

## Manipulation du Pilote JDBC via JavaApp :

### ✓Objectifs :

- Établir une connexion à la base de données (BD).
- Exécuter une requête Update avec un Statement
- Obtenir les résultats dans un ResultSet
- Examiner quelques méthodes du ResultSet.

### ✓Rappel :

- JDBC est un API-JAVA permettant de se connecter à une base de données en utilisant des requêtes SQL.
- Pour se connecter à MySQL depuis Java, vous devez utiliser le pilote JDBC (*Connector/J*) de MySQL.

### ✓Création d'une application JDBC :

- Après l'importation des packages nécessaires (*import java.sql.\** ;), suivre les 5 étapes suivantes : **E.1- Enregistrer le pilote JDBC :** initialiser un pilote afin d'ouvrir un canal de communication avec la BD. **E.2- Ouvrir une connexion :** utiliser de la méthode *DriverManager.getConnection()* pour créer un objet Connection, qui représente une connexion physique avec la base de données.  
**E.3- Exécuter une requête :** instancier un objet de type Statement pour créer et soumettre une instruction SQL à la BD.  
**E.4- Extraire les données de l'ensemble de résultats :** utiliser la méthode *ResultSet.getXXX()* appropriée pour récupérer les données de l'ensemble de résultats.  
**E.5- Nettoyer l'environnement :** fermer explicitement toutes les ressources de la base de données.

Considérons une **application JAVA** permettant de gérer les notes des étudiants inscrits dans le module "Architecture Distribuée" en SMI-S6. Les informations des étudiants doivent être stockées dans un serveur de base de données (avec SGBD *MySQL*) dans une table dont la structure est la suivante :

ID	CNE	Nom	Note
Type <i>Entier (PK, AI)</i>	Type <i>Long</i>	Type <i>Texte(100)</i>	Type <i>Réel</i>

## Exercice 1 : Data Definition Language (LDD / DDL)

1. Établir une classe Java "*createdb*" pour créer une base de données "**DBStds**", sur le serveur DataBase-*MySQL* ?
  - ✓ Protocole de connexion : *jdbc:mysql*
  - ✓ Adresse IP du serveur base de données : **10.18.1.1**
  - ✓ Port MySQL : **3306**
  - ✓ Authentification : *User : root* et *PassWord : FSTet.23*
2. Créer une classe Java "*createtbstd*" qui permet de créer la table "**STUDENT**" dans la base de données "**DBStds**" ?

## Exercice 2 : Data Manipulation Language (LMD / DML)

Il s'agit dans cette partie de manipuler les données des étudiants inscrits dans le module "Architecture Distribuée". 3. Définir une classe Java "*addsutds*", pour inscrire les étudiants dans la table "**STUDENT**" de la BD "**DBStds**" :

ID	CNE	Nom	Note
1	20854367	Boujdain Amal	11
2	19237645	Zougrar Imad	14
3	27361238	Dadi Maroan	04

4. Pour traiter les demandes de vérification, une classe "*updtlnote*" à mettre en place pour modifier les notes, comme

le cas d'une demande de l'étudiante "**Boujdain Amal**", changer la note en **09/20** au lieu du ~~**11/20**~~? 5. Éditer une classe java " **delnotenv** ", afin de supprimer les étudiants ayant une note Non-Validé (**Note < 05/20**) ? **Exercice 3 :**

### **Langage de requêtes**

6. Créer une classe Java " **listecr** ", pour afficher la liste des étudiants convoqués à la session de rattrapage (**Note<10/20**) ? en ordre croissant des noms ?
7. Éditer une classe Java " **beststd** ", qui permet d'afficher les informations de l'étudiant **major** de promo ?