JDBC – Prepared Statement

Guillaume Dufrêne – Lionel Seinturier

Université de Lille – Sciences et Technologies

Prepared Statement

Définition

Requête SQL précompilée (*prepared*) et paramétrée

Format

- requête SQL “à trou”

Exemple

```
PreparedStatement pst =
  cx.prepareStatement("SELECT * FROM personnes WHERE nom=? AND age>?");
```

Prepared Statement

Utilisation

```
PreparedStatement pst =                                          #1        #2
  cx.prepareStatement("SELECT * FROM personnes WHERE nom=? AND age>?");

pst.setString( 1, "Paul" );
pst.setInt( 2, 25 );

ResultSet rs = pst.executeQuery();
```

Tous les types de requêtes SQL peuvent être paramétrés
Exemple

```
PreparedStatement pst =
  cx.prepareStatement("UPDATE ages SET age = ? WHERE nom = ?");
```

Prepared Statement

Avantages

- Efficace
- Meilleure protection contre l'injection de code SQL
- Délimiteurs de valeur corrects

```
                          "root"    "' OR '1' = '1"
boolean checkPassword(String l, String p) {
  return
    ctx.createStatement()
    .executeQuery(
        "SELECT * FROM personnes WHERE login = '" + l + "'"
        + "AND password = '" + p + "'"
    )
    .next();
}
```