

V- JDBC [Connexion JAVA-Oracle]

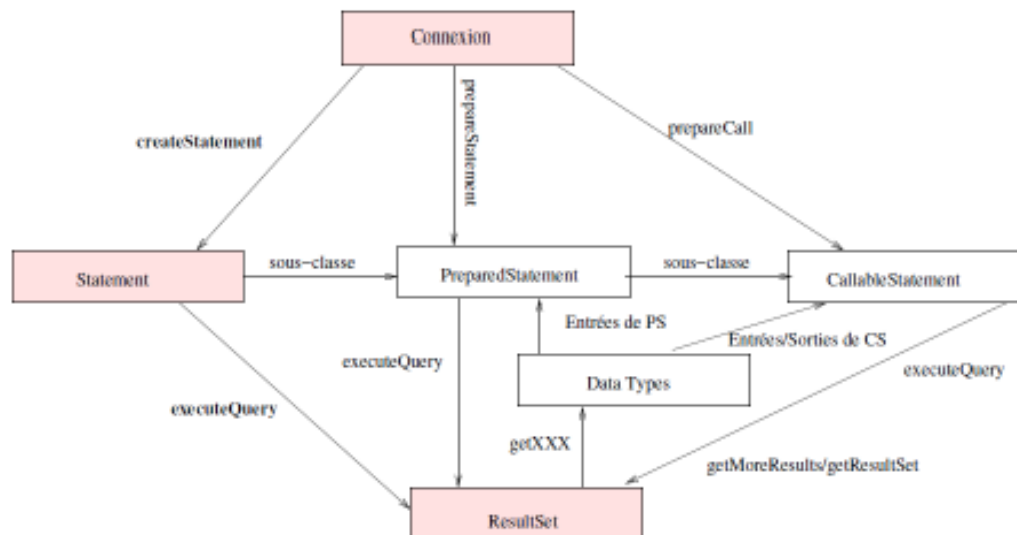
1. Usage :

JDBC (Java DataBase Connectivity) permet d'exécuter, depuis un programme Java, l'ensemble des ordres SQL reconnus par la base de données cible. La base de données doit reconnaître le langage ANSI SQL-2. JDBC est une API (Application Programming Interface) qui permet d'exécuter des instructions SQL et fait partie du JDK (Java Development Kit Paquetage java.sql : import java.sql.*;



2. JDBC : Fonctionnement

a. Les classes et interfaces du package java.sql



b. Etapes d'un programme utilisant JDBC :

1. Mettre en place le pilote ou driver.
2. Etablir une connexion avec une source de données.
3. Effectuer les requêtes.
4. Utiliser les données obtenues pour des affichages, des traitements statistiques, etc.
5. Mettre à jour les informations de la source des données.
6. Terminer la connexion.
7. Eventuellement, recommencer en 1.

3. Exemples

- Mettre en place le pilote ou driver :

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

- Etablir une connexion avec une source de données :

(Créer un objet Connexion) : `Connection maConnexion=DriverManager.getConnection(url);`

Avec url : String contenant l'adresse de la base de données

(Créer un objet Statement) : `Statement maRequeteSQL=maConnexion.createStatement();`

- Effectuer les requêtes :

```
ResultSet monResultat= maRequeteSQL.executeQuery(texteRequeteSQL);
```

Avec texteRequeteSQL : String contenant le texte de la requête. Par exemple : `SELECT * FROM Client`

- Utiliser les données obtenues pour des affichages, des traitements statistiques, etc.

```
ResultSet monResultat = maRequeteSQL.executeQuery (SELECT login, nomClient FROM Client");
```

ResultSet et ses méthodes :

- ☐ Résultat dans un ResultSet
- ☐ Parcours analogue à celui d'un curseur avec la méthode next, et accès aux colonnes avec getXXX

EXEMPLE :

```
While (monResultat.next()) {  
    String nom = monResultat.getString("nomClient");  
    int login=monResultat.getInt ("login");  
    //traitement des données récupérées  
}
```

- ☐ Le premier next positionne sur la première ligne.
- ☐ Paramètres de getXXX : nom de l'attribut ou rang dans la requête (sous forme d'entier). Obligatoire pour les attributs calculés. (MAX(...)) ou quand les noms ne sont pas connus (SELECT *...)

EXEMPLE :

```
While (monResultat.next()) {  
    String nom = monResultat.getString(2);  
    int login=monResultat.getInt (1);  
    //traitement des données récupérées  
}
```