

Bases de Données TP5

L3 Informatique 2011-2012

Java et les bases de données : introduction à JDBC

Dans ce TP on va utiliser les classes de la JDBC pour se connecter à la base de données. Connectez-vous sur la machine de votre choix. Pour se connecter à Postgres (qui se trouve sur Nivose), vous avez besoin du driver JDBC pour Postgres. Un tel driver est simplement une classe Java qu'il faut ajouter dans le classpath pour que le compilateur sache la retrouver. Pour cela, vous pouvez exécuter la commande suivante:

```
export CLASSPATH=/ens/reaultm/Public/BD6/TP5/postgresql.jdbc4.jar:$CLASSPATH
```

Vous devrez par la suite intégrer cette commande à vos fichiers de configuration. Importez sur votre compte les fichiers `PasswordField.java`, `ChaineHotels.java` et `ConnectionHotel.java` disponibles dans le répertoire `/ens/reaultm/Public/BD6/TP5`. La classe `PasswordField.java` permettent de taper au clavier un mot qui ne s'affiche pas à l'écran. Elle est utiles pour entrer le mot de passe qui vous permet de vous connecter à Postgres via un exécutable java. Vous n'avez pas besoin de comprendre le code de ces classes. Les classes `ChaineHotels.java` et `ConnectionHotel.java` contiennent un squelette de code Java compilable et exécutable. La classe `ChaineHotels` propose un menu à l'utilisateur (création de table, insertion de tuples, etc.). La plupart de ces fonctionnalités ne sont pas encore implémentées (un message vous l'indique si vous tenter de l'activer). Le but de ce TP est d'implémenter ces fonctionnalités.

Exercice 1 - Test de la connexion

Compiler les classes. Pour exécuter, tapez :

```
java ChaineHotels login
```

où `login` correspond à votre compte Postgres et le mot de passe demandé ensuite est également celui de votre compte Postgres. La connexion réussit si vous voyez un menu s'afficher, sinon un message d'erreur apparaît.

Exercice 2 - Utilisation de Statement

Cet exercice vise à la gestion de la relation `Hotel` définie par le schéma suivant :

`HOTEL(num_hotel, nom_hotel, ville, étoiles, directeur)`

Les demandes descriptives supplémentaires sont les suivantes:

- Un hôtel peut avoir de 0 à 5 étoiles;
- la chaîne hôtelière a des hôtels seulement dans Paris, Lyon et Marseille;
- deux hôtels ne peuvent pas avoir le même directeur.

Le code sql correspondant est donc :

```

create table hotel(
    num numeric(6) not null primary key,
    nom varchar(40) not null,
    ville varchar(9) not null,
    etoiles numeric(1),
    directeur varchar(40) not null unique,
    constraint ch_etoiles check (etoiles>=0 and etoiles<=5),
    constraint ch_ville check (ville in ('paris','marseille','lyon'))
);

```

Lisez attentivement le code de la fonction main du fichier `ChaineHotels.java`. Avant d'afficher le menu, elle crée une instance de la classe `ConnectionHotel.java`. Regardez maintenant le constructeur de cette classe. Vous constaterez qu'après avoir chargé le driver, une connexion est établie avec la base. A partir de cette connexion, vous allez pouvoir exécuter des instructions SQL (Statement, PreparedStatement, etc.). La documentation est disponible à l'adresse suivante, dans le package `java.sql`:

<http://docs.oracle.com/javase/6/docs/api/java/sql/package-summary.html>

Quel est le type de `conn` ? que peut-on faire avec ?

La fonction `printMenu` imprime un menu à l'écran et attend le choix de l'utilisateur. Ce choix est filtré par une instruction `switch`. Implémentez dans l'ordre tous les cas du `switch` de la fonction `printMenu()` en utilisant exclusivement des objets de type `Statement`.

Remarque : pour lire des valeurs à partir du terminal, vous pouvez utiliser les fonctions `readInt()` et `readString()` définies dans la classe `ChaineHotels`.

Exercice 3 - Utilisation de PreparedStatement

Lorsqu'une instruction SQL est susceptible d'être exécutée plusieurs fois, il est plus efficace de la précompiler d'abord (son plan d'exécution est calculé une seule fois). Pour cela, il faut utiliser des objets de type `PreparedStatement`.

Un exemple d'utilisation est le suivant:

```

PreparedStatement pstmt = conn.prepareStatement("UPDATE EMPLOYEES
SET SALARY = ? WHERE ID = ?");

pstmt.setInt(1,153833);
pstmt.setInt(2,92);
pstmt.execute(); // execution de l'instruction sql
pstmt.clearParameters();//nettoyage des parametres
pstmt.setInt(1,234309);
pstmt.setInt(2,104);
pstmt.execute();
pstmt.close();

```

L'instruction SQL est définie en remplaçant les valeurs des paramètres par un point d'interrogation. Les paramètres sont ensuite donnés en fonction de leur type (méthodes `setInt`, `setString`, etc.) et de leur position : le premier point d'interrogation a pour position 1.

Implémenter les cas 5, 6 et 7 en utilisant des objets de type `PreparedStatement`.