

Bases de données, ENS Cachan & Ulm

TP n° 3 – Import de fichier, Ordres SQL avancés, JDBC

Pierre Senellart (pierre@senellart.com)

7 mars 2008

Le but de ce TP est d'expérimenter, sur une base de données réelle, quelques requêtes complexes, et de se familiariser avec l'interface Java d'accès à une base de données.

I Le standard Unicode

Unicode est un *répertoire de caractères* ayant pour vocation d'attribuer un code (entier, entre 0 et $17 \cdot 2^{16} - 1$) à chacun des caractères des systèmes d'écriture du monde entier. Ainsi, le code Unicode (en hexadécimal) de « é » est E9, tandis que celui de « Æ » est 5D0. Nous allons au cours de ce TP travailler avec le fichier `UnicodeData.txt` qui décrit certaines des propriétés des caractères Unicode. Ce fichier est disponible à l'URL `ftp://ftp.unicode.org/Public/UNIDATA/UnicodeData.txt` et son format est décrit dans le document disponible à l'URL `http://unicode.org/Public/UNIDATA/UCD.html`, auquel on se référera pour les questions suivantes.

1. Dans PHPMyAdmin, créer une table `Unicode` à 15 colonnes représentant le plus fidèlement possible la structure du fichier `UnicodeData.txt` ; on choisira avec soin le type de chacune des colonnes (se référer à la documentation en ligne de MySQL au besoin).
2. Utiliser la fonction d'importation de PHPMyAdmin (choisir l'option CSV avec `LOAD DATA`) pour importer le contenu du fichier `UnicodeData.txt` dans la base de données `Unicode`.
3. Contrôler le contenu de la table pour vérifier que l'importation s'est bien déroulée. En particulier :

- (a) Y a-t-il autant de lignes dans la base que de lignes dans le fichier `UnicodeData.txt` ? Le premier chiffre peut être obtenu avec une requête SQL avec fonction d'agrégation :

```
SELECT COUNT(*) FROM Unicode
```

Le deuxième chiffre peut-être obtenu avec l'outil de ligne de commande `wc`. En cas de problème, comprendre l'origine de celui-ci, corriger le schéma, vider la table et refaire l'importation.

- (b) Vérifier rapidement le contenu de la table. En cas de besoin, il peut être utile de faire quelques corrections locales sur la table avec un ordre SQL de *mise à jour* :

```
UPDATE Table SET Colonne=valeur WHERE Condition
```

4. Quel est le résultat d'un tri de la table selon le code Unicode ? Faire en sorte que les caractères apparaissent dans l'ordre naturel. Pour cela, deux possibilités : stocker les codes comme des chaînes de caractères complétées à gauche par des zéros (utiliser la fonction MySQL `LPAD`) ou les stocker comme codes décimaux (utiliser la fonction MySQL `CONV`). Modifier de la même façon les colonnes contenant le code du caractère en capitale, en bas-de-casse, et en casse de titre (respectivement, *uppercase*, *lowercase* et *titlecase*).

2 Requêtes

Pour cet exercice, on utilisera l'outil de ligne de commande `mysql`. Le but est de construire les requêtes SQL répondant à chacune des questions ci-dessous. Ne pas oublier de sauvegarder les requêtes dans un fichier pour en garder une trace !

1. Afficher les nom des caractères c_u , c_l tel que c_l est le bas-de-casse correspondant à c_u , mais c_u n'est pas la capitale correspondant à c_l . On aura besoin d'utiliser des alias de noms de tables :

```
SELECT t1.colonne1, t2.colonne2 FROM Table1 t1, Table2 t2 WHERE ...
```

2. Donner la liste des types de décomposition de compatibilité. On utilisera une requête `SELECT DISTINCT` qui fusionne les doublons, et on se référera à la documentation de MySQL sur les fonctions s'appliquant aux chaînes de caractères.

- Donner, pour chaque valeur numérique distincte, le nombre de caractères ayant cette valeur numérique. On utilisera une requête d'agrégation :

```
SELECT FonctionAgregation(colonne1), colonne2 FROM ... WHERE ... GROUP BY colonne2
```

- Créer une vue, du nom de UniqueCanonDecomposition, formée des codes et noms des couples de caractères (c, c') tels que c se décompose canoniquement en l'unique caractère c' . On utilisera la syntaxe :

```
CREATE VIEW ... AS SELECT ...
```

On peut renommer les colonnes d'une vue de la manière suivante :

```
CREATE VIEW Vue AS SELECT col1 nouveau_nom FROM Table
```

- Créer une vue, du nom de UniqueCompatDecomposition, formée des codes et noms des couples de caractères (c, c') et du type t tels que c a une décomposition de compatibilité de type t en l'unique caractère c' . On pourra utiliser l'opérateur d'expression régulière REGEXP.
- Créer une vue, du nom de UniqueDecomposition, formée de la réunion des deux vues précédentes. On utilisera l'opérateur UNION.
- Utiliser la vue précédente pour afficher tous les caractères ayant une liste de décomposition constituée de plus d'un caractère. On utilisera une requête imbriquée, de la forme :

```
SELECT ... FROM ... WHERE colonne NOT IN (SELECT ... )
```

3 JDBC

Pour la réalisation de programmes en ligne de commande ou avec interface graphique (par opposition à des applications Web) qui manipulent des données, on pourra utiliser Java en conjonction avec MySQL grâce à l'interface de programmation JDBC. Cette interface standardisée permet de communiquer avec divers systèmes de gestions de bases de données de manière uniforme.

Plusieurs variables d'environnement doivent être positionnés pour pouvoir utiliser Java et le connecteur JDBC de MySQL sur les machines de TP. Si votre shell de login est bash, vous pouvez mettre les lignes de commande suivantes dans votre fichier `.bash_profile`; pour un autre shell, adapter en conséquence.

```
export PATH=/import/senellar/jdk/bin:$PATH
export CLASSPATH=/import/senellar/jdbc/mysql-connector-java-5.0.8-bin.jar:
export JAVA_HOME=/import/senellar/jdk
```

L'utilisation de JDBC est illustré par l'exemple suivant (téléchargeable depuis la page Web des TP) :

```
import java.sql.*;

public class HelloMySQL
{
    public static void main(String args[]) {
        try {
            // Chargement du connecteur MySQL
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException ex) {
            System.err.println("Impossible de trouver le connecteur JDBC MySQL.");
            System.exit(1);
        }

        try {
            Connection cx=DriverManager.getConnection(
                "jdbc:mysql://dptserv01/base", "login", "mdp");

            Statement st=cx.createStatement();
            ResultSet rs=st.executeQuery("SELECT * FROM Unicode");
            while(rs.next()) {
                System.out.println(rs.getString("name"));
            }
        } catch(SQLException e) {
            System.err.println("Erreur MySQL: " + e.getMessage() + ".");
        }
    }
}
```

```
}  
}  
}
```

1. Compiler et exécuter l'exemple ci-dessus.
2. Écrire un programme Java qui, en utilisant la table Unicode, affiche des informations sur un caractère Unicode. Le caractère Unicode est fourni comme argument de ligne de commande, soit sous la forme d'un caractère, soit sous la forme d'un nombre en hexadécimal. Les informations à afficher sont :
 - (a) le code Unicode ;
 - (b) le caractère correspondant ;
 - (c) le nom du caractère ;
 - (d) la catégorie Unicode ;
 - (e) la décomposition, si elle existe ;
 - (f) la valeur numérique, si elle existe.

Par exemple :

```
0000E9 é LATIN SMALL LETTER E WITH ACUTE (Ll) 0065 0301  
002153 ⅓ VULGAR FRACTION ONE THIRD (No) <fraction> 0031 2044 0033 (=1/3)
```

Remarque : En fait, Java dispose déjà de fonctionnalités d'accès aux propriétés Unicode, voir notamment la classe `Character`, donc l'utilisation d'une base de données est ici un peu superflu.

4 Compléments

1. Certaines plages de caractères ayant tous les mêmes propriétés ne sont pas explicites dans `UnicodeData.txt`, mais sont délimitées par deux caractères abstraits `First` et `Last` dont le nom est entre chevrons (p. ex., `<CJK Ideograph Extension A, First>`). Écrire un programme Java qui remplace dans la base de données MySQL ces deux caractères virtuels par l'ensemble des caractères correspondants. Bien tester le comportement du programme avant de faire les modifications sur la base !
2. La manière dont les décompositions sont indiqués dans le fichier `UnicodeData.txt` n'est guère pratique pour les exploiter avec MySQL. Proposer une modification de schéma, plus adaptée au modèle relationnel, et écrire un programme Java qui effectue la transformation des données. Utiliser la nouvelle organisation des données pour créer une vue donnant, pour un caractère donné, l'ensemble des caractères qui l'utilisent dans leur décomposition.