Principes d'utilisation des systèmes de gestion de bases de données

JDBC (4)

L3, Informatique

Emmanuel Waller, LRI, Orsay

pbs MP : exécution d'ordres BD

- principe et déroulement
- ordres sans paramètre
- ordres avec paramètres
- appel procédure stockée
- SQL dynamique
- curseurs

pbs MP : curseurs

- rappel : curseur
- exemples
- détails

exemple 1

- Train(client varchar2(10), dest varchar2(10), jour integer)
- afficher la première destination de Cassavetes par ordre alphabétique et le jour du voyage
- On suppose qu'il y en a au moins une
- commençons intuitivement sur un exemple

```
Statement s = c.createStatement();
ResultSet rset =
   s.executeQuery(
    "select dest, jour from train"
    + "where client = 'Cassavetes' order by dest");
rset.next();
System.out.println("Cassavetes va à " + rset.getString(1)
                          + " le jour " + rset.getInt("JOUR"));
r.close();
s.close();
```

rappel : fonctionnement curseur

1. Déclaration du curseur
2. Remplissage en une seule fois par exécution de la requête
3. Récupération des lignes une par une

   (parcours séquentiel du curseur par un pointeur logique)
4. Libération de la zone : elle devient inaccessible

rappel : parcours du curseur

- Analogue à parcours de fichier séquentiel
- Notion de pointeur logique
- Après le remplissage : pointeur logique positionné avant la 1ère ligne
- une étape :
  1. Avance le pointeur
  2. Lit la ligne pointée par le pointeur
  3. Si pas de ligne pointée (l'avant-dernière étape avait lu la dernière) : indicateur de ligne trouvée passe à faux

```
Statement s = c.createStatement();
ResultSet rset =
   s.executeQuery(
    "select dest, jour from train"
    + "where client = 'Cassavetes' order by dest");
rset.next();
System.out.println("Cassavetes va à " + rset.getString(1)
                          + " le jour " + rset.getInt("JOUR"));
r.close();
s.close();
```

curseur JDBC

- sans paramètre ou avec
- déclaration et nommage :
  - pas vraiment : création « objet ordre »
  - Statement ou PreparedStatement selon si paramètres

- Remplissage : executeQuery
  - Renvoie objet de la classe ResultSet
  - = « la zone curseur », la table
- Avancement du pointeur et indicateur de fin de curseur
  - Méthode de la classe ResultSet
  - Boolean next() throws SQLException
  - True ssi nouvelle ligne trouvée

- Récupération de la ligne :
  - Colonne par colonne
  - ResultSet getXXX throws SQLException
    - Où XXX tout type primitif Java
  - Driver JDBC convertit donnée du curseur en type Java XXX
  - Colonne désignée au choix par :
    - Position :
      - Commence à 1
      - Plus efficace
    - Nom (majuscules)

- Exemple : String
  - String getString (int indiceColonne)
  - String getString (String nomColonne)
- Cf ci-dessus :
  - PreparedStatement.setXXX
  - CallableStatement.getXXX

- Libération :
  - void ResultSet close() throws SQLException
  - Sinon fait implicitement lors :
    - Fermeture (ou réexécution) de l'ordre qui l'a généré
    - GC

```
Statement s = c.createStatement();
ResultSet rset =
   s.executeQuery(
    "select dest, jour from train"
    + "where client = 'Cassavetes' order by dest");
rset.next();
System.out.println("Cassavetes va à " + rset.getString(1)
                          + " le jour " + rset.getInt("JOUR"));
r.close();
s.close();
```

Exemple 2

- afficher la première destination p ar ordre alphabétique de Cassavetes et le jour du voyage, et la deuxième si elle existe
- On suppose au moins une

```
Statement s = c.createStatement();
ResultSet rset =
   s.executeQuery(
    "select dest, jour from train"
    + "where client = 'Cassavetes' order by dest");
rset.next();
System.out.println("Cassavetes va à " + rset.getString(1)
                          + " le jour " + rset.getInt("JOUR"));
if (rset.next())
    System.out.println("Cassavetes va à " + rset.getString(1)
                          + " le jour " + rset.getInt("JOUR"));
r.close();
s.close();
```

Exemple 3

- Les afficher toutes
- On ne suppose rien
- s.executeQuery(. . .);

  while (rset.next())

  . . . getString(1) . . .
- Si aucun : on n'entre pas dans la boucle
- Si un ou plusieurs :
  - on se positionne sur la prochaine ligne, qu'on affiche
  - Après la dernière : on teste rset.next et on ne rentre pas dans la boucle