Ch. 6. Tables texte

De Apache OpenOffice Wiki
< FR | Documentation | HSQLDB Guide

Sommaire

Chapitre 6. Tables texte

Le document original, à valeur normative, est à l'adresse : http://hsqldb.org/doc/guide/ch06.html

Hsqldb supporte en standard les tables texte

Bob Preston

Groupe de développement HSQLDB

Fred Toussi

HSQLDB Development Group

<ft@cluedup.com>

$Date: 2007/08/28 13:19:09 $

La prise en charge des tables texte pour HSQLDB a été originellement développée par Bob Preston indépendamment du projet. C'est quand il

rejoignit le projet qu'il incorpora cette fonctionnalité à partir de la version 1.7.0, avec de nombreuses améliorations, dont notamment l'usage de commandes SQL conventionnelles pour spécifier les fichiers utilisés comme tables texte.

En quelques mots, les tables texte sont des fichiers .csv ou autres fichiers délimités traités comme des tables SQL. N'importe quel fichier csv ordinaire ou autre fichier délimité peut être utilisé. La pleine puissance des requêtes SQL peut être executée sur ces fichiers, dont les commandes SELECT, INSERT, UPDATE et DELETE. Les index et index uniques peuvent être établis, et les contraintes de clefs externes peuvent être utilisées pour renforcer l'intégrité référentielle entre les tables texte elles-mêmes ou même avec des tables conventionnelles.

HSQLDB avec la prise en charge des tables texte est la seule solution conséquente (comprehensive) combinant la puissance de SQL et l'interface universelle de JDBC pour manipuler les données enregistrées dans des fichiers texte. Elle aura un vaste champ d'application au delà du domaine Java de HSQLDB déjà établi.

Buts de l'implémentation :

1. Nous projetons de finaliser les DDL (Data Definition Language: Langage de définition des données) pour les tables texte afin que les distributions futures de HSQLDB utilisent les mêmes scripts DDL.
2. Nous projetons de prendre en charge les tables texte comme des tables GLOBAL TEMPORARY ou GLOBAL BASE dans le domaine SQL.

L'implémentation

Définition des tables

Les tables texte sont définies comme les tables conventionnelles avec le mot-clé TEXT:

```
CREATE TEXT TABLE <tablename> (<column definition> [<constraint definition>])
```

De plus, une commande SET définit le fichier et le caractère de séparation que la table texte utilise:

```
SET TABLE <tablename> SOURCE <quoted_filename_and_options> [DESC]
```

Les tables texte ne peuvent pas être créées dans des bases en mémoire seule: (bases n'ayant pas de fichier de script).

Champs d'applications et réassignations

(Scope and Reassignment)

- Une table texte qui n’est assignée à aucun fichier est en lecture seule (READ ONLY) et vide (EMPTY).
- Une table texte temporaire a la portée et la durée de vie d’une session SQL (d’une connexion JDBC).
- Réassigner la définition d’une table texte dans un nouveau fichier a des implications dans les domaines suivants:

1- L’utilisateur doit avoir les droits de l’administrateur.

2- Les transactions existantes sont effectuées dès lors.

3- Les contraintes, incluants les clefs externes qui font référence à cette table, sont gardées intactes. C’est la responsabilité de l’administrateur que d’assurer cette intégrité.

Depuis la version 1.7.2 le nouveau fichier source est parcouru et les index sont construits lors de l’assignation à la table. Dès lors chaque violation d'une des contraintes NOT NULL, UNIQUE ou PRIMARY KEY entraîne l'annulation de l'assignation. Toutefois, les contraintes de clés externes ne sont pas vérifiées au moment de l'assignation ou de la réassignation du fichier source.

Valeurs nulles dans les colonnes de tables texte

Il y a eu du changement depuis la version 1.7.2 pour prendre en charge et les valeurs nulles et les chaînes de caractères vides.

- Les champs vides sont gérés comme NULL. Ce sont des champs ou il n’y a soit rien, soit des espaces entre les séparateurs.
- Les chaines de caractères vides encadrées d’apostrophes sont traitées comme des chaines vides.

Configuration

Le séparateur de champ par défaut est la virgule (,). Un séparateur de champ différent peut être défini dans la déclaration SET TABLE SOURCE. Par exemple, pour changer le séparateur de champ d’une table MaTable en une barre verticale, insérez la ligne suivante dans la déclaration SET TABLE SOURCE, par exemple:

```
SET TABLE MaTable SOURCE "MonFichier;fs=|"
```

HSQLDB traitant les CHAR’s, VARCHAR’s et LONGVARCHAR’s de la même manière, on peut assigner des séparateurs différents aux deux derniers.Quand un autre séparateur est attribué à un champ VARCHAR ou LONGVARCHAR, il

termine tous les champs CSV de ce type. Par exemple, si le premier champ est de type CHAR et le second de type LONGVARCHAR, le séparateur fs est défini comme le pipe (|) et le vs comme le point (.), alors les données pour une ligne dans le fichier CSV ressemblent à ceci:

```
1er champ de données|2e champ de données.3e champ de données
```

L'exemple suivant montre comment changer le séparateur par défaut en pipe (|), le séparateur de VARCHAR en point (.) et le séparateur LONGVARCHAR en tilde (~). Insérez la ligne suivante dans la déclaration SET TABLE SOURCE, par exemple:

```
SET TABLE MaTable SOURCE "MonFichier;fs=|;vs=.;lvs=~"
```

HSQLDB reconnait aussi les indicateurs spéciaux suivants pour les séparateurs :

Indicateurs spéciaux pour les séparateurs

\semi	Point-virgule (semicolon)
\quote	Guillemets (quote)
\space	Caractère Espace
\apos	apostrophe
\n	Nouvelle ligne – Utilisé comme une ancre de fin (comme $ dans les expressions régulières) newline - Used as an end anchor (like $ in regular expressions)
\r	Retour chariot
\t	tab
\\	backslash
\u####	Un caractère unicode exprimé en hexadécimal

De plus, HSQLDB fournit la prise en charge de fichiers csv avec trois options booléennes supplémentaires: ignore_first, quoted et all_quoted. L'option ignore_first (faux par défaut) indique à HSQLDB d'ignorer la première ligne d'un fichier. Cette option est utilisée quand la première ligne contient les en-têtes de colonnes. L'option all_quoted (faux par défaut) indique au programme qu'il doit encadrer chaque champ d'apostrophes (') à l'écriture du fichier source. L'option quoted (vrai par défaut) utilise les apostrophes seulement quand nécessaire pour distinguer un champ contenant le caractère séparateur. Il peut être réglé sur faux pour interdire l'usage de tout apostropher et traiter les caractères apostrophes comme des caractères normaux. Ces options doivent être spécifiées dans la déclaration SET TABLE SOURCE

```
SET TABLE MaTable SOURCE "MonFichier;ignore_first=true;all_quoted=true"
```

Lorsque les options par défaut all_quoted= false et quoted=true sont établies, les champs écrits dans une ligne du fichier csv seront apostrophés seulement s'ils contiennent le séparateur ou le caractère apostrophe. Ce dernier est doublé s'il est utilisé dans une chaine de caractères. Si all_quoted=false et quoted=false le caractère apostrophe n'est pas doublé. Avec cette option il n'est pas possible d'insérer une chaine contenant le séparateur, puisqu'il deviendrait impossible à distinguer d'un séparateur véritable. Lors de la lecture d'un fichier source existant, le programme traite chaque champ séparément. Il détermine si un champ est apostrophé seulement si le premier caractère est l'apostrophe. Ensuite il interprète le reste du champ sur ce postulat.

Le codage des caractères du fichier source est ASCII par défaut. Pour pouvoir supporter UNICODE ou des fichiers source préparés avec différents encodages ce paramètre peut être changé en UTF-8 ou n'importe quel autre encodage. Le choix par défaut est encoding=ASCII et l'option encoding=UTF-8 ou tout autre encodage pris en charge peut être utilisé.

Pour conclure, HSQLDB offre la possibilité de lire un fichier texte de la fin vers le début et de le rendre en lecture seule, en placant le mot-clé "DESC" à la fin de la déclaration SET TABLE SOURCE :

```
SET TABLE MaTable SOURCE "MonFichier" DESC
```

Cette option fournit une fonctionnalité similaire à la queue de commande Unix, par relecture du fichier à chaque fois qu'une commande select est exécutée. L'utilisation de cette fonctionnalité règle la table en mode lecture seule. Ensuite, il ne sera plus possible de changer le statut lecture seule avec `SET TABLE <NomTable> READONLY TRUE .`

Les fichiers sources des tables texte sont chargés en mémoire. Le nombre maximum de lignes de données qui sont en mémoire à chaque instant est controllé par la propriété textdb.cache_scale .La valeur par défaut de textdb.cache_scale est 10 et peut être modifiée en règlant la propriété dans le fichier des propriétés de la base de données. Le nombre de lignes en mémoire est calculé comme 3*(2^échelle), ce qui se traduit en 3 072 lignes pour le réglage par défaut de textdb.cache_scale setting (10). Cette propriété peut aussi être définie individuellement pour chaque table texte:

```
SET TABLE MaTable SOURCE "MonFichier;ignore_first=true;all_quoted=true;cache_scale=12"
```

Déconnecter les tables texte

Ce qui suit décrit le comportement présent dans la version 1.8.0.8 et ultérieur.

Les tables texte doivent être déconnectées de leur source de données sous-jacente, par exemple le fichier texte.

Vous pouvez explicitement déconnecter une table texte de son fichier en émettant la déclaration suivante:

```
SET TABLE MaTable SOURCE OFF
```

Par le fait, MaTable sera vidée et en lecture seule. Toutefois, la description de la source de données sera préservée, et la table pourra être reconnectée grâce à cette description.

```
SET TABLE MaTable SOURCE ON
```

A l'ouverture d'une base de données, Si le fichier source d'une table texte existante manque la table reste déconnectée de sa source de données, mais la description de la source est préservée. Cela permet au fichier source manquant d'être ajouté au répertoire et à la table d'être reconnectée (au fichier source) avec la commande ci-dessus.

Problèmes liés aux fichiers texte

- Les localisations de fichiers sont restreintes aux sous-répertoires du dossier contenant la base de données, à moins que la propriété textdb.allow_full_path soit réglée sur vrai dans le fichier des propriétés de la base de données.
- Les lignes vierges sont permises n'importe ou dans le fichier texte, et sont ignorées.
- La localisation d'un fichier d'une table texte créé avec

```
  SELECT <liste de sélection> INTO TEXT <NomTable> FROM
```

est le dossier qui contient la base de données et le nom du fichier est basé sur le nom de la table. Le nom de la table est transformé en nom de fichier par remplacement de tous les caractères non-alphanumériques en caractère "underscore" ou de soulignement (_), conversion en minuscules, et ajout du suffixe ".csv"

- Il est possible de définir une clé primaire ou une colonne d'identité dans les tables texte.
- Quand un fichier d'une table source est utilisé avec l'option ignore_first=true , la première ligne, ignorée, est remplacée par une ligne vierge après une commande SHUTDOWN COMPACT.

- Le fichier d’une table source existante doit contenir des champs CHARACTER qui ne commencent pas par le caractère apostrophe (il peut cependant en exister des instances). Ces champs sont lus comme des chaines de caractères littérales. Ou alors, si un champ commence par le caractère apostrophe, il est alors interprété comme une chaine “apostrophée” qui doit finir par le caractère apostrophe. De plus chaque instance du caractère apostrophe se trouve alors doublé. Quand un champ contenant le caractère apostrophe ou le caractère séparateur est écrit par le programme vers le fichier source, le champ est encadré par des caractères apostrophes et chaque instance de l’apostrophe dans le champ est doublée.
- Vous pouvez insérer des sauts de lignes ou le caractère Retour chariot dans les chaines pour le type de champ CHARACTER. Cette fonction est inactive quand les deux propriétés quoted et all_quoted sont fausses.
- Les commandes ALTER TABLE qui ajoutent ou suppriment des colonnes ne sont pas prises en charge pour des tables texte non vides.

Propriétés globales des fichiers texte

Liste complète des propriétés globales prises en charge dans les fichiers *.properties

- textdb.fs
- textdb.lvs
- textdb.quoted
- textdb.all_quoted
- textdb.ignore_first
- textdb.encoding
- textdb.cache_scale
- textdb.allow_full_path

Importer un fichier Table texte dans / vers une table traditionnelle

Le dossier src/org/hsqldb/sample de votre distribution de HSQLDB contient un fichier nommé load_binding_lu.sql. C’est un fichier SQL fonctionnel qui importe un fichier texte délimité par le caractère pipe du dossier des fichiers de bases de données dans une table normale existante. Vous pouvez éditer une copie de ce fichier et l’utiliser directement avec **SqlTool**, ou vous pouvez utiliser le code SQL tel quel comme modèle (en utilisant n’importe quel client SQL)

Exemples commentés

- Liaison dynamique avec un fichier .csv (http://user.services.openoffice.org/fr/forum/ftopic11191.html)