

TD 7

1 JDBC

1.1 Préliminaires

En cas de doute sur les détails d'une fonction, on pourra consulter la partie de la documentation Java concernant l'API JDBC : <http://java.sun.com/j2se/1.5.0/docs/api>.

(Par ailleurs, le matériel Java nécessaire pour faire ce TD et considéré comme prérequis est très simple.)

1. Récupérez sur la page du module le script SQL *jdbc1.sql* qui crée la table et les procédures suivantes. Consultez-le et sauvez-le chez vous.

```
table t(nom varchar2(10) primary key, age integer) -- des personnes
procedure rajeunit(n varchar2) -- decremente l'age de n de 1
function f return integer -- compte le nombre de lignes de t
```

2. Récupérez sur la page du module le programme Java *Menu.java* qui contient un menu pour appeler vos futures fonctions. Chaque exercice du TD consistera à écrire une fonction Java, que vous insérerez dans ce menu. Vous ne rattraperez pas les erreurs, sauf si c'est demandé explicitement dans l'énoncé. Compilez et exécutez ce programme pour vérifier que tout fonctionne.
Attention : les fonctions de ce menu sont toutes sans paramètre et de type void, mais vous pourrez avoir à changer cela.

1.2 Connexion

1. Ecrivez une fonction de connexion sur un compte Oracle dont le login et le mot de passe sont lus au clavier. La connexion est stockée dans une variable locale du main.
2. Fonction de déconnexion.
3. Testez une connexion correcte, puis une incorrecte : expliquez ce que vous constatez.
4. Une fois que la fonction de connexion fonctionne, vous la modifierez pour que le login soit "en dur" dans le code et qu'il n'y ait rien à lire au clavier. En outre, vous effectuerez cette connexion dès le lancement du programme, donc sans passer par le menu. Ceci afin d'accélérer les tests dans la suite du TD.

1.3 Exécution d'ordres BD

1. Ordres SQL non requête.
 - (a) Fonction qui incrémente de 1 l'âge des personnes dont le nom est *toto* ou dont l'âge est inférieur ou égal à 22 (donc sans paramètre). Testez les différentes valeurs de retour de *executeUpdate* : 0 ou plus (vous pourrez avoir à modifier la base).
 - (b) Même question mais le nom et l'âge sont lus au clavier. Vous utiliserez la classe *Statement* (et non *PreparedStatement*). (Vous ne testerez plus les valeurs de retour de *executeUpdate*.)

(c) Même question mais l'ordre sera exécuté plusieurs fois (disons deux fois pour simplifier l'exercice).

2. Appel de procédures stockées.

(a) Fonction qui exécute la procédure stockée PL/SQL *rajeunit* ci-dessus avec un paramètre lu au clavier.

(b) Fonction qui exécute la fonction stockée PL/SQL *f* ci-dessus.

(c) (En fin de TD.) Même question pour *rajeunit*, mais le paramètre est passé par concaténation, pas avec le "?".

1.4 Curseurs

1. Affichez les noms des personnes ayant strictement plus de 20 ans (donc sans paramètre). Testez avec zéro, une ou plusieurs lignes renvoyées (vous pourrez avoir à modifier la base).

2. (En fin de TD.) Même question mais en exécutant la requête deux fois, avec à chaque fois l'âge lu au clavier (donc avec *PreparedStatement*).

1.5 SQL dynamique

(En fin de TD.) Fonction qui lit au clavier un nom de table et la détruit (*drop*). Comparez à PL/SQL.