

BD, Architecture Client-serveur & Accès JDBC

Guide sommaire d'utilisation de MySQL

MySQL est un SGBD basé sur SQL. MySQL propose essentiellement deux applications logicielles : le *serveur MySQL* d'une part, et le *client MySQL* d'autre part. Dans le cadre de ce travail nous nous intéressons plus particulièrement au client MySQL, nous l'utilisons pour interroger une base de données déjà existante.

Pour installer MySQL sur vos machines, vous pouvez vous connecter sur <http://www.mysql.fr/> et télécharger la version 5.*.* (par exemple la version 5.0.45) du moteur.

Télécharger ensuite dans votre bibliothèque le driver : `.;c:\BasesMysql\nom_du_connecteur.jar`;

Mise en route du serveur MySQL :

Pour commencer, vous pouvez lancer le serveur MySQL selon les mêmes procédures habituelles.

Mise en route du client MySQL :

Une fois le serveur lancé, vous vous y connectez en lançant le client MySQL. Pour vous connecter au serveur, il faut fournir un nom d'utilisateur, et dans la plupart des cas, un mot de passe.

Quelques commandes utiles :

Utiliser les supports des différents cours et TD applicatifs.

Travail demandé

1. Accès JDBC

On considère la base de données MabaseDeDonnees contenant les tables aux relations suivantes :

Coureur (numLicence, nom, prenom, dateNaissance date)

Course (numCourse, codePostal, ville);

Resultats (numCourse, numLicence, temps time, rang)

Les attributs **Coureur**, **Course** et **Resultats** sont définis sur le même domaine. Les clés des relations sont soulignées.

Les valeurs de ces tables sont les suivantes :

Coureur

20022911	Quinqueton	Joel	1952-11-29
20020511	Py	Bernard	1942-11-05
20022009	Boe	Jean Marie	1952-08-18
20022010	Durand	Sylvain	1967-06-30
20022011	Lochard	Eric Olivier	1952-06-15

Course

200201	34000	Montpellier
200202	78000	Versailles
200203	06000	Nice
200204	30000	Nimes
200205	66000	Perpignan
200206	64000	Biarritz

Resultat

200204	20022911	01:25:00	15
200202	20020511	01:20:00	6
200202	20022009	01:35:00	12
200203	20022010	01:35:00	8
200203	20022011	01:40:00	15
200203	20022911	01:55:00	20
200204	20020511	01:15:00	2

. Objectifs

- Réaliser dans un premier temps une connexion au serveur selon l'approche classique JDBC
- Utiliser quelques requêtes SQL via l'interface JDBC
- Créer une interface **ConnexionJDBC** qui déclare trois méthodes :
 - openConnection()** ouvrant une connexion (de type java.sql.Connection) sur la base courante. Cette méthode propage les exceptions de type SQLException.
 - getConnection()** retournant la connexion courante
 - closeConnection()** fermant la connexion courante. Cette méthode propage les exceptions de type SQLException.
- Créer une classe **MaConnexionJDBC** implantant l'interface **ConnexionJDBC** et possédant une variable d'instance «connexion».
- Créer une classe tournois et définir la méthode **traitePremiereRequete()** qui affiche le contenu de la table Coureur.

Rappel

2 Utilisation de JDBC

2.1 Déclaration du pilote JDBC

```
import java.sql.*;

public class ExempleJDBC
{
    public static void main(String[] Args)
    {
        try {
            Class.forName("org.gjt.mm.mysql.Driver");
        }
        catch (Exception E) {
            System.err.println("Pas de pilote !");
        }

        ... connexion et utilisation de la base ...
    }
}
```

- L'appel à `forName` déclenche un chargement dynamique du pilote.
- Un programme peut utiliser plusieurs pilotes, un pour chaque base de données.
- Le pilote doit être accessible à partir de la variable d'environnement `CLASSPATH`.

2.2 Connexion à la base de données

Voilà le code de connexion à la base :

```
try {
    String url = "jdbc:mysql://localhost/dbessai";
    Connection conn =
        DriverManager.getConnection(url, "user", "password");

    ... utilisation de la base ...
}
catch (SQLException E) {
    System.err.println(E.getMessage());
}
```

L'URL est de la forme

`jdbc:sous-protocole:sous-nom`
`jdbc:oracle://srv.dil.univ-mrs.fr:1234/dbtest`
`jdbc:odbc:msql;USER=fred;PWD=secret`

2.3 Les requêtes en JDBC

Un exemple d'utilisation :

```

Statement st = conn.createStatement();

ResultSet rs = st.executeQuery(
    "SELECT Nom,Prenom,Age FROM personne " +
    "ORDER BY age"
);

while (rs.next()) {
    System.out.println("Nom      : " + rs.getString(1));
    System.out.println("Prénom   : " + rs.getString(2));
    System.out.println("Age      : " + rs.getString(3));
}

rs.close(); // Fermeture
st.close();
conn.close();

```

2.4 La classe java.sql.ResultSet

Accès aux valeurs :

```

Type getType( int );
Type getType( String );
boolean next();

```

2.5 Correspondance des types Java / SQL

SQL	Java
CHAR	String
VARCHAR	String
LONGVARCHAR	String
NUMERIC	java.math.BigDecimal
DECIMAL	java.math.BigDecimal
BIT	boolean
TINYINT	byte
SMALLINT	short
INTEGER	int
BIGINT	long
REAL	float
FLOAT	double
DOUBLE	double
BINARY	byte[]
VARBINARY	byte[]
LONGVARBINARY	byte[]

2.6 Correspondance des dates et heures

Correspondance des dates

SQL	Java	Explication
DATE	java.sql.Date	codage de la date
TIME	java.sql.Time	codage de l'heure
TIMESTAMP	java.sql.Timestamp	codage de la date et de l'heure

3 Modification de la base

3.1 Insertion de lignes

Un exemple :

```
Statement st = conn.createStatement();

int nb = st.executeUpdate(
    "INSERT INTO personne(Nom,Age) " +
    "VALUES ('" + nom + "', " + age + ")"
);

System.out.println(nb + " ligne(s) insérée(s)");
st.close();
```

Ce principe est aussi utilisable pour les instructions UPDATE et DELETE.

3.2 Difficultés à manipuler des données

Un exemple :

```
Statement st = conn.createStatement();

int nb = st.executeUpdate(
    "UPDATE personne " +
    "SET Age = " + age + " " +
    "WHERE Nom = '" + nom + "'"
);
```

Inconvénients : solution coûteuse (boucle) et difficile à mettre en oeuvre.

3.3 SQL Préformaté

Code SQL avec partie variable :

```
PreparedStatement st = conn.prepareStatement(
    "UPDATE personne SET Age = ? " +
    "WHERE Nom = ? "
);

for( ... ) {
    st.setInt(1, age[i]);
    st.setString(2, nom[i]);
    st.execute();
}
```

On rappelle aussi que :

- le protocole de connexion et type de pilote : `com.mysql.jdbc.Driver`
- nom de la machine sur laquelle se trouve la base : `// jdbc:mysql://localhost/`
- numéro de port de connexion à la base : `Pas de port`