



M2106

Programmation et administration
des bases de données

COURS 4 : ACCÈS AUX BASES DE DONNÉES VIA JDBC

Mai 23, 2017

BELFODIL Adnene – BENDIMERAD Anes



- Architecture générale
- Utilisation du driver JDBC
- JDBC et PLSQL

- Architecture générale
- Utilisation du driver JDBC
- JDBC et PLSQL

JDBC (Java DataBase Connectivity)

JDBC est une API qui permet l'accès à des bases de données avec le langage SQL depuis un programme Java. L'API JDBC est 'presque' totalement indépendante des SGBD. Elle fournit uniquement des interfaces qui doivent être implémentées par les drivers correspondants.

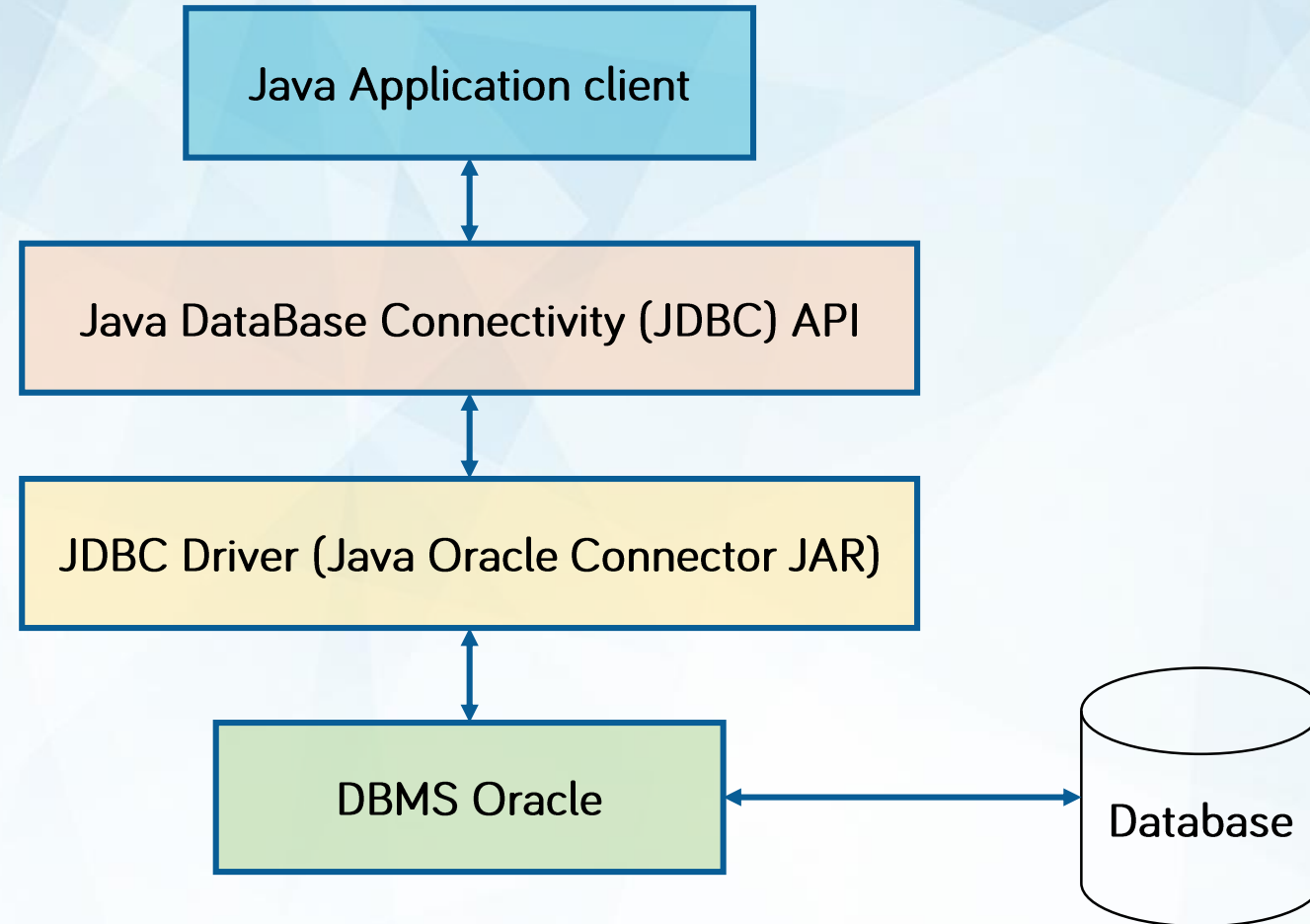
Drivers d'un SGBD

Un **driver** est une bibliothèque qui implémente les **méthodes fournies par JDBC** pour pouvoir interagir avec un SGBD particulier (Dépend du SGBD).

Driver Oracle

Dans ce cours, nous allons travailler avec le driver correspondant à Oracle:

<http://www.oracle.com/technetwork/database/features/jdbc/default-2280470.html>



- Architecture générale
- Utilisation du driver JDBC
- JDBC et PLSQL

Driver Oracle

Avant de pouvoir interagir avec une base oracle, il est nécessaire de télécharger le driver et l'inclure dans le projet





Syntaxe - ouverture

```
DriverManager.getConnection  
(chaine_de_connexion,  
nom d'utilisateur, mdp)
```

`jdbc:oracle:thin:@IP_address:port:DB_name`

Exemple

```
Connection con = DriverManager.getConnection  
(jdbc:oracle:thin:@134.214.112.67:1521:orcl,  
AdneneAnes, my_mot_de_passe)
```



Syntaxe – création d'interface de requêtage

```
Statement stmt=con.createStatement();
```

Statement permet de :

1. Interroger la base **executeQuery** (LID)
2. modifier la base **updateQuery** (LMD)
3. Tout autre requête SQL **execute** (ex : Create)



Syntaxe - Sélection

```
ResultSet rs=stmt.executeQuery  
("select ...");  
while(rs.next()) {  
System.out.println(rs.getType(#id_col));  
}
```

Exemple

```
ResultSet rs=stmt.executeQuery  
("select name,age from contact");  
while(rs.next()) {  
System.out.println(rs.getString(1) +  
rs.getInt(2));  
}
```



Syntaxe - Manipulation

```
int nb_lignes_modifies =  
stmt.executeUpdate("DML Query");
```

Exemple

```
int nb_lignes_modifies =  
stmt.executeUpdate("Delete * from contact");
```



Syntaxe – Création

```
boolean succes =  
stmt.execute ("Create Table ...");
```

Exemple

```
Success = stmt.execute ("Create Table  
contact ( name varchar2(100),  
age number(3)  
)");
```



Syntaxe – Fermeture

```
con.close();
```

- Architecture générale
- Utilisation du driver JDBC
- **JDBC et PLSQL**

Motivation

- L'interface `CallableStatement` permet d'appeler des procédures ou des fonctions stockées.
- On indique le nom de la procédure ou de la fonction requise lors de l'initialisation de l'objet `CallableStatement` grâce à la méthode `prepareCall()` de l'interface `Connection`.
- Deux formes possibles, selon que la procédure ou la fonction stockée comporte des paramètres lors de l'appel ou n'en comporte pas

Comment ?

- Sans paramètres :
`CallableStatement cst = myconnexion prepareCall ("call nom procedure");`
- Avec des paramètres :
`CallableStatement cst = myconnexion prepareCall ("? = call nom fonction
(? , ? , ...)")`
Ou
`CallableStatement cst = myconnexion prepareCall ("call nom procedure (? ,
? , ...)") ;`

Exemple – Appel de fonction PLSQL

```
CallableStatement cst = myconnexion.prepareCall ("? = call  
getUserNameById (? )");  
cst.setInt (2,13291);  
cst.registerOutParameter (1,java.sql.Types.VARCHAR2);  
boolean succes = cst.execute ( );  
String username = cst.getString (1);  
cst.close ( );
```

Exemple – Appel de procédure PLSQL

```
CallableStatement cst = myconnexion.prepareCall ("call  
updateContactById (?, ?, ?)"); #(id in, new_name in, old_name out)  
cst.setInt (1, 13291);  
cst.setString (2, 'Adnene');  
cst.registerOutParameter (3, java.sql.Types.VARCHAR2);  
boolean succes = cst.execute ( );  
String old_name = cst.getString (3);  
cst.close ( );
```

Projet :

Proposer une petite application Java qui fait usage d'une base de donnée en utilisant JDBC et PL/SQL.

Exemple :

Une interface permettant à un reviewer de donner un score à un film et de visualiser les scores d'un film ou d'un ensemble de films en utilisant la base de données du TP2

Organisation :

Le projet se fait par binôme ou trinôme, une démonstration du projet d'une durée de 5 minutes est prévue par groupe à la dernière séance.