JAVA

JDBC : Traitement des requêtes SQL : L'interface Statement

Traitement des requêtes SQL

Pour traiter une requête SQL, on dispose de plusieurs objets capables d'envoyer celle- ci à la base de données :

- ***Statement*** : objet utilisé pour l'exécution d'une requête SQL statique retournant les résultats qu'elle produit.
- ***PreparedStatement*** : utilisé lorsqu'il est nécessaire d'exécuter une requête plusieurs fois, avec des paramètres différents.
- ***CallableStatement*** : objet utilisé pour appeler une procédure stockée. Des instances de ces objets sont disponibles grâce à l'instance de Connection.

L'interface Statement : les requêtes simples

L'objet Statement représente une instruction de traitement SQL. Il est créé par l'intermédiaire d'une instance de Connection par la méthode :

public Statement createStatement() throws SQLException;

Exemple :

```
try {
Statement state = connect.createStatement();
    }
 catch (SQLException e) { ... }
```

Méthodes permettant de faire exécuter

- Une requête SQL de consultation (SELECT), retournant les résultats de celle-ci :
  - public ResultSet executeQuery(String sql) throws SQLException;

- Une requête de modification (UPDATE, DELETE, INSERT, CREATE, DROP), ne renvoyant que le nombre d'occurrences affectées par celle-ci :
  - public int executeUpdate(String sql) throws SQLException;

- Un lot de requêtes, renvoyant un tableau d'entier correspondant au résultat de la requête :
  - public int[] executeBatch() throws SQLException;

- utilisé conjointement avec les méthodes :
  - public void addBatch(String sql) throws SQLException;
  - public void clearBatch() throws SQLException;

Exemples

```
try {
 ResultSet resultat = state.executeQuery( "SELECT
DISTINCT nom FROM eleves ORDER BY nom;");
     }
catch (SQLException e) { ... }
```

Ceci permet d'obtenir les noms des élèves de la table éleves, classés par ordre alphabétique

Exemples

```
try {
state.executeUpdate("CREATE TABLE vendeur" + " (NumVendeur integer" +
", Nom char(15)" +
 ", Prenom char(10)" +
", DateEmbauche date" +
", NumChef integer" +
 ", Salaire numeric(6, 0)" +
 ", Commission numeric(4, 1)" +
", ContratID logical" + ");");
}
catch (SQLException e) { ... }
```

Ceci permet de créer une table vendeur dans la base de données courante.

Exemples

```
try {
state.addBatch("DELETE FROM vendeur WHERE numvendeur =
'06897'");
state.addBatch("UPDATE vendeur SET sal = 1215 WHERE sal < 1215");
int[] results = state.executeBatch();
}
catch (SQLException e) { ... }
```

Ceci permet de supprimer l'occurrence du vendeur ayant le numéro 06897 et de fixer tous les salaires à 1215 minimum

Le tableau d'entier results contiendra par exemple [1,15], ce qui signifie que la première requête ("DELETE…") aura affecté une seule ligne, tandis que la seconde ("UPDATE…") en aura affecté 15.

Tester