

TP BDR - JDBC

Introduction:

L'objectif de ce TP est savoir accéder à un ou plusieurs SGBD depuis une application java, en utilisant l'interface JDBC. Savoir parcourir le résultat d'une requête, définir une requête paramétrée puis l'invoquer, présenter le résultat d'une requête, interroger le dictionnaire (métadonnées) du SGBD. Savoir mettre en œuvre l'évaluation de requêtes réparties en utilisant JDBC : combiner les résultats de plusieurs requêtes posées sur différents SGBD.

Vous créez sur MySQL la base de données tennis dont le schéma est :

Joueur2 (NuJoueur, Nom, Prenom, AnNaiss, Nationalite)

Gain2 (NuJoueur, LieuTournoi, Annee, Prime, Sponsor)

Rencontre2 (NuGagnant, NuPerdant, LieuTournoi, Annee, Score)

Les attributs NuGagnant, NuPerdant et NuJoueur sont définis sur le même domaine. Les clés des relations sont soulignées.

Le contenu de la base est fourni en PJ.

Accès à une base de données avec JDBC

JDBC permet d'accéder au SGBD depuis une application écrite en langage java.

Accéder à de la documentation ici <http://docs.oracle.com/javase/7/docs/technotes/guides/jdbc/>

ou ici <http://www.commentcamarche.net/contents/558-java-l-api-jdbc>

Afficher la documentation en ligne pour faire le TP.

Utilisation des bibliothèques Java

Les bibliothèques java sont regroupées en packages. Un package contient plusieurs interfaces et plusieurs classes. Une classe (et une interface) contient des méthodes et des variables. La documentation en ligne permet de naviguer dans les bibliothèques pour déterminer les classes, interfaces et méthodes à utiliser pour construire une application java. Les classes et interfaces JDBC du package java.sql permettent d'accéder à une base de données. Naviguer dans la documentation du package java.sql pour répondre aux questions suivantes :

- a) Donner le nom et le type des paramètres des méthodes getConnection de la classe DriverManager.
- b) A quoi sert la méthode next de l'interface ResultSet ?
- c) A quoi sert la méthode setMaxRows de l'interface Statement ?
- d) Quelles sont les sous-interfaces de l'interface Statement ?

1. Première connexion à une base de données

Installer l'environnement logiciel (Java, JDBC, MySQL) puis exécuter le programme Joueur.

1.1) Etudier le programme Joueur.java. (Voir en annexe). Commenter brièvement les lignes importantes pour expliquer chaque étape du programme.

2 Accès paramétré à la base de données

2.1) Ecrire un programme MaxPrime qui calcul la prime maximale pour un tournoi et pour une année donnée (année à saisir par l'utilisateur).

2.2) Copier et modifier le programme précédent pour obtenir le programme MaxPrime2.java qui exécute la même requête en boucle, en demandant à chaque itération une nouvelle valeur à l'utilisateur. Le programme

MaxPrime2 se termine lorsque la saisie de l'utilisateur est vide. Pour améliorer les performances du programme, deux conditions sont requises :

- la connexion vers le SGBD est ouverte une seule fois pendant toute la durée du programme (i.e., réutiliser le même objet de type Connection à chaque itération).
- le SGBD analyse la requête une seule fois pendant toute la durée du programme (i.e., utiliser l'interface PreparedStatement pour définir une requête paramétrée ; utiliser la méthode setInt pour affecter une valeur à un paramètre de la requête).

3 Requête générique

3.1) Quel est le rôle de l'interface ResultSetMetaData et de sa méthode getColumnCount ?

3.2) Ecrire le programme *Generique.java* pour traiter une requête SQL quelconque passée en paramètre, et afficher les valeurs des tuples du résultat sous la forme « val1 val2 ... valn » (un tuple par ligne).

Tester l'exécution avec la requête donnant tous les Joueurs nés en 1972 :

```
java Generique "select * from Joueur2 where annaiss = 1972"
```

3.3) Compléter le programme pour afficher en en-tête le nom des attributs du résultat.

Exemple d'exécution (ne pas chercher à tabuler le résultat)

NUJOUEUR	NOM	PRENOM	ANNAISS	NATIONALITE
1	MARTINEZ	Conchita	1972	Espagne
14	SAMPRAS	Pete	1972	Etats-Unis

4 Schéma d'une relation

4.1) Quel est le rôle de l'interface DatabaseMetaData et comment obtenir une instance de ce type à partir d'une connexion? Expliquer la méthode getColumn de l'interface DatabaseMetaData. Dans un SGBD les relations sont généralement organisées en hiérarchie à 2 niveaux (catalogue et schéma). Ainsi, une relation appartient à un schéma lui-même appartenant à un catalogue.

4.2) Créer le programme Schema.java qui affiche le nom et le type des attributs d'une relation passée en paramètre.

Exemple d'exécution :

```
java Schema JOUEUR2 // écrire le nom de la relation en MAJUSCULE
Le schéma de JOUEUR2 est : // résultat affiché
NOM          TYPE
-----
NUJOUEUR     NUMBER
NOM           VARCHAR2
PRENOM        VARCHAR2
ANNAISS       NUMBER
NATIONALITE   VARCHAR2
```

5 Jointure inter-bases

Soit une deuxième base de données (située dans un autre SGBD différent du premier), contenant la relation SPONSOR (NOM, NATIONALITE) // **nationalité des sponsors**

La deuxième base de données est accessible seulement en lecture, par une connexion à la base MySQL en tant qu'utilisateur «anonyme» avec le mot de passe «anonyme». La relation Sponsor contient 100 000 tuples (Vous pouvez générer des tuples et insérer dans la base). Soit la requête R1 : «Donner le nom et la nationalité des joueurs avec le nom et la nationalité de leurs sponsors, dans l'ordre des noms de joueur».

5.1) Quelle est la durée approximative d'une lecture séquentielle de la relation Sponsor ? Répondre en écrivant le programme Generique2.java (obtenu en modifiant Generique.java), puis en utilisant la commande time : `time java Generique2 "requête"...` // mesure le temps de réponse de la requête

Veiller à omettre l'affichage du résultat dans le terminal pour éviter de mesurer le temps d'affichage au lieu du temps de lecture des données.

Environnement Technique pour le TP

Installer l'environnement logiciel (Java, JDBC)