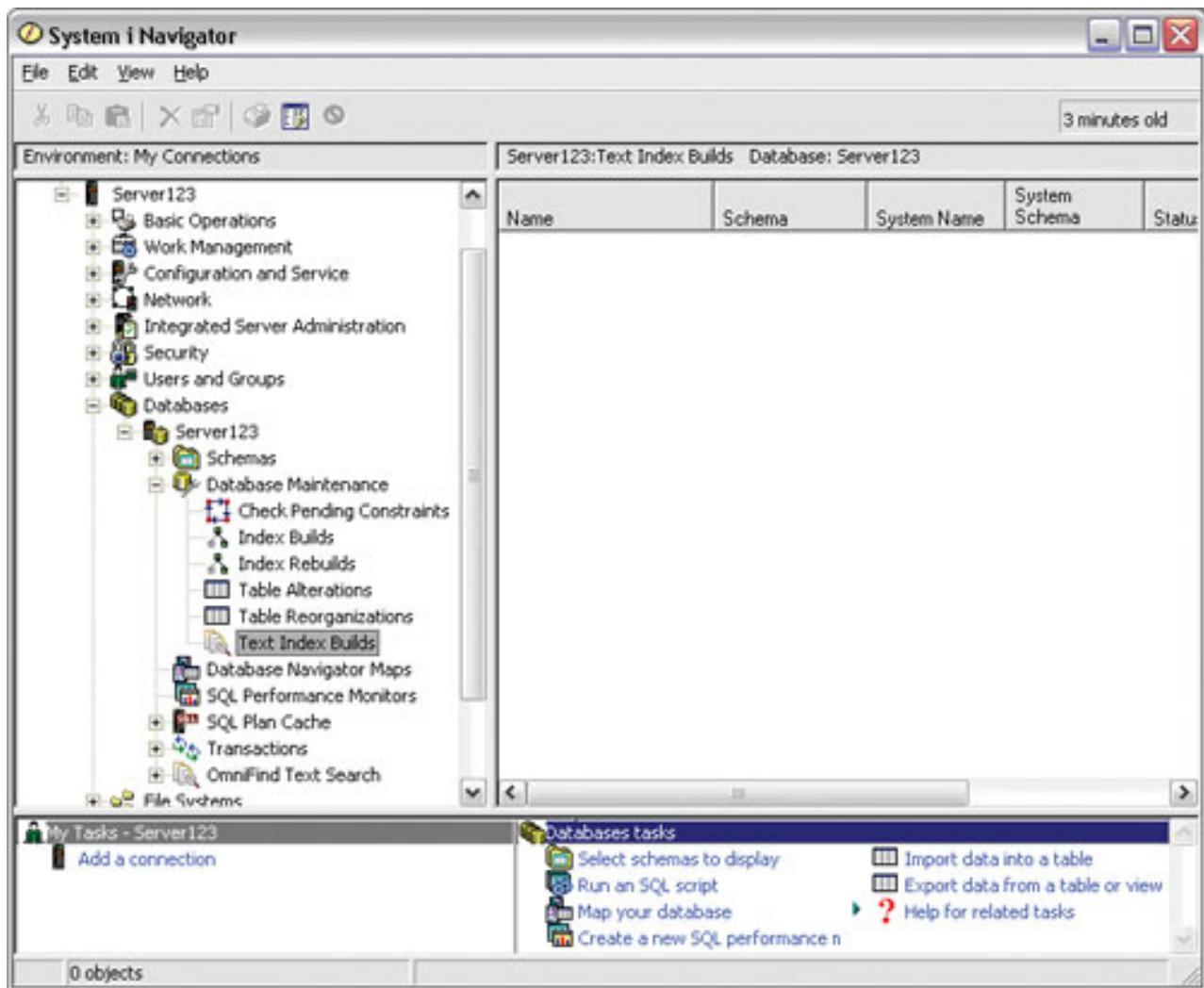
Affichage des générations d'index de recherche de texte :

Vous pouvez visualiser les index de texte générés par la base de données à l'aide du programme System i Navigator. Cette vue permet de déterminer quand les index de recherche de texte deviennent disponibles pour vos applications.

Pour afficher les index de recherche de texte générés, procédez comme suit :

1. Dans la fenêtre **System i Navigator**, développez le système que vous voulez utiliser.
2. Développez le noeud **Databases**.
3. Développez la base de données que vous voulez utiliser.
4. Développez **Database Maintenance**.
5. Sélectionnez **Text Index Builds**.

Ce panneau affiche uniquement les mises à jour d'index de recherche de texte en cours d'exécution. Il est vide s'il n'existe pas de mises à jour ou de réorganisations en cours sur le système.



Tables d'administration de recherche de texte

Les tables d'administration de QSYS2 permettent de prendre en charge les serveurs de recherche de texte et les index.

Table d'administration QSYS2.SYSTEXTDEFAULTS

La table d'administration QSYS2.SYSTEXTDEFAULTS contient des paramètres par défaut et leurs valeurs. Cette table est créée pendant l'installation de OmniFind Text Search for DB2 for i.

Le tableau suivant illustre le contenu de la table d'administration QSYS2.SYSTEXTDEFAULTS.

Tableau 31. Contenu de la table d'administration QSYS2.SYSTEXTDEFAULTS

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
NAME	VARCHAR(30)	Non	Nom d'un paramètre par défaut de la base de données utilisée pour la recherche de texte.
VALUE	VARCHAR(512)	Non	Valeur du paramètre par défaut de la recherche de texte.

Tableau 31. Contenu de la table d'administration QSYS2.SYSTEXTDEFAULTS (suite)

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
TYPE	INTEGER	Non	Réservé.

Table d'administration QSYS2.SYSTEXTINDEXES

La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de schéma et un nom de collection associée sur le serveur de recherche de texte.

Le tableau suivant répertorie le contenu de la table d'administration QSYS2.SYSTEXTINDEXES. La clé unique de cette table est la colonne INDEXSCHEMA associée à la colonne INDEXNAME. La clé primaire est la colonne INDEXID.

Tableau 32. Contenu de la table d'administration QSYS2.SYSTEXTINDEXES

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
INDEXID	INTEGER	Non	ID d'index unique généré pour l'index de recherche de texte
INDEXSCHEMA	VARCHAR(128)	Non	Nom de schéma de l'index de recherche de texte
INDEXNAME	VARCHAR(128)	Non	Nom non qualifié de l'index de recherche de texte
TABLESCHEMA	VARCHAR(128)	Non	Nom de schéma de la table de base
TABLENAME	VARCHAR(128)	Non	Nom non qualifié de la table de base
TABLEIASP	SMALLINT	Non	ASP indépendant de la table de base
COLLECTIONNAME	VARCHAR(255)	Non	Nom de la collection associée sur le serveur de recherche de texte
SERVERID	INTEGER	Non	ID de serveur associé à l'index de recherche de texte
TAKEOVERSERVERID	INTEGER	Oui	Réservé pour une utilisation ultérieure
TAKEOVERSERVERPULSE	TIMESTAMP	Oui	Réservé pour une utilisation ultérieure
SEARCHARGS	VARBINARY(1024)	Oui	Réservé pour une utilisation ultérieure
ALIASSCHEMA	VARCHAR(128)	Non	Alias du schéma de la table de base utilisée dans la procédure stockée SYSPROC.SYSTS_CREATE. Si aucun alias n'est spécifié, cette valeur est identique à TABLESCHEMA.
ALIASNAME	VARCHAR(128)	Non	Alias du nom de la table de base utilisée dans la procédure stockée SYSPROC.SYSTS_CREATE. Si aucun alias n'est spécifié, cette valeur est identique à TABLENAME.
STAGINGTABLENAME	VARCHAR(128)	Oui	Nom de la table de journal utilisée pour l'index de recherche de texte
EVENTTABLENAME	VARCHAR(128)	Non	Nom de la table d'événements utilisée pour l'index de recherche de texte

Tableau 32. Contenu de la table d'administration QSYS2.SYSTEXTINDEXES (suite)

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
OFINDEXTABLENAME	VARCHAR(128)	Non	Nom de la table utilisée pour l'index de recherche de texte dans OmniFind Text Search Server for DB2 for i.
UPDATEMINIMUM	INTEGER	Non	Nombre minimum d'entrées devant exister dans la table de journal pour déclencher une mise à jour incrémentielle de l'index de recherche de texte
UPDATEFREQUENCY	VARCHAR(512)	Non	Fréquence de mise à jour de l'index de recherche de texte spécifiée par la procédure stockée SYSPROC.SYSTS_CREATE.
UPDATEMODE	INTEGER	Non	Indique le mode de mise à jour de l'index de recherche de texte. L'entier 0 (zéro) indique la mise à jour initiale de l'index de recherche de texte. La valeur 1 indique les mises à jour incrémentielles suivantes.
REORGANIZATIONMODE	INTEGER	Non	Indique le mode de réorganisation de l'index de recherche de texte.
CREATETIME	TIMESTAMP	Non	Heure de création de l'index de recherche de texte
LASTUPDATETIME	TIMESTAMP	Oui	Heure de la dernière mise à jour de l'index de recherche de texte
LASTUPDATESTATUS	CHAR	Oui	Indique le statut interne pour l'optimisation du processus de nettoyage après une mise à jour incrémentielle ou initiale de l'index de recherche de texte. Les valeurs habituelles sont les suivantes : <ul style="list-style-type: none"> • (Null) : indique que l'index n'a jamais été mis à jour. • 'C' : indique qu'une mise à jour initiale a été lancée. Si le mode de mise à jour n'est pas incrémentiel, la mise à jour initiale est encore en cours d'exécution ou a échoué. • 'N' : indique qu'une mise à jour incrémentielle a été exécutée. D'autres codes sont utilisés en interne au cours du processus de mise à jour. Le processus de mise à jour utilise ces codes pour déterminer les actions de reprise spécifiques qui doivent être exécutées si la mise à jour échoue.
SCHEDULERTASKID	INTEGER	Oui	Réservé pour une utilisation ultérieure
EXPRESSIONLISTS	CLOB (32 K)	Oui	Réservé pour une utilisation ultérieure
EXPRESSIONNUMBERS	VARBINARY(32)	Oui	Réservé pour une utilisation ultérieure
USEREXITFUNCTION	VARCHAR(18)	Oui	Réservé pour une utilisation ultérieure

Tableau 32. Contenu de la table d'administration QSYS2.SYSTEXTINDEXES (suite)

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
REMARKS	VARCHAR(2000)	Oui	Remarques saisies dans l'option COMMENTS du paramètre options_configuration_index de la procédure stockée SYSPROC.SYSTS_CREATE.
TABLEMBR	VARCHAR(10)	Oui	Membre de la table sur lequel l'index de texte est construit. Ce paramètre permet de suivre le membre indexé si le fichier contient plusieurs membres. Si la valeur est nulle, le membre est le seul membre de la table.

Concepts associés

«Nom d'alias de serveur», à la page 11

Vous pouvez utiliser un nom d'alias de serveur pour affecter un nom significatif à un serveur.

Table d'administration QSYS2.SYSTEXTCOLUMNS

La table d'administration QSYS2.SYSTEXTCOLUMNS vous permet de visualiser des informations sur les colonnes de texte d'un index de recherche de texte. Chaque index de recherche de texte contient un ID d'index, des noms de colonne de texte et le nom de schéma de la table de base.

Le tableau suivant répertorie le contenu de la table d'administration QSYS2.SYSTEXTCOLUMNS. La clé primaire de cette table est la colonne INDEXID associée à la colonne COLUMNNAME. La clé étrangère est la colonne INDEXID.

Tableau 33. Contenu de la table d'administration QSYS2.SYSTEXTCOLUMNS

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
INDEXID	INTEGER	Non	ID d'index unique généré pour l'index de recherche de texte
COLUMNNAME	VARCHAR(128)	Non	Nom non qualifié de la colonne de texte
TABLESCHEMA	VARCHAR(128)	Non	Nom de schéma de la table de base
TABLERNAME	VARCHAR(128)	Non	Nom non qualifié de la table de base
LANGUAGE	VARCHAR(5)	Non	Langue utilisée par le serveur de recherche de texte pour le traitement linguistique des documents texte. La valeur par défaut est en_US (anglais).
FUNCTIONSCHEMA	VARCHAR(128)	Oui	Schéma d'une fonction définie par l'utilisateur utilisée par OMNIFIND pour accéder aux documents texte qui résident dans une colonne d'un type de données non pris en charge ou qui sont stockés ailleurs.
FUNCTIONNAME	VARCHAR(18)	Oui	Nom d'une fonction définie par l'utilisateur utilisée par OMNIFIND pour accéder aux documents texte qui résident dans une colonne d'un type de données non pris en charge ou qui sont stockés ailleurs.

Tableau 33. Contenu de la table d'administration QSYS2.SYSTEXTCOLUMNS (suite)

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
CCSID	INTEGER	Non	ID du jeu de caractères codés utilisé pour un index de recherche de texte sur une colonne contenant des données de type binaire.
FORMAT	VARCHAR(30)	Non	Format des documents texte contenus dans la colonne. Les formats pris en charge sont TEXT, HTML, XML et INSO.
KEYCOLUMNCOUNT	INTEGER	Non	Nombre de colonnes clé associées à l'index de recherche de texte.
KEYCOLUMNNAMES	VARCHAR(1200)	Non	Noms des colonnes clé associées à l'index de recherche de texte.

Table d'administration QSYS2.SYSTEXTSERVERS

La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.

Le tableau suivant répertorie le contenu de la table d'administration QSYS2.SYSTEXTSERVERS. La clé unique de cette table est la colonne SERVERNAME associée à la colonne SERVERPORT. La clé primaire est la colonne SERVERID.

Tableau 34. Contenu de la table d'administration QSYS2.SYSTEXTSERVERS

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
SERVERID	INTEGER	Non	ID unique généré pour le serveur de recherche de texte
SERVERNAME	VARCHAR(128)	Non	Nom de système hôte ou adresse IP du serveur de recherche de texte
SERVERADRINFO	VARBINARY(3000)	Oui	Représentation interne de la valeur des paramètres SERVERNAME et SERVERPORT déterminée par la procédure stockée SYSPROC.SYSTS_START
SERVERPORT	INTEGER	Non	Numéro de port du serveur de recherche de texte
SERVERPATH	VARCHAR(512)	Non	Chemin de serveur associé au serveur de recherche de texte
SERVERTYPE	INTEGER	Non	Type de serveur du serveur de recherche de texte La valeur 0 (zéro) indique un serveur de recherche de texte IBM i. La valeur 1 indique un serveur de recherche de texte Linux. La valeur 2 indique un serveur de recherche de texte Windows.
SERVERAUTHTOKEN	VARCHAR(256)	Non	Jeton d'authentification du serveur de recherche de texte.
SERVERMASTERKEY	VARCHAR(36)	Non	Clé de serveur associée au serveur de recherche de texte

Tableau 34. Contenu de la table d'administration QSYS2.SYSTEXTSERVERS (suite)

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
SERVERCLASS	INTEGER	Non	Classe de serveur du serveur de recherche de texte La valeur 0 (zéro) indique un serveur de production disponible pour la sélection automatique. La valeur 9 indique un serveur de test, qui n'est jamais attribué automatiquement.
SERVERSTATUS	INTEGER	Non	Indique si le serveur peut être utilisé comme serveur de recherche de texte pour créer des index de recherche de texte. La valeur par défaut est 0 (zéro). Elle indique que le serveur peut être utilisé à cette fin.
ALIASNAME	VARCHAR(128)	Oui	Le nom d'alias est unique, s'il n'est pas nul, et distingue les majuscules des minuscules.

Concepts associés

«Nom d'alias de serveur», à la page 11

Vous pouvez utiliser un nom d'alias de serveur pour affecter un nom significatif à un serveur.

Table d'administration QSYS2.SYSTEXTCONFIGURATION

Le table d'administration QSYS2.SYSTEXTCONFIGURATION contient les paramètre de configuration de l'index de recherche de texte communiqués par la procédure stockée SYSPROC.SYSTS_CREATE.

Le tableau suivant répertorie le contenu de la table d'administration QSYS2.SYSTEXTCONFIGURATION. La clé primaire de cette table est la colonne INDEXID associée à la colonne PARAMETER. La clé étrangère est la colonne INDEXID.

Tableau 35. Contenu de la table d'administration QSYS2.SYSTEXTCONFIGURATION

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
INDEXID	INTEGER	Non	ID d'index unique généré pour l'index de recherche de texte
PARAMETER	VARCHAR(30)	Non	Paramètres spécifiés pour l'index de recherche de texte dans la procédure stockée SYSPROC.SYSTS_CREATE.
VALUE	VARCHAR(512)	Non	Valeurs des paramètres spécifiés.

Référence associée

«SYSPROC.SYSTS_CREATE», à la page 19

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_CREATE pour créer un index de recherche de texte. Cette procédure stockée active une colonne de texte pour l'indexation de recherche de texte. Vous pouvez ensuite utiliser l'index de recherche de texte dans des requêtes SQL contenant les fonctions CONTAINS ou SCORE.

Table d'administration QSYS2.SYSTEXTSERVERHISTORY

Pour consulter l'historique des serveurs utilisés pour la procédure stockée SYSPROC.SYSTS_DROP, vous pouvez afficher la table auxiliaire QSYS2.SYSTEXTSERVERHISTORY.

SQLCODE -0196

Le tableau suivant illustre le contenu de la table d'administration QSYS2.SYSTEXTSERVERHISTORY. La clé unique de cette table est la colonne INDEXID associée à la colonne SERVERID. La clé étrangère est la colonne INDEXID.

Tableau 36. Contenu de la table d'administration QSYS2.SYSTEXTSERVERHISTORY

Nom de colonne	Type de données	Accepte une valeur nulle ?	Description
INDEXID	INTEGER	Non	ID d'index généré pour un index de recherche de texte
SERVERID	INTEGER	Non	ID du serveur sur lequel l'index de recherche de texte doit être supprimé sur SYSPROC.SYSTS_DROP.

Référence associée

«SYSPROC.SYSTS_DROP», à la page 33

Vous pouvez appeler la procédure stockée SYSPROC.SYSTS_DROP pour supprimer un index de recherche de texte créé avec la procédure stockée SYSPROC.SYSTS_CREATE.

Messages et codes

Cette rubrique décrit les messages et les codes retour SQL du produit OmniFind Text Search for DB2 for i. Les messages sont répertoriés par ordre numérique.

Messages d'OmniFind

Cette rubrique décrit les messages d'OmniFind pour OmniFind Text Search for DB2 for i. Les messages sont répertoriés par ordre numérique.

Des messages sont ajoutés au fichier de messages d'OmniFind (QDBTSLIB/QOMFMSGF) pour les erreurs suivantes :

Tableau 37. Messages d'OmniFind

Numéro	Type	Message
OMF0011	Informations	Index de recherche de texte restaurés avec des option de configuration différentes.
OMF0012	Avertissement	Le type FORMAT de l'index en cours de création n'est pas XML. Les recherches en XML ne sont pas prises en charge.
OMF0334	Erreur	L'objet "{0}"/"{1}" que vous avez spécifié n'est pas pris en charge. Un index de recherche de texte ne peut être construit que sur une table SQL, un alias SQL ou un membre unique d'un fichier physique.
OMF0358	Erreur	L'utilisateur actif n'a pas les autorisations requises pour effectuer l'opération demandée.
OMF0359	Erreur	La restauration a échoué. Le serveur de texte associé à cet index de texte est indisponible.

SQLCODE -0196 Impossible de supprimer la colonne &3 dans &1 dans &2.

Explication : Le programme a tenté de supprimer la colonne &3. Il est impossible de supprimer cette colonne car une vue, une contrainte, un déclencheur ou un index en dépend et le paramètre RESTRICT a été spécifié, ou la colonne fait partie de la clé de partition.

Action de l'utilisateur : Indiquez le paramètre CASCADE dans l'instruction ALTER TABLE pour

supprimer la colonne ainsi que les vues, les contraintes, les déclencheurs et les index qui en dépendent. Si la colonne fait partie de la clé de partition, indiquez le paramètre DROP PARTITIONING dans l'instruction ALTER TABLE afin de supprimer le partitionnement pour cette table. Renouvelez la requête.

SQLSTATE: 42817

SQLCODE -5003 Impossible d'exécuter cette opération quand le contrôle de validation est actif.

Explication : Quand le contrôle de validation est actif, il est impossible d'exécuter les opérations suivantes avec les paramètres COMMIT(*CHG), COMMIT(*CS) ou COMMIT(*ALL) :

- Instruction DROP SCHEMA
- Instruction GRANT ou REVOKE sur un objet auquel l'accès est réservé à un dépositaire de droits
- Instruction CREATE en mode de désignation SQL pour un objet auquel l'accès est réservé à un dépositaire de droits
- Instruction DROP sur un index de recherche de texte

Ces opérations ne peuvent être ni validées ni annulées.

Action de l'utilisateur : Indiquez COMMIT(*NONE) puis entrez à nouveau l'instruction.

SQLSTATE : 42922

SQLCODE -20423 Une erreur s'est produite pendant le traitement de la recherche de texte.

Explication : Une erreur s'est produite pendant le traitement d'une fonction de recherche de texte CONTAINS ou SCORE. L'erreur a eu lieu sur le serveur *nom_serveur* avec l'index de recherche de texte *nom_index*. Code raison : *code_raison*. Description du problème : *description*.

nom_serveur : Nom de système hôte ou adresse IP et port du serveur de recherche de texte sur lequel l'erreur s'est produite.

nom_index : Nom de l'index utilisé dans le processus de recherche de texte.

Remarque : Cette valeur inclut le schéma et un point avec le nom d'index dans un même jeton.

code_raison : Code raison renvoyé par OmniFind Text Search Server for DB2 for i.

description : Texte de description renvoyé par OmniFind Text Search Server for DB2 for i.

Réaction du système : Le traitement de l'instruction est impossible.

Action de l'utilisateur : Contactez votre administrateur système pour vérifier l'installation du produit OmniFind Text Search Server for DB2 for i.

SQLSTATE: 38H10

Référence associée

«CONTAINS», à la page 38

Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.

«SCORE», à la page 41

Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

SQLCODE -20424 La recherche de texte n'est pas prise en charge pour la raison suivante :*code_raison*.

Explication : Un problème a été détecté dans l'une des tables d'administration de la recherche de texte. Code raison : *code_raison*.

1 L'une des tables d'administration de recherche de texte est introuvable (QSYS2.SYSTEXTINDEXES, QSYS2.SYSTEXTCOLUMNS ou QSYS2.SYSTEXTSERVERS).

3 Le module de prise en charge de la recherche de texte n'est pas installé.

4 La colonne STATUS dans la table QSYS2.SYSTEXTSERVERS contient la valeur 1, ce qui indique que la recherche de texte n'est plus prise en charge.

7 Aucun OmniFind Text Search Server for DB2 for i n'a été défini.

Réaction du système : Le traitement de l'instruction est impossible.

Action de l'utilisateur : Contactez votre administrateur système pour vous assurer que la prise en charge de la recherche de texte a été configurée sur votre système.

SQLSTATE: 38H11

Référence associée

«Table d'administration QSYS2.SYSTEXTSERVERS», à la page 106
 La table d'administration QSYS2.SYSTEXTSERVERS contient des informations sur les serveurs de recherche de texte installés.
 «Table d'administration QSYS2.SYSTEXTINDEXES», à la page 103
 La table d'administration QSYS2.SYSTEXTINDEXES vous permet de visualiser des informations sur chaque index de recherche de texte. Chaque index de recherche de texte possède un nom, un nom de schéma et un nom de collection associée sur le serveur de recherche de texte.
 «Table d'administration QSYS2.SYSTEXTCOLUMNS», à la page 105
 La table d'administration QSYS2.SYSTEXTCOLUMNS vous permet de visualiser des informations sur les colonnes de texte d'un index de recherche de texte. Chaque index de recherche de texte contient un ID d'index, des noms de colonne de texte et le nom de schéma de la table de base.

SQLCODE -20425 La recherche de texte est interdite pour la colonne *nom_colonne*.

Explication : Une fonction de recherche de texte CONTAINS ou SCORE a spécifié la colonne *nom_colonne* dans la table *nom_table* dans schéma *table*. Le traitement de la recherche de texte est impossible car il n'existe aucun index de texte pour cette colonne.

Réaction du système : Le traitement de l'instruction est impossible.

Action de l'utilisateur : Vérifiez que la colonne et la table sont enregistrées dans OmniFind Text Search Server for DB2 for i.

SQLSTATE: 38H12

Référence associée

«CONTAINS», à la page 38
 Vous pouvez utiliser la fonction CONTAINS pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie un résultat qui indique si une occurrence a été trouvée.
 «SCORE», à la page 41
 Vous pouvez utiliser la fonction SCORE pour rechercher un index de recherche de texte à l'aide de critères que vous indiquez dans un argument de recherche. La fonction renvoie une note de pertinence qui mesure à quel degré un document correspond à la requête.

SQLCODE -20426 La procédure administrative de recherche de texte est déjà en cours d'exécution.

Explication : Une procédure administrative de recherche de texte, par exemple une mise à jour, est

déjà en cours d'exécution sur cet index.

Réaction du système : Le traitement de l'instruction est impossible.

Action de l'utilisateur : Appelez à nouveau la procédure stockée d'administration lorsque la procédure stockée en cours d'exécution sera terminée.

SQLSTATE : 38H13

SQLCODE -20427 Une erreur s'est produite pendant la procédure administrative de recherche de texte.

Explication : Une erreur s'est produite pendant une procédure administrative de recherche de texte. Code raison : *code_raison*. Description du problème : *description*. Le texte d'erreur décrit le problème.

Réaction du système : L'instruction CALL échoue avec le code SQL suivant.

Action de l'utilisateur : Corrigez le problème décrit par le *message d'erreur* puis appelez à nouveau la procédure stockée administrative.

SQLSTATE : 38H14

CPF32fa Opération non autorisée sur l'index de recherche de texte &2 dans &1.

Explication : Une opération non prise en charge a été tentée sur un index de recherche de texte. Les index de recherche de texte n'autorisent pas certaines opérations qui sont admises pour les vues et index traditionnels de DB2.

Si vous avez tenté de supprimer un index, l'opération a échoué car le contrôle de validation est actif.

Action de l'utilisateur : Exécutez les opérations d'administration de recherche de texte à l'aide des procédures stockées SQL d'administration incluses dans OmniFind Text Search Server for DB2 for i.

Pour plus d'informations sur les index de recherche de texte et sur les restrictions applicables, reportez-vous au centre de documentation à l'adresse <http://www.ibm.com/systems/i/infocenter/>

CPF32fb L'opération sur l'index de recherche de texte &2 dans &1 a échoué.

Explication : Une opération a été tentée sur l'index de recherche de texte &2 dans &1. L'opération demandée n'est pas autorisée actuellement. Code raison &3. Les codes raison et leurs significations sont les suivants :

1. Un produit obligatoire n'est pas installé.
2. Le serveur de recherche de texte demandé &4 est indisponible ou non défini.
3. Une restauration de l'index a été tenté alors que cet index existe déjà. L'index existant n'a pas été modifié pour refléter l'index enregistré.

4. Un objet dépendant &5 dans &6, type &7, n'existe pas.
5. Un objet &5 dans &6, type &7, est indisponible.
6. Le pool de mémoire secondaire (ASP) de l'unité d'exécution courante ne correspond pas au pool de mémoire secondaire de l'index de recherche de texte.
7. Un index de recherche de texte existe déjà pour la colonne &8, table &5, dans la bibliothèque &6.

Action de l'utilisateur : Corrigez le problème et recommencez l'opération.

Pour plus d'informations sur les index de recherche de texte, reportez-vous au centre de documentation à l'adresse <http://www.ibm.com/systems/i/infocenter/>

CPI321E Fichier &1 différé dans la bibliothèque &2.

Explication : Le fichier &1 dans la bibliothèque &2 a été différé pendant la requête de restauration avec le DFRID spécifié &4. Code raison : &3. Les codes raison sont les suivants :

1. Le fichier de base &5 dans la bibliothèque &6 n'existait pas quand &1 a été créé pour la restauration.
2. La création du fichier &1 a échoué pour une raison autre que l'absence du fichier de base.
Reportez-vous aux messages précédents pour déterminer pourquoi la création du fichier a échoué.

3. La création d'un ou plusieurs membres a échoué pour le fichier &1. Reportez-vous aux messages précédents pour déterminer pourquoi la création des membres a échoué.
4. Le fichier représente un index de recherche de texte et les objets de programme sous licence requis n'existent pas sur le système.
5. Le fichier représente un index de recherche de texte et la recréation de l'index a échoué.

Action de l'utilisateur : Code raison 1 : Restaurez le fichier de base manquant ou utilisez la commande Restore Deferred Objects (RSTDFROBJ) en spécifiant le DFRID &4 dans l'une ou l'autre des commandes utilisées.

Codes raison 2 et 3 : Corrigez les problèmes ayant causé l'échec de la création puis utilisez la commande Restore Deferred Objects (RSTDFROBJ) en spécifiant le DFRID &4 dans la commande.

Code raison 5 : Reportez-vous aux messages précédents dans le journal de travail, corrigez les erreurs puis utilisez la commande Restore Deferred Objects (RSTDFROBJ) en spécifiant le DFRID &4 dans la commande.

Information associée

RSTDFROBJ (Restaurer objets différés)

Licence du code et informations de limitation de responsabilité

IBM vous concède une licence non exclusive de droits d'auteur vous autorisant à utiliser tous les exemples de code de programmation à partir desquels vous pouvez générer des fonctions similaires adaptées à vos besoins spécifiques.

SOUS RESERVE DE TOUTE GARANTIE LEGALE QUI NE PEUT ETRE EXCLUE, IBM, SES DEVELOPPEURS ET SES FOURNISSEURS NE FOURNISSENT AUCUNE GARANTIE EXPLICITE OU IMPLICITE, Y COMPRIS, ET DE FACON NON LIMITATIVE, TOUTE GARANTIE IMPLICITE D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE, ET TOUTE GARANTIE EN NON-CONTREFACON CONCERNANT LE LOGICIEL OU LE SUPPORT TECHNIQUE, LE CAS ECHEANT.

IBM, SES DEVELOPPEURS OU FOURNISSEURS NE PEUVENT EN AUCUN CAS ETRE TENUS RESPONSABLES DES DOMMAGES SUIVANTS, ET CE, MEME S'ILS ONT ETE INFORMES DE LEUR POSSIBLE SURVENANCE :

1. PERTE OU DETERIORATION DE VOS DONNEES ;
2. PREJUDICES MORAUX, ACCESSOIRES, DIRECTS OU INDIRECTS ; OU
3. PERTE DE BENEFICE, D'ACTIVITE COMMERCIALE, DE REVENU, DE CLIENTELE, OU D'ECONOMIES ESCOMPTEES.

CERTAINES LEGISLATIONS N'AUTORISENT PAS LA LIMITATION OU L'EXCLUSION DE PREJUDICES ACCESSOIRES, DIRECTS OU INDIRECTS, AUQUEL CAS CERTAINES DE CES EXCLUSIONS OU LIMITATIONS QUI PRECEDENT NE VOUS SERONT PAS APPLICABLES.

Annexe. Remarques

Ce document a été créé pour des produits et des services proposés aux Etats-Unis.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non-IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LES INFORMATIONS SONT LIVREES EN «L'ETAT» SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Livret Contractuel IBM, des Conditions Internationales d'Utilisation de Logiciels IBM, des Conditions d'Utilisation du Code Machine ou de tout autre contrat équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Ce document contient des exemples de programme d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les modèles de programmes sont fournis "en l'état", sans garantie d'aucune sorte. IBM ne sera en aucun cas responsable des dommages liés à l'utilisation des modèles de programmes.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (nom de votre société (année)). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. _indiquez l'année ou les années_.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Documentation sur l'interface de programmation

La présente publication pour OmniFind Text Search Server for DB2 for i décrit des interfaces de programmation qui permettent au client d'écrire des programmes pouvant utiliser les services d'OmniFind Text Search Server for DB2 for i.

Marques

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines Corp. aux Etats-Unis et/ou dans certains autres pays. Les autres noms de produits et de services peuvent appartenir à IBM ou à des tiers. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web Copyright and trademark information à www.ibm.com/legal/copytrade.shtml.

Adobe, le logo Adobe, PostScript et le logo PostScript sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Windows est une marque de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Dispositions

Les droits d'utilisation relatifs à ces publications sont soumis aux dispositions suivantes.

Usage personnel : Vous pouvez reproduire ces publications pour votre usage personnel, non commercial, sous réserve que toutes les mentions de propriété soient conservées. Vous ne pouvez distribuer ou publier tout ou partie de ces informations ou en faire des oeuvres dérivées sans le consentement exprès d'IBM.

Usage commercial : Vous pouvez reproduire, distribuer et afficher ces publications uniquement au sein de votre entreprise, sous réserve que toutes les mentions de propriété soient conservées. Vous ne pouvez reproduire, distribuer, afficher ou publier tout ou partie de ces publications en dehors de votre entreprise, ou en faire des oeuvres dérivées, sans le consentement exprès d'IBM.

Excepté les droits d'utilisation expressément accordés dans ce document, aucun autre droit, licence ou autorisation, implicite ou explicite, n'est accordé pour ces publications ou autres informations, données, logiciels ou droits de propriété intellectuelle contenus dans ces publications.

IBM se réserve le droit de retirer les autorisations accordées ici si, à sa discrétion, l'utilisation des publications s'avère préjudiciable à ses intérêts ou que, selon son appréciation, les instructions susmentionnées n'ont pas été respectées.

Vous ne pouvez télécharger, exporter ou réexporter ces informations qu'en total accord avec toutes les lois et règlements applicables dans votre pays, y compris les lois et règlements américains relatifs à l'exportation.

IBM NE DONNE AUCUNE GARANTIE SUR LE CONTENU DE CES PUBLICATIONS. LES PUBLICATIONS SONT LIVREES EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES PUBLICATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

